

Alessandro Micarelli  
John Stamper  
Kitty Panourgia (Eds.)

LNCS 9684

# Intelligent Tutoring Systems

13th International Conference, ITS 2016  
Zagreb, Croatia, June 7–10, 2016  
Proceedings

 Springer

*Commenced Publication in 1973*

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

## Editorial Board

David Hutchison

*Lancaster University, Lancaster, UK*

Takeo Kanade

*Carnegie Mellon University, Pittsburgh, PA, USA*

Josef Kittler

*University of Surrey, Guildford, UK*

Jon M. Kleinberg

*Cornell University, Ithaca, NY, USA*

Friedemann Mattern

*ETH Zurich, Zürich, Switzerland*

John C. Mitchell

*Stanford University, Stanford, CA, USA*

Moni Naor

*Weizmann Institute of Science, Rehovot, Israel*

C. Pandu Rangan

*Indian Institute of Technology, Madras, India*

Bernhard Steffen

*TU Dortmund University, Dortmund, Germany*

Demetri Terzopoulos

*University of California, Los Angeles, CA, USA*

Doug Tygar

*University of California, Berkeley, CA, USA*

Gerhard Weikum

*Max Planck Institute for Informatics, Saarbrücken, Germany*

More information about this series at <http://www.springer.com/series/7408>

Alessandro Micarelli · John Stamper  
Kitty Panourgia (Eds.)

# Intelligent Tutoring Systems

13th International Conference, ITS 2016  
Zagreb, Croatia, June 7–10, 2016  
Proceedings



*Editors*

Alessandro Micarelli  
Roma Tre University  
Rome  
Italy

Kitty Panourgia  
Neoanalysis Ltd  
Athens  
Greece

John Stamper  
Carnegie Mellon University  
Pittsburgh, PA  
USA

ISSN 0302-9743                      ISSN 1611-3349 (electronic)  
Lecture Notes in Computer Science  
ISBN 978-3-319-39582-1              ISBN 978-3-319-39583-8 (eBook)  
DOI 10.1007/978-3-319-39583-8

Library of Congress Control Number: 2016939998

LNCS Sublibrary: SL2 – Programming and Software Engineering

© Springer International Publishing Switzerland 2016

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

This Springer imprint is published by Springer Nature  
The registered company is Springer International Publishing AG Switzerland

## Preface

The 13th International Conference on Intelligent Tutoring Systems, ITS 2016, was held in Zagreb, Croatia, during June 7–10, 2016.

The theme of the conference was: “Adaptive Learning in Real World Contexts.” It stressed the need for devising learning systems that can adapt adequately to users, furnishing them with the knowledge they are seeking in real-world contexts, namely, systems that are effectively usable in everyday learning situations, such as courses in schools or training programs in companies, but also in informal situations, such as *Web- or App-provided help* in using new technologies. The above theme encouraged conference participants to think about this educational need in our increasingly complex everyday world.

The call for scientific papers solicited work presenting substantive new research results in using advanced computer technologies and interdisciplinary research for enabling, supporting, and enhancing human learning. A posters track was also organized, providing an interactive forum for authors to present research prototypes to conference participants, as well as work in progress.

The international Program Committee consisted of 102 leading members of the intelligent tutoring systems community (34 senior and 68 regular) as well as highly promising younger researchers. The Program Committee chairs were Alessandro Micarelli from Roma Tre University, Italy, and John Stamper from Carnegie Mellon University, USA.

Scientific papers were reviewed by three reviewers (one of whom was senior) through a double-blind process. Only 15 % of submitted papers were accepted as full papers, about 27 % were accepted as short papers, and just over 30 % were accepted as posters. These rates show that ITS 2016 was a rather selective, high-quality conference. We believe that the chosen full papers describe some very significant research and the short papers some very interesting new ideas, while the posters present research in progress that deserves close attention.

In the review process we generally respected the reviewers’ evaluations, especially those made by the senior reviewers. Only in special cases did we depart from the latter’s recommendations, and only to upgrade the papers involved.

A separate young researchers’ track provided a forum in which PhD students could present and discuss their work during its early stages, meet peers with related interests, and work with more senior members of the field (mentors). The young researchers’ track chairs were Darina Dicheva from Winston-Salem State University, USA, and Toby Dragon from Ithaca College, USA. This track received 11 submissions, of which three were accepted as papers and four as posters.

The management of the review process and the preparation of the proceedings were handled through EasyChair.

The ITS 2016 program also included the following workshops and tutorial selected by the workshop chairs, Stephen E. Fancsali from Carnegie Learning, Pittsburgh, USA, and Tsukasa Hirashima from Hiroshima University, Japan.

Workshops:

- First International Workshop on Supporting Dynamic Cognitive Affective and Metacognitive Processes (SD-CAM) by Jason M. Harley and Claude Frasson.
- 2nd International Workshop on Social Computing in Digital Education (SocialEdu 2016) by Andrew Koster, Tiago Thompsen Primo, Rosa Maria Vicari, Takao Terano and Fernando Koch.
- First International Workshop on Intelligent Mentoring Systems (IMS 2016) by Amali Weerasinghe, Vania Dimitrova, Lydia Lau and Antonija Mitrovic.
- 2nd International Workshop on Affect, Meta-Affect, Data and Learning (AMADL 2016) by Benedict du Boulay.
- Building ITS Bridges Across Frontiers by Stefan Trausan-Matu, Stefano Cerri and Mihai Dascalu.
- 5th Workshop on Intelligent Support for Learning in Groups (ISLG 2016) by Jennifer Olsen, Erin Walker, Roberto Martinwz- Maldonado, Ilya Goldin and Jihie Kim.

Tutorial:

- Educational Data Analysis Using LearnSphere (Ran Liu, Michae Eagle, Philip Pavlik, John Stamper)

In addition to the aforementioned contributors, we would also like to thank all the authors, the members of the Program Committee and the external reviewers, the Steering Committee and in particular its chair, Claude Frasson.

We are also thankful to our conference scientific sponsors for their support, and in particular Springer for sponsoring the Best Paper Award and NSF (National Science Foundation) for funding the YRT. Last but not least, we would like to salute the Institute of Intelligent Systems under the auspices of which this conference was held.

April 2016

Alessandro Micarelli  
John Stamper  
Kitty Panourgia

# Organization

## Conference Committee

### Program Chairs

Alessandro Micarelli      Roma Tre University, Italy  
John Stamper                Carnegie Mellon University, USA

### Organization Chair

Kitty Panourgia            Neoanalysis, Greece

### Workshops and Tutorial Chairs

Stephen E. Fancsali        Carnegie Learning, Pittsburgh, USA  
Tsukasa Hirashima        Hiroshima University, Japan

### Young Researcher Track Chairs

Darina Dicheva            Winston-Salem State University, USA  
Toby Dragon                Ithaca College, USA

### Award Chair

Claude Frasson            University of Montreal, Canada

The conference was held under the auspices of the **Institute of Intelligent Systems**.



## Program Committee

### Program Chairs

Alessandro Micarelli	Roma Tre University, Italy
John Stamper	Carnegie Mellon University, USA

### Senior Program Committee

Esma Aimeur	University of Montreal, Canada
Ryan Baker	Teachers College, Columbia University, USA
Jacqueline Bourdeau	TELU-UQAM, Canada
Kristy Elizabeth Boyer	North Carolina State University, USA
Bert Bredeweg	University of Amsterdam, The Netherlands
Paul Brna	University of Leeds, UK
Luigia Carlucci Aiello	Sapienza Università di Roma, Italy
Stefano A. Cerri	LIRMM: University of Montpellier and CNRS, France
Michel Desmarais	Ecole Polytechnique de Montreal, Canada
Vania Dimitrova	University of Leeds, UK
Benedict Du Boulay	University of Sussex, UK
Isabel Fernandez-Castro	University of the Basque Country, Spain
Claude Frasson	University of Montreal, Canada
Gilles Gauthier	UQAM, Canada
W. Lewis Johnson	Alelo Inc., USA
Judy Kay	The University of Sydney, Australia
Jean-Marc Labat	Université Paris 6, France
Susanne Lajoie	McGill University, Canada
H. Chad Lane	University of Illinois, Urbana-Champaign, USA
James Lester	North Carolina State University, USA
Diane Litman	University of Pittsburgh, USA
Rose Luckin	University College London, UK
Bruce McLaren	Carnegie Mellon University, USA
Tanja Mitrovic	University of Canterbury, Christchurch, New Zeland
Riichiro Mizoguchi	Japan Advanced Institute of Science and Technology, Japan
Wolfgang Nejdl	L3S and University of Hannover, Germany
Roger Nkambou	Université du Québec à Montréal (UQAM), Canada
Niels Pinkwart	Humboldt Universität zu Berlin, Germany
Carolyn Rose	Carnegie Mellon University, USA
Elliot Soloway	University of Michigan, USA
Stefan Trausan-Matu	Politehnica University of Bucharest, Romania
Gerhard Weber	University of Education Freiburg, Germany
Beverly Park Woolf	University of Massachusetts, USA
Kalina Yacef	The University of Sydney, Australia

**Program Committee**

Mohammed Abdel Razek	King Abdulaziz University, Saudi Arabia
Fabio Akhras	Renato Archer Center of Information Technology, Italy
Galia Angelova	Bulgarian Academy of Sciences, Bulgaria
Ana Arruarte	University of the Basque Country, Spain
Roger Azevedo	North Carolina State University, USA
Tiffany Barnes	North Carolina State University, USA
Maria Bielikova	Slovak University of Technology in Bratislava, Slovakia
Ig Ibert Bittencourt	Federal University of Alagoas, Brazil
Emmanuel Blanchard	IDÚ Interactive Inc., Canada
Stephen B. Blessing	University of Tampa, USA
Mary Jean Blink	TutorGen, Inc., USA
Christopher Brooks	University of Michigan, USA
Winslow Burleson	Arizona State University, USA
Nicola Capuano	University of Salerno, Italy
Ted Carmichael	University of North Carolina at Charlotte, USA
Chih-Kai Chang	National University of Tainan, Taiwan
Maher Chaouachi	McGill University, Canada
Min Chi	North Carolina State University, USA
Chih-Yueh Chou	Yuan Ze University, Taiwan
Mark Core	University of Southern California, USA
Evandro Costa Federal	University of Alagoas, Brazil
Alexandra Cristea	University of Warwick, UK
Scott Crossley	Georgia State University, USA
Cyrille Desmoulins	Université Joseph Fourier, France
Philippe Dessus	LSE, Grenoble, France
Darina Dicheva	Winston-Salem State University, USA
Sidney D'Mello	University of Notre Dame, USA
Peter Dolog	Aalborg University, Denmark
Michael Eagle	North Carolina State University, USA
Stephen Fancsali	Carnegie Learning, Inc., USA
Oliver Ferschke	Carnegie Mellon University, USA
Mark Floryan	University of Virginia, USA
Davide Fossati	Carnegie Mellon University, Qatar
Nobuko Fujita	University of Windsor, Canada
Ashok Goel	Georgia Institute of Technology, USA
José González-Brenes	Pearson, USA
Jason Harley	University of Montreal, Canada
Yusuke Hayashi	Hiroshima University, Japan
Cecily Heiner	Southern Utah University, USA
Tsukasa Hirashima	Hiroshima University, Japan
Ulrich Hoppe	University of Duisburg-Essen, Germany
Seiji Isotani	University of Sao Paulo, Brazil
Patricia Jaques	UNISINOS, Brazil

Heisawn Jeong	Hallym University, South Korea
Clement Jonquet	University of Montpellier - LIRMM, France
Imène Jraïdi	University of Montreal, Canada
Akihiro Kashiara	University of Electro-Communications, Japan
Nguyen-Thinh Le	Humboldt Universität zu Berlin, Germany
Philippe Lemoisson	Cirad, France
Carla Limongelli	Roma Tre University, Italy
Chao-Lin Liu	National Chengchi University, Taiwan
Ran Liu	Carnegie Mellon University, USA
Derek Lomas	University of California San Diego, USA
Vanda Luengo	Université Pierre et Marie Curie, France
Tatsunori Matsui	Waseda University, Japan
Manolis Mavrikis	London Knowledge Lab, UK
Riccardo Mazza	University of Lugano/University of Applied Sciences of Southern Switzerland
Kazuhisa Miwa	Nagoya University, Japan
Paul Mulholland	The Open University, UK
Germana Nobrega	Universidade de Brasília, Brazil
Amy Ogan	Carnegie Mellon University, USA
Luc Paquette	University of Illinois, Urbana-Champaign, USA
Zach Pardos	UC Berkeley, USA
Olga C. Santos	aDeNu Research Group, UNED, Spain
Erin Walker	Arizona State University, USA
Amali Weerasinghe	The University of Adelaide, Australia
Joseph Jay Williams	Harvard University, USA
Ruth Wylie	Arizona State University, USA

## **Organizing Committee**

### **Organization Chair**

Kitty Panourgia                      General Coordination/Proceedings/Program

### **Members**

Natalia Kakourou	Conference Publicity
Katerina Milathianaki	Registration
Alexia Kakourou	Coordination on Site
Simona Todorova	Conference Data Processing

### **Site Architect**

Isaak Tselepis

## Steering Committee

### Chair

Claude Frasson                      University of Montreal, Canada

### Members

Stefano Cerri	University of Montpellier II, France
Isabel Fernandez-Castro	University of the Basque Country, Spain
Gilles Gauthier	University of Quebec at Montreal, Canada
Guy Gouardères	University of Pau, France
Mitsuru Ikeda	Japan Advanced Institute of Science and Technology, Japan
Marc Kaltenbach	Bishop's University, Canada
Alan Lesgold	University of Pittsburgh, USA
James Lester	North Carolina State University, USA
Roger Nkambou	University of Quebec at Montreal, Canada
Giorgos Papadourakis	Technological Educational Institute, Crete, Greece
Fabio Paragua	Federal University of Alagoas, Brazil
Elliot Soloway	University of Michigan, USA
Daniel Suthers	University of Hawaai, USA
Stefan Trausen-Matu	University Politehnica of Bucharest, Romania
Beverly Woolf	University of Massachussets, USA

## Advisory Committee

### Members

Maria Grigoriadou	University of Athens, Greece
Judith Kay	University of Sidney, Australia

## External Reviewers

Ana Arruarte	Ye Mao
Nigel Bosch	Noboru Matsuda
François Bouchet	Miki Matsumuro
Acey Boyce	Thomas McTavish
Christa Cody	Joseph Michaelis
Mutlu Cukurova	Caitlin Mills
Jon A. Elorriaga	Junya Morita
Erik Harpstead	Behrooz Mostafavi
Tobias Hecking	Ivelina Nikolova
Andrew Hicks	Anna Pierri
Farzaneh Khoshnevisan	Arnab Saha
Mikel Larrañaga	Emmanuel Sander
Zhongxiu Liu	Chris Teplovs
Alexandra Luccioni Vorobyova	Guojing Zhou



## **Sponsors**

### **Best Paper Award**



### **Funding the YRT**



# Contents

## Full Papers

Understanding Procedural Knowledge for Solving Arithmetic Task by Externalization . . . . .	3
<i>Kazuhisa Miwa, Hitoshi Terai, and Kazuya Shibayama</i>	
Do Erroneous Examples Improve Learning in Addition to Problem Solving and Worked Examples? . . . . .	13
<i>Xingliang Chen, Antonija Mitrovic, and Moffat Mathews</i>	
Automatic Question Generation: From NLU to NLG. . . . .	23
<i>Karen Mazidi and Paul Tarau</i>	
Using Eye-Tracking to Determine the Impact of Prior Knowledge on Self-Regulated Learning with an Adaptive Hypermedia-Learning Environment. . . . .	34
<i>Michelle Taub and Roger Azevedo</i>	
Informing Authoring Best Practices Through an Analysis of Pedagogical Content and Student Behavior . . . . .	48
<i>Matthew Roy and Rohit Kumar</i>	
Timing Game-Based Practice in a Reading Comprehension Strategy Tutor. . .	59
<i>Matthew E. Jacovina, G. Tanner Jackson, Erica L. Snow, and Danielle S. McNamara</i>	
Evaluation of the Formal Models for the Socratic Method . . . . .	69
<i>Nguyen-Thanh Le and Nico Huse</i>	
Stealth Assessment in ITS - A Study for Developmental Dyscalculia. . . . .	79
<i>Severin Klingler, Tanja Käser, Alberto-Giovanni Busetto, Barbara Solenthaler, Juliane Kohn, Michael von Aster, and Markus Gross</i>	
Mastery-Oriented Shared Student/System Control Over Problem Selection in a Linear Equation Tutor. . . . .	90
<i>Yanjin Long and Vincent Aleven</i>	
Providing the Option to Skip Feedback in a Worked Example Tutor. . . . .	101
<i>Amruth N. Kumar</i>	

Tell Me How to Teach, I'll Learn How to Solve Problems. . . . .	111
<i>Noboru Matsuda, Nikolaos Barbalios, Zhengzheng Zhao, Anya Ramamurthy, Gabriel J. Stylianides, and Kenneth R. Koedinger</i>	
Scale-Driven Automatic Hint Generation for Coding Style . . . . .	122
<i>Rohan Roy Choudhury, Hezheng Yin, and Armando Fox</i>	
Estimating Individual Differences for Student Modeling in Intelligent Tutors from Reading and Pretest Data . . . . .	133
<i>Michael Eagle, Albert Corbett, John Stamper, Bruce M. McLaren, Angela Wagner, Benjamin MacLaren, and Aaron Mitchell</i>	
Building Pedagogical Models by Formal Concept Analysis . . . . .	144
<i>Giuseppe Fenza and Francesco Orciuoli</i>	
Predicting Learning from Student Affective Response to Tutor Questions. . . . .	154
<i>Alexandria K. Vail, Joseph F. Grafsgaard, Kristy Elizabeth Boyer, Eric N. Wiebe, and James C. Lester</i>	
Integrating Real-Time Drawing and Writing Diagnostic Models: An Evidence-Centered Design Framework for Multimodal Science Assessment . . . . .	165
<i>Andy Smith, Osman Aksit, Wookhee Min, Eric Wiebe, Bradford W. Mott, and James C. Lester</i>	
The Bright and Dark Sides of Gamification . . . . .	176
<i>Fernando R.H. Andrade, Riichiro Mizoguchi, and Seiji Isotani</i>	
Behavior Changes Across Time and Between Populations in Open-Ended Learning Environments . . . . .	187
<i>Brian Gauch and Gautam Biswas</i>	
Are Pedagogical Agents' External Regulation Effective in Fostering Learning with Intelligent Tutoring Systems? . . . . .	197
<i>Roger Azevedo, Seth A. Martin, Michelle Taub, Nicholas V. Mudrick, Garrett C. Millar, and Joseph F. Grafsgaard</i>	
Intervention-BKT: Incorporating Instructional Interventions into Bayesian Knowledge Tracing . . . . .	208
<i>Chen Lin and Min Chi</i>	
<b>Short Papers</b>	
"i-Read": A Collaborative Learning Environment to Support Students with Low Reading Abilities . . . . .	221
<i>Nizar Omheni and Ahmed Hadj Kacem</i>	

Integrating Support for Collaboration in a Computer Science Intelligent Tutoring System . . . . . 227  
*Rachel Harsley, Barbara Di Eugenio, Nick Green, Davide Fossati, and Sabita Acharya*

Wheel-Spinning in a Game-Based Learning Environment for Physics . . . . . 234  
*Thelma D. Palaoag, Ma. Mercedes T. Rodrigo, Juan Miguel L. Andres, Juliana Ma. Alexandra L. Andres, and Joseph E. Beck*

Using Multi-level Modeling with Eye-Tracking Data to Predict Metacognitive Monitoring and Self-regulated Learning with CRYSTAL ISLAND . . . . . 240  
*Michelle Taub, Nicholas V. Mudrick, Roger Azevedo, Garrett C. Millar, Jonathan Rowe, and James Lester*

The Mobile Fact and Concept Training System (MoFaCTS). . . . . 247  
*Philip I. Pavlik Jr., Craig Kelly, and Jaclyn K. Maass*

Coordinating Knowledge Integration with Pedagogical Agents: Effects of Agent Gaze Gestures and Dyad Synchronization . . . . . 254  
*Yugo Hayashi*

An Investigation of Conversational Agent Interventions Supporting Historical Reasoning in Primary Education. . . . . 260  
*Stergios Tegos, Stavros Demetriadis, and Thrasyvoulos Tsiatsos*

Impact of Question Difficulty on Engagement and Learning. . . . . 267  
*Jan Papoušek, Vít Stanislav, and Radek Pelánek*

Are There Benefits of Using Multiple Pedagogical Agents to Support and Foster Self-Regulated Learning in an Intelligent Tutoring System? . . . . . 273  
*Seth A. Martin, Roger Azevedo, Michelle Taub, Nicholas V. Mudrick, Garrett C. Millar, and Joseph F. Grafsgaard*

Can Peers Rate Reliably as Experts in Small CSCL Groups? . . . . . 280  
*Ioannis Magnisalis, Stavros Demetriadis, and Pantelis M. Papadopoulos*

Peer Review in Mentorship: Perception of the Helpfulness of Review and Reciprocal Ratings . . . . . 286  
*Oluwabunmi Adewoyin, Roberto Araya, and Julita Vassileva*

Motivational Gamification Strategies Rooted in Self-Determination Theory for Social Adaptive E-Learning. . . . . 294  
*Lei Shi and Alexandra I. Cristea*

Adaptive Training of the Metacognitive Skill of Knowledge Monitoring in Intelligent Tutoring Systems . . . . . 301  
*Tiago Roberto Kautzmann, Talvany Carlotto, and Patrícia A. Jaques*

Persuading an Open Learner Model in the Context of a University Course: An Exploratory Study . . . . .	307
<i>Blandine Ginon, Clelia Boscolo, Matthew D. Johnson, and Susan Bull</i>	
Blinded by Science?: Exploring Affective Meaning in Students' Own Words . . . . .	314
<i>Sarah E. Schultz, Naomi Wixon, Danielle Alessio, Kasia Muldner, Winslow Burlison, Beverly Woolf, and Ivon Arroyo</i>	
A Framework for Parameterized Design of Rule Systems Applied to Algebra . . . . .	320
<i>Eric Butler, Emina Torlak, and Zoran Popović</i>	
Cognitive Tutors Produce Adaptive Online Course: Inaugural Field Trial . . . .	327
<i>Noboru Matsuda, Martin van Velsen, Nikolaos Barbalios, Shuqiong Lin, Hardik Vasa, Roya Hosseini, Klaus Sutner, and Norman Bier</i>	
Optimizing Pattern Weights with a Genetic Algorithm to Improve Automatic Working Memory Capacity Identification . . . . .	334
<i>Jason Bernard, Ting-Wen Chang, Elvira Popescu, and Sabine Graf</i>	
Stratified Learning for Reducing Training Set Size . . . . .	341
<i>Peter Hastings, Simon Hughes, Dylan Blaum, Patricia Wallace, and M. Anne Britt</i>	
Combining Worked Examples and Problem Solving in a Data-Driven Logic Tutor . . . . .	347
<i>Zhongxiu Liu, Behrooz Mostafavi, and Tiffany Barnes</i>	
NDLtutor: An Automated Conversational Agent to Facilitate Metacognitive Skills in Fully-Negotiated OLMs. . . . .	354
<i>Raja M. Suleman, Riichiro Mizoguchi, and Mitsuru Ikeda</i>	
Concept Maps Similarity Measures for Educational Applications. . . . .	361
<i>Carla Limongelli, Matteo Lombardi, Alessandro Marani, Filippo Sciarrone, and Marco Temperini</i>	
Can Adaptive Pedagogical Agents' Prompting Strategies Improve Students' Learning and Self-Regulation? . . . . .	368
<i>François Bouchet, Jason M. Harley, and Roger Azevedo</i>	
Automatic Extraction of Prerequisites Among Learning Objects Using Wikipedia-Based Content Analysis . . . . .	375
<i>Carlo De Medio, Fabio Gasparetti, Carla Limongelli, Filippo Sciarrone, and Marco Temperini</i>	

Using Electroencephalogram to Track Learner’s Reasoning in Serious Games . . . . . 382  
*Ramla Ghali, Claude Frasson, and Sébastien Ouellet*

Behavior and Learning of Students Using Worked-Out Examples in a Tutoring System . . . . . 389  
*Nick Green, Barbara Di Eugenio, Rachel Harsley, Davide Fossati, and Omar AlZoubi*

The Frequency of Tutor Behaviors: A Case Study . . . . . 396  
*Vincent Aleven and Jonathan Sewall*

Towards an Effective Affective Tutoring Agent in Specialized Education . . . . . 402  
*Aydée Liza Mondragon, Roger Nkambou, and Pierre Poirier*

Embedding Intelligent Tutoring Systems in MOOCs and e-Learning Platforms . . . . . 409  
*Vincent Aleven, Jonathan Sewall, Octav Popescu, Michael Ringenberg, Martin van Velsen, and Sandra Demi*

Using Cloze Procedure Questions in Worked Examples in a Programming Tutor . . . . . 416  
*Amruth N. Kumar*

The Effect of Friendship and Tutoring Roles on Reciprocal Peer Tutoring Strategies . . . . . 423  
*Michael A. Madaio, Amy Ogan, and Justine Cassell*

CRISTAL: Adapting Workplace Training to the Real World Context with an Intelligent Simulator for Radiology Trainees . . . . . 430  
*Hope Lee, Amali Weerasinghe, Jayden Barnes, Luke Oakden-Rayner, William Gale, and Gustavo Carneiro*

**Posters**

A System for Gamifying Ubiquitous Learning Situations Supported by Multiple Technologies . . . . . 439  
*Alejandro Ortega-Arranz, Juan A. Muñoz-Cristóbal, Alejandra Martínez-Monés, Miguel L. Bote-Lorenzo, and Juan I. Asensio-Pérez*

Combining Speech-Acts and Socio-Historical Theories to Monitor and Analyze the Cognitive Evolution of Students on VLE’s Records . . . . . 441  
*Gustavo Schwarz, João C. Gluz, and Liliana M. Passerino*

Incorporating Student Choice in E-learning . . . . . 443  
*Avi Segal, Naor Guetta, Amir Taboul, Guy Shani, and Ya’akov (Kobi) Gal*

Towards a Mobile Python Tutor: Understanding Differences in Strategies Used by Novices and Experts . . . . .	447
<i>Geela Fabric, Antonija Mitrovic, and Kouros Neshatian</i>	
Modeling Negative Affect of Novice Programming Students using Keyboard Dynamics and Mouse Behavior . . . . .	449
<i>Larry A. Veal and Ma. Mercedes T. Rodrigo</i>	
What Is More Important for Student Modeling: Domain Structure or Response Times? . . . . .	451
<i>Jiří Řihák and Radek Pelánek</i>	
Evaluating Affect in a Learning Environment for Java . . . . .	453
<i>Ramón Zatarain-Cabada, María Lucía Barrón-Estrada, Francisco González-Hernández, and Carlos A. Reyes-García</i>	
Implicit Social Networks for Social Recommendation of Scholarly Papers . . .	455
<i>Shaikhah Alotaibi and Julita Vassileva</i>	
A Context-Based Similarity Algorithm for Enhancing Learning Scenarios Reuse . . . . .	458
<i>Mariem Chaabouni, Mona Laroussi, Claudine Piau-Toffolon, Christophe Choquet, and Henda Ben Ghezala</i>	
Towards the Recommendation of Resources in Coursera . . . . .	461
<i>Carla Limongelli, Matteo Lombardi, and Alessandro Marani</i>	
Triangle Block Model for Bridging Conceptual Representation to Numerical Representation in Arithmetic Word Problems: A Brief Report of Practical Use by Fourth Grade Students . . . . .	464
<i>Tsukasa Hirashima, Kazutoshi Furukubo, Yusuke Hayashi, Sho Yamamoto, and Kazushige Maeda</i>	
POLARISQL: An Online Tutoring System for Learning SQL Language . . . .	467
<i>Soraya Chachoua, Jamal Malki, and Pascal Estraillier</i>	
Empirical Evaluation of Intelligent Tutoring Systems with Ontological Domain Knowledge Representation: A Case Study with Online Courses in Higher Education . . . . .	469
<i>Ani Grubišić, Slavomir Stankov, Branko Žitko, Suzana Tomaš, Emil Brajković, Tomislav Volarić, Daniel Vasić, and Ines Šarić</i>	
Adaptive Testing by Bayesian Networks with Application to Language Assessment . . . . .	471
<i>Francesca Mangili, Claudio Bonesana, Alessandro Antonucci, Marco Zaffalon, Elisa Rubegni, and Loredana Addimando</i>	

When the Going Gets Tough...: Challenge, Emotions, and Difference of Perspective . . . . . 474  
*Naomi Wixon, Sarah Schultz, Danielle Alessio, Kasia Muldner, Winslow Burlesson, Beverly Woolf, and Ivon Arroyo*

Lost in Springdale: An Interactive Narrative for Adult Literacy Learners . . . . 476  
*Amy M. Johnson, Matthew E. Jacovina, G. Tanner Jackson, Elizabeth L. Tighe, and Danielle S. McNamara*

An Investigation of Learner’s Actions in Problem-Posing Activity of Arithmetic Word Problems . . . . . 478  
*Ahmad Afif Supianto, Yusuke Hayashi, and Tsukasa Hirashima*

Toward a Trace-Based PROMETHEE II Method to Answer “What Can Teachers Do?” in Online Distance Learning Applications. . . . . 480  
*Hoang Nam Ho, Mourad Rabah, Samuel Nowakowski, and Pascal Estraillier*

Enhancing Student Modeling for Collaborative Intelligent Tutoring Systems . . . . . 485  
*Jennifer K. Olsen, Vincent Aleven, and Nikol Rummel*

Toward Embodied Game-Based Intelligent Tutoring Systems . . . . . 488  
*Ivon Arroyo, Yuting Liu, Naomi Wixon, and Sarah Schultz*

Towards Computer-Assisted Curricula Design Using Probabilistic Graphical Models . . . . . 491  
*Waleed Alsanie, Issa Alkurtass, and Abdullah Al-Hamoud*

Predicting Spontaneous Facial Expressions from EEG . . . . . 494  
*Mohamed S. Benlamine, Maher Chaouachi, Claude Frasson, and Aude Dufresne*

An Empirical Evaluation of Learning Style and Knowledge Level Adaptation . . . . . 498  
*Mohammad Alshammari, Rachid Anane, and Robert J. Hendley*

Text-Based Emotion Recognition Approach . . . . . 500  
*Mohammed Abdel Razek and Claude Frasson*

The Questions of Ethics in Learning Analytics . . . . . 502  
*Madeth May and Sébastien Iksal*

An Analysis of Feature Selection and Reward Function for Model-Based Reinforcement Learning . . . . . 504  
*Shitian Shen, Chen Lin, Behrooz Mostafavi, Tiffany Barnes, and Min Chi*



On the Evaluation of the Expert and the Learner Models of Logic-Muse Tutoring System . . . . .	506
<i>Roger Nkambou, Ange Adrienne Nyamen Tato, Janie Brisson, Clauvice Kenfack, Serge Robert, and Pamela Kissok</i>	
Tools for Improving Teachers' Daily Tasks: Does It Really Help? . . . . .	509
<i>Fábio Goulart Andrade, Júlia Marques Carvalho da Silva, and Maurício Covolan Rosito</i>	
A Brief Overview of Logic-Muse, an Intelligent Tutoring System for Logical Reasoning Skills. . . . .	511
<i>Clauvice Kenfack, Roger Nkambou, Serge Robert, Ange Adrienne Nyamen Tato, Janie Brisson, and Pamela Kissok</i>	
Pilot Study with RALL-E: Robot-Assisted Language Learning in Education . . .	514
<i>Ning Wang and W. Lewis Johnson</i>	
Adapting Exercise Selection to Learner Self-esteem and Performance . . . . .	517
<i>Juliet Okpo, Matt Dennis, Kirsten Smith, Judith Masthoff, and Nigel Beacham</i>	
Do Summaries Support Learning from Post-problem Reflective Dialogues?. . . .	519
<i>Sandra Katz, Patricia Albacete, and Pamela Jordan</i>	
Social Interaction with Intelligent Tutoring Systems: An Investigation of Power and Related Affect . . . . .	521
<i>Katharina Roetzer</i>	
Efficiency vs. Immersion: Interface Design Trade-offs for an Exploratory Learning Environment . . . . .	523
<i>Toby Dragon, Mark Floryan, Grayson Wilkins, and Thomas Sparks</i>	
Dynamic Generation of Dilemma-Based Situations in Virtual Environments . . .	526
<i>Azzeddine Benabbou, Domitile Lourdeaux, and Dominique Lenne</i>	
<b>Young Researchers Track</b>	
An Implementation Architecture for Scenario-Based Simulations. . . . .	531
<i>Raja Lala, Johan Jeurig, and Jordy van Dortmund</i>	
A Student-Directed Immersive Intelligent Tutoring System for Language Learning . . . . .	534
<i>Jun Seong Choi and Jong H. Park</i>	
How to Present Example-Based Support Adaptively in Intelligent Tutoring Systems . . . . .	538
<i>Xingliang Chen, Antonija Mitrovic, and Moffat Mathews</i>	

The Automatic Generation of Knowledge Spaces from Problem Solving Strategies . . . . . 541  
*Ivica Milovanović and Johan Jeuring*

Using Multi-Channel Data to Assess, Understand, and Support Affect and Metacognition with Intelligent Tutoring Systems. . . . . 543  
*Michelle Taub and Roger Azevedo*

AMNESIA, a Dynamic Environment for Progressive Assessment of Cognitive Functions. . . . . 545  
*Asma Ben Khedher and Claude Frasson*

Comparisons of Different Types of Feedback of Linear Equation Aide (LEA): A Mobile Assisted Learning Application on Linear Equations . . . . . 547  
*Rex P. Bringula, Jan Sepli De Leon, Bernadette Anne Pascual, Kharl John Rayala, Kevin Sendino, and Marc Rodin Ligas*

**Author Index** . . . . . 549

# **Full Papers**

# Understanding Procedural Knowledge for Solving Arithmetic Task by Externalization

Kazuhisa Miwa<sup>1(✉)</sup>, Hitoshi Terai<sup>2</sup>, and Kazuya Shibayama<sup>1</sup>

<sup>1</sup> Nagoya University, Nagoya 464-8601, Japan  
miwa@is.nagoya-u.ac.jp

<sup>2</sup> Kinki University, Iizuka 820-8555, Japan

**Abstract.** Students build cognitive models for solving a crypt-arithmetic task in a learning environment that enables them to formally describe various types of procedural knowledge in a group learning setting in which each student is allowed to refer to the procedural rules described by the other group members. Experimental evaluation showed that: (1) three-quarters of participants successfully constructed valid models with the system, and (2) participants learned to describe procedural knowledge more precisely not only for the training task (crypt-arithmetic task) but also for a transfer task (bug identification for a multi-column subtraction problem).

**Keywords:** Procedural knowledge · Cognitive models · Externalization

## 1 Introduction

In cooperation with the experimental approach, the model-based approach is a primary methodology in cognitive science. Cognitive scientists have used computational models as research tools for understanding the human mind. The authors have examined functions of cognitive modeling as a learning tool, and proposed the learning by the creating cognitive models paradigm [8]. Fum et al. indicated three advantages of computational cognitive modeling: clarity and completeness, better exploration and evaluation, and serendipity and emergence [6]. We believe that these functions may provide students the opportunity to learn more about human cognitive information processing.

Previous studies have confirmed that creating cognitive models improves learners' theory-based thinking. The studies revealed that students more actively explained experimental data from the theoretical perspectives by creating cognitive models through simulating the experimental results [9, 15]. Another benefit of learning by creating cognitive models, i.e., active construction of mental models, was also confirmed [7]: people tend to construct a mental model of an object they understand. Acquiring sophisticated mental models is a key issue in both natural and social science education [5, 14]. A mental model is a structural, behavioral, or functional analogous representation of a real-world or imaginary situation, event, or process [12]. A mental model can be manipulated and draw

expectations on target phenomenon; thus, allowing people to predict hypothesized situations by such mental simulations.

There are many trials for improving students' mental model construction in natural science domains, but only few in the psychology domain [14]. An approach for acquiring mental models of human mental operations, i.e., cognitive information processing, is to identify the procedural knowledge used in solving a task. For example, when mathematics teachers infer the mental models of students' solving an arithmetic task, they are required to identify procedural rules that students utilize when solving the task [3]. We call such mental models rule-based mental models.

In a preceding study with two class practices for undergraduates and graduates, participants were required to construct a computational running model for solving subtraction problems and then develop a bug model that simulated other students' arithmetic errors [7]. Analyses indicated that by creating cognitive models, participants learned to identify buggy procedures that produced systematic errors and to predict expected erroneous answers. These results support the claim that building computational cognitive models enhances the participants' construction of rule-based mental models, and their mental simulations by operating the mental model. However this benefit emerges only in the students who successfully constructed the computational subtraction model. Half of the students were not able to program the model; therefore the benefit of the approach was limited.

The current study aims to develop a learning environment wherein students more easily construct computational models, and hence expand the benefit of the approach. In the preceding study, students programmed rule-based models on the production system architecture. A web-based production system architecture for novice users, called DoCoPro, was developed based on the server and client model for educational use [10]. Some students face difficulties in model construction using such a general production system architecture. Therefore, we developed a training environment wherein students more easily describe procedural rules, and examine the validity of the rules while confirming those operations.

Another purpose of developing the learning environment in the current study is to have students experience the model-based approach in cognitive science, and understand various advantages of the approach. One important component of human problem solving is that human inner (mental) functions and externally observable behaviors are tightly connected. Slight changes in procedural knowledge, such as a lack of specific knowledge, strongly influence problem-solving paths [11, 13]. In the learning environment, students hypothesize a set of procedural knowledge and examine what problem solving path emerges from the set. Then, students modify the set by revising, removing, and adding some of the rules as procedural knowledge, and again observe what external changes in behavior emerge based on the changes of inner functions. This type of design-and-test process enhances students' understandings of the nature of human problem solving with the advantages of the model-based approach in cognitive science.

## 2 Task

The task used in our study is a crypt-arithmetic task. In this study, we propose an environment wherein students learn procedural knowledge to perform the task while building a computational model. The following is an example problem:

$$\begin{array}{r}
 \text{DONALD} \quad \text{D}=5 \\
 +\text{GERALD} \\
 \text{-----} \\
 \text{ROBERT}
 \end{array}$$

The problem is *prima facie* simple; however, the cognitive information processing for its solution is relatively complex. In fact, the multiple types of procedural knowledge are used during the solution processes. The following are some examples.

- **Numeral processing:** If a column is  $x + y = z$ , and both  $x$  and  $y$  are known, then we can infer  $z$  by adding  $x$  and  $y$ . For example, in the rightmost column, we know  $D$  equals 5; therefore, 0 is assigned to letter  $T$  by applying this procedure.
- **Specific numeral processing:** If a column is  $x + y = x$ , then we can infer that  $y$  equals 0 or 9. For example, in the fifth column, we obtain that  $E$  equals 0 or 9 independently, without any other information.
- **Parity processing:** If a column is  $x + x = y$ , and we have a carry from the right column, then we can infer that  $y$  is an odd number. For example, in the second column, we obtained a carry by the inference in the first (i.e., rightmost) column; therefore, we conclude that  $R$  is an odd number.
- **Inequality processing:** If a column is  $x + y = z$ , and no carry is sent to the left column, then we can infer that  $z$  is greater than  $x$  (or  $y$ ). For example, in the sixth column, we know that  $D$  equals 5, and no carry is sent to the left column; therefore,  $R$  is greater than 5.

University students easily understand such procedural knowledge sets if they are given; however, they may face challenges finding the knowledge by themselves and externalizing it while solving the problem.

## 3 Learning System

We developed a learning environment to enable students to find and formally describe various types of procedural knowledge while solving problems. A distinctive feature of the environment is the group learning setting wherein three group members (in some cases, two group members) collaboratively construct their individual model. Specifically, each student is allowed to build his/her model while referring to rules described by other group members.

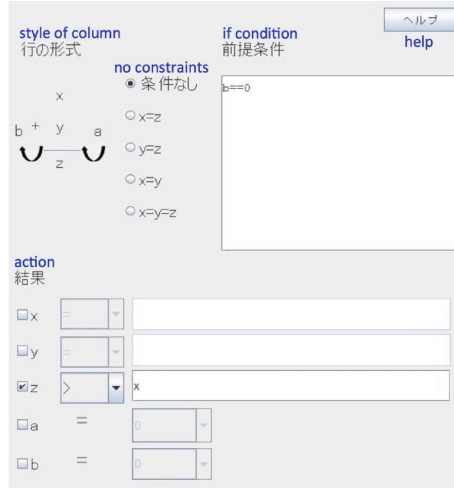


Fig. 1. An example screenshot of the knowledge editor.

### 3.1 Knowledge Editor

The system comprises two modules: the knowledge editor and problem-solving simulator. First, students externalize a set of procedural knowledge, i.e., describing rules, for solving crypt-arithmic tasks with the knowledge editor.

Figure 1 demonstrates an example screenshot of the knowledge editor wherein the rule of inequality processing is described, namely, If a column is  $x + y = z$ , and no carry is sent to the left column ( $b == 0$  in the figure), then we can infer that  $z$  is greater than  $x$ .

### 3.2 Problem Solving Simulator

The problem-solving simulator is mounted on the learning system. The problem solver that simulates behavior has the potential to perform an exhaustive search for the assignments of digits to letters. Specifically, it selects one of the letters that has not been determined and systematically assigns each digit to a letter. If a contradiction is found in the process of inference, another assignment is tested. If the problem solver has no procedural knowledge, it is impossible to derive the solution because the problem space spreads exhaustively. Students are required to give the problem solver adequate procedural knowledge with the knowledge editor.

Figure 2 indicates an example screenshot of the problem-solving simulator, which presents a problem status (the assignment status of digits to letters) and further presents an inference status (a series for information processing step by step). A list of rules installed for the problem solver is presented on the right-hand side

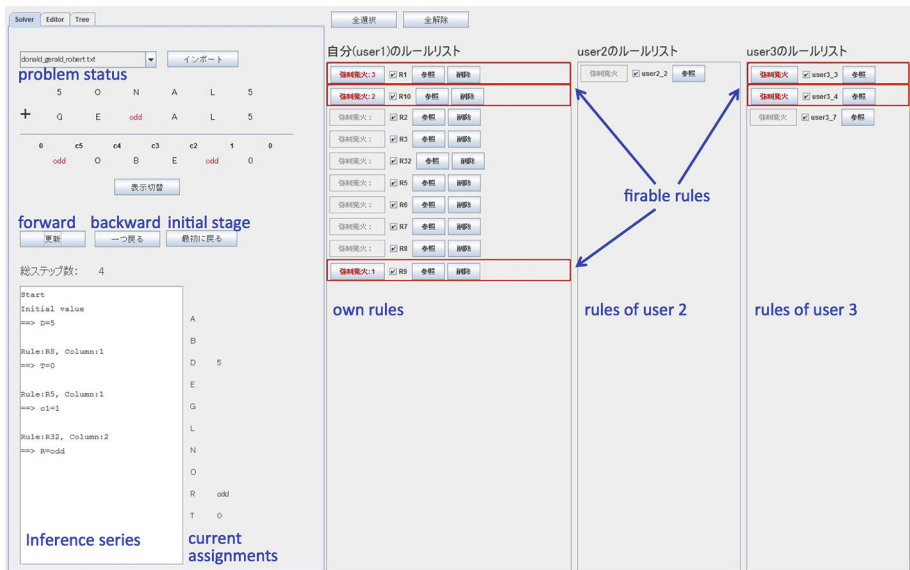


Fig. 2. An example screenshot of the problem-solving simulator.

of the window. Group members can refer the rules provided by other members. Rules that can fire at a specific problem solving step are marked by bold red lines. In this case, five rules (three own rules and two others' rules) are available. The conflict resolution mechanism is simple, and the most specific rule that provides the most specific inference result has priority for firing. Students can test any rule by forcibly firing and confirm the resulting inferences. Students can copy other members' rules to their own model, and revise those for their own use. Moreover, students can modify the model very easily. For example, if we delete the check from each item of the list, students can simulate the behavior of the problem solver with that knowledge excluded.

The system also presents the problem solver's behavior, represented as a search tree of problem-solving processes. Students can confirm inference steps one by one, forwarding the inference by clicking the inference button. At any point of the problem-solving process, students can install, delete, or revise knowledge using the editor and restart the inference from the problem-solving point.

The system can simulate a variety of problem-solving processes. For example, the complete problem solver arrives at the solution from approximately 21 to 42 steps. However, if the specific rule for the fifth column ( $O + E = O$ ), namely, If a column is  $x + y = x$ , then we can infer that  $y$  equals 0 or 9, is excluded from the knowledge set, then the problem solver requires more than 150 steps for a solution using the trial-and-error method.



## 4 Evaluation

### 4.1 Participants and Procedures

Participants in the practice included 36 undergraduates from Nagoya University. Ten groups comprising three members and three groups comprising two members were constructed. In the initial phase for one hour, they learned how to manage the knowledge editor and operate the problem-solving simulator. Specifically, participants were given an example problem:  $MEST + BADE = MASER$ ; they installed seven pieces of procedural knowledge for solving the given problem with the tutor's guidance, and they simulated behavior at each stage of the construction process.

After the instruction phase, participants performed pretests. After 10 min of rest, in the 70-min-long training phase, they were given the problem:  $DONALD + GERALD = ROBERT$ , and they, by themselves, were required to find a procedural knowledge set for the solution, install it in the problem solver with the knowledge editor, and construct a model. In the first part of the training phase, lasting 30 min, they built their model individually. Then in the second part of the training phase, lasting 40 min, they revised their model in the group setting wherein they were allowed to refer to other group members' rules. In the final phase, they performed posttests that consisted of equivalent problems to the pretests.

### 4.2 Pretests and Posttests

To examine whether students learn to construct rule-based mental models through trainings for externalizing procedural knowledge with our system, we conducted two tests for evaluation: review and transfer tests.

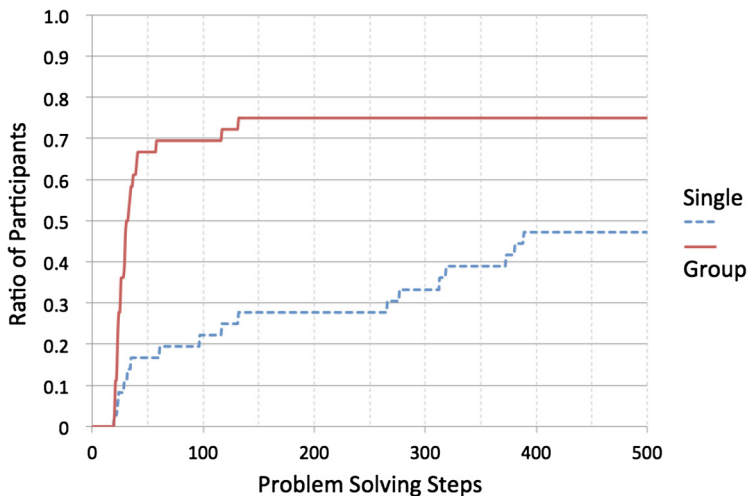
**Review Test.** Participants were presented with an example crypt-arithmetic task:  $AB + CB = DE$ . They were required to describe a rule applicable to the initial stage of problem solving, and describe it both with natural languages and the format of the system. The expected rule is: "If a column is  $x + x = y$  and a carry from the right column is zero, then we can infer that  $y$  is an even number."

**Transfer Test.** Participants were presented with multi-column subtraction problems and their wrong answers. Two cases were presented:  $9008 - 3149 = 3969$  and  $806303 - 182465 = 623748$ . In the identification task, participants were required to infer erroneous procedures and describe those in natural language. The error in the problem was as follows: If the top digit of the focused column is zero, then reach a possible column to the left across zero columns from which a carry is borrowed and return back across zero columns to the current column at which the answer is required. The other task was the replication task, wherein the participants were required to predict wrong answers drawn by the buggy procedures identified in the identification task. Two problems were used:  $708 - 139$  and  $900600803 - 123732349$ ; the predicted answers were 479 and 587778364, respectively.

### 4.3 Result

The following is a representative model construction process. During the simulation process, participants encountered a crucial stage of problem solving and hypothesized some procedural knowledge required for processing that specific stage of problem solving. They tried to provide the problem solver with the procedural knowledge, but usually, they initially failed in the installation. They noticed the failure by forwarding the problem solving by one step and confirming that the expected result was not obtained. Through the trial-and-error processes, once they accurately installed the knowledge set and passed through the crucial problem-solving stage, they forwarded the inference process and faced another specific stage of problem solving. They again tried to identify the procedural knowledge required for the next stage.

**Problem Solving Steps.** Our initial interest is at what rate and to what degree the participants accurately identified the procedural knowledge set and successfully had the problem solver achieve the solution. Figure 3 indicates the class results: The horizontal axis indicates problem-solving steps, and the vertical axis indicates the ratio of participants who constructed the model that reached the solution by the problem-solving step indicated on the horizontal axis. The lower line shows the performance of models constructed at the end of the individual model construction phase; and the upper line shows the performance at the end of the group model construction phase. Figure 3 indicates that, throughout group activities, 75% of participants constructed models that solved the problem within 150 steps.



**Fig. 3.** Percentage of undergraduate participants who constructed successful models for problem solving.

**Table 1.** Results of review test.

	Natural language		System format	
	Correct	Incorrect	Correct	Incorrect
Pre	3	33	2	34
Post	21	15	16	20

**Table 2.** Result of identification task in transfer test.

	Correct	Incorrect
Pre	10	26
Post	22	14

**Table 3.** Result of replication task in transfer test. SE in parenthesis.

Pretest	Posttest
0.75 (0.13)	1.28 (0.13)

**Review Test.** Table 1 shows the results of the review test. The exact chi-test shows that the number of successful participants who accurately described the target rule increased from the pretests to posttests ( $p < 0.01$ , two-tailed in language;  $p < 0.01$ , two-tailed in system format).

**Transfer Test.** Tables 2 and 3 show the results of the transfer test. Table 2 shows that the number of successful participants who accurately describe that the buggy rule in the identification task increased ( $p < 0.01$ , two-tailed). In addition, Table 3 shows the transition of the score of the replication task whose full mark was two, thus showing that the score significantly increased from the pretests to the posttests ( $F(1, 35) = 16.74$ ,  $p < 0.01$ ).

## 5 Discussion and Conclusions

We developed a learning environment to enable students to spontaneously find and formally describe various types of relevant procedural knowledge when solving crypt-arithmetic tasks. Our experiment found that three-fourths of participants constructed valid models with our system, and appeared to succeed in identifying and externalizing procedural knowledge for solving such a relatively complex arithmetic task.

The rate of successful participants who programmed valid models reached a satisfactory level; however, nine participants, i.e., one-fourth of the total, still failed to build the models. Two reasons exist for the failure. The first reason is that seven of the unsuccessful participants installed one or more buggy rules that caused incorrect assignments; therefore, the search for assignments stopped

in the middle of problem solving. For these participants, notifications that alert them of buggy rules may be effective. Other two unsuccessful participants did not install sufficient rules to reach the solution. In the early stage of problem solving, thus no rules were found for application, leading the problem-solver to the trial-and-error search. The model did not reach the solution even after the number of problem solving steps exceeded 500. For such participants, our learning environment provides the mechanism that enables them to refer to the rules proposed by other group members. However, this function for group problem solving did not work for them.

The scores of both the review and transfer tests significantly increased from the pretests to post tests. These results support that learning with our system enhances the participants' construction of rule-based mental models and their performance for mental simulations for predicting results in hypothesized situations. These effects come from participants' meta-cognitive activities. Multiple approaches exist for enhancing meta-cognitive activities, such as instruction, verbalization, self-regulation, debriefing, and self-explanation [1, 2, 4].

In our practice, participants were required to externalize their procedural knowledge for problem solving. There are two difficulties in externalization: inaccessibility and ambiguity. First, procedural rules used for solving cryptarithmic tasks are not difficult for university students. In fact, when participants are presented with each rule, they easily understand the operations of the rule. However, some of them were not able to notice such rules as being applicable nor actually apply them, meaning that the knowledge was inaccessible, and second, accurately describing rules is a difficult task for many students. Even when students can apply a rule, they often face difficulties in describing it accurately. In a representative case, some conditions for rule firing are excluded from the rule description even though the rule's action is correctly described, thus meaning the knowledge is ambiguous. In our learning system, students are forced to explicitly externalize each rule, and confirm whether the expected behavior is observed while executing the problem-solving simulator. This design-execute-confirm cycle enhances participants' externalization activities, thereby resulting in positive effects on their meta-cognitive activities.

Finally, Fig. 3 indicates that group activities greatly improved the participants' model performance. This implies that the learning design based on interaction among members in our learning environment has substantial effects. We categorized the participants into three groups; high, the participants who were able to build a model that reached the solution within 150 steps during the single learning phase (the first 30 min in the learning phase); middle, those who build such a model during the group learning phase (the last 40 min); and low, those who were not able to build such a model. The ratio of rules copied from others to all the rules of his/her model is 0.009 in high, 0.154 in middle, and 0.080 in low. This implies that the middle-level participants referred to more rules from others, thus raising the standard of the class activities.

## References

1. Atkinson, R.K., Renkl, A., Merrill, M.M.: Transitioning from studying examples to solving problems: effects of self-explanation prompts and fading worked-out steps. *J. Educ. Psychol.* **95**(4), 774 (2003)
2. Azevedo, R., Hadwin, A.F.: Scaffolding self-regulated learning and metacognition-implications for the design of computer-based scaffolds. *Instruct. Sci.* **33**(5), 367–379 (2005)
3. Brown, J.S., Burton, R.R.: Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Sci.* **2**, 155–192 (1978)
4. Conati, C., Vanlehn, K.: Toward computer-based support of meta-cognitive skills: a computational framework to coach self-explanation. *Int. J. Artif. Intell. Educ. (IJAIED)* **11**, 389–415 (2000)
5. De Jong, T., van Joolingen, W.R.: Scientific discovery learning with computer simulations of conceptual domains. *Rev. Educ. Res.* **68**, 179–201 (1998)
6. Fum, D., Missier, F.D., Stocco, A.: The cognitive modeling of human behavior: why a model is (sometimes) better than 10,000 words. *Cognitive Syst. Res.* **8**, 135–142 (2007)
7. Miwa, K., Kanzaki, N., Terai, H., Kojima, K., Nakaike, R., Morita, J., Saito, H.: Learning mental models of human cognitive processing by creating cognitive models. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M.F. (eds.) *AIED 2015. LNCS*, vol. 9112, pp. 287–296. Springer, Heidelberg (2015)
8. Miwa, K., Morita, J., Nakaike, R., Terai, H.: Learning through intermediate problems in creating cognitive models. *Interact. Learn. Environ.* **22**, 326–350 (2014)
9. Miwa, K., Morita, J., Terai, H., Kanzaki, N., Kojima, K., Nakaike, R., Saito, H.: Use of a cognitive simulator to enhance students' mental simulation activities. In: Trausan-Matu, S., Boyer, K.E., Crosby, M., Panourgia, K. (eds.) *ITS 2014. LNCS*, vol. 8474, pp. 398–403. Springer, Heidelberg (2014)
10. Miwa, K., Nakaike, R., Morita, J., Terai, H.: Development of production system for anywhere and class practice. In: *Proceedings of the 14th International Conference of Artificial Intelligence in Education*, pp. 91–99 (2009)
11. Miwa, K.: A cognitive simulator for learning the nature of human problem solving. *J. Jpn. Soc. Artif. Intell.* **23**(6), 374–383 (2008)
12. Nersessian, N.: *Creating Scientific Concepts*. MIT press, Cambridge (2008)
13. Newell, A., Simon, H.A.: *Human Problem Solving*. Prentice-Hall, Englewood Cliffs (1972)
14. Rutten, N., van Joolingen, W.R., van der Veen, J.T.: The learning effects of computer simulations in science education. *Comput. Educ.* **58**, 136–153 (2012)
15. Saito, H., Miwa, K., Kanzaki, N., Terai, H., Kojima, K., Nakaike, R., Morita, J.: Educational practice for interpretation of experimental data based on a theory. In: *Proceedings of 21th International Conference on Computers in Education*, pp. 234–239 (2013)

# Do Erroneous Examples Improve Learning in Addition to Problem Solving and Worked Examples?

Xingliang Chen<sup>(✉)</sup>, Antonija Mitrovic, and Moffat Mathews

Intelligent Computer Tutoring Group,  
University of Canterbury, Christchurch, New Zealand  
xingliang.chen@pg.canterbury.ac.nz,  
{tanja.mitrovic,moffat.mathews}@canterbury.ac.nz

**Abstract.** Learning from Problem Solving (PS), Worked Examples (WE) and Erroneous Examples (ErrEx) have all proven to be effective learning strategies. However, there is still no agreement on what kind of assistance (in terms of different learning activities) should be provided to students in Intelligent Tutoring Systems (ITSs) to optimize learning. A previous study [1] found that alternating worked examples and problem solving (AEP) was superior to using just one type of learning tasks. In this paper, we compare AEP to a new instructional strategy which, in addition to PS and WEs, additionally offers erroneous examples to students. The results indicate that erroneous examples prepare students better for problem solving in comparison to worked examples. Explaining and correcting erroneous examples also leads to improved debugging and problem-solving skills.

**Keywords:** Intelligent tutoring system · Worked examples · Erroneous examples · Assistance · Problem-solving · SQL-Tutor

## 1 Introduction

A worked example consists of a problem statement, its solution and additional explanations, and therefore provides a high level of assistance to students. WEs reduce the cognitive load on the student's working memory, thus allowing the student to learn faster and deal with more complex problems [2]. Previous research compared the effectiveness of learning from examples to unsupported problem solving [3, 4], and showed that WEs are beneficial for learning in well-structured domains. The benefits of WEs were demonstrated in many studies for novices, but problem solving was found to be superior to WEs for more advanced students [5]. The effects of Problem Solving only (PS), Worked-Examples only (WE), Worked-Examples/Problem-Solving pairs (WE-PS) and Problem-Solving/Worked-examples pairs (PS-WE) have been studied on novices [6]. The WE and WE-PS conditions resulted in significantly higher learning effectiveness compared to the PS and PS-WE conditions. However, van Gog [7] later claimed that the WE-PS and PS-WE conditions were not comparable, because the examples and problems should be identical within and across pairs. Consequently, she

employed an example-problem sequence (EP condition) and a problem-example sequence (PE condition) for learning. The students learned significantly more in the EP condition than in the PE condition.

In comparison to unsupported problem solving, ITSs provide adaptive feedback, hints and other types of help to students. Several recent studies investigated the effects of learning from WEs compared to learning from tutored problems solving (TPS) in ITSs; a few of those studies found no difference in learning gain but WEs resulted in shorter learning time [8–10]. Contrary to that, a study [1] conducted in SQL-Tutor, a constraint-based tutor that teaches database querying in SQL, found that students learned more from TPS than from WEs; furthermore, the best condition was alternating worked examples with problem solving (AEP), which presented isomorphic pairs of WE and TPS to students.

Several recent studies focused on erroneous examples, which provide incorrect solutions and require students to find and fix errors [11, 12]. Große and Renkl [12] investigated whether both correct and incorrect examples affect learning in the domain of probability. They found that erroneous examples were beneficial on far transfer for high prior knowledge students. Durkin and Rittle-Johnson [11] found that providing both WEs and ErrExs resulted in higher procedural and declarative knowledge in comparison to the WE only condition. They did not find any differences between novices and advanced students.

Surprisingly, there have not been many studies on the benefits of learning from erroneous examples with ITSs. Tsovaltzi et al. [13] investigated the effect of studying erroneous examples of fractions in an ITS. They found that erroneous examples with interactive help improved 6<sup>th</sup> grade students' metacognitive skills. Furthermore, 9<sup>th</sup> and 10<sup>th</sup> graders improved their problem solving skills and conceptual knowledge when using ErrEx with interactive help. Booth et al. [14] demonstrated that students who explained correct and incorrect examples significantly improved their post-test performance in comparison with those who only received WEs in the Algebra I Cognitive Tutor. Additionally, the ErrEx condition and the combined WE/ErrEx condition were beneficial for improving conceptual understanding of algebra, but not for procedural knowledge.

The goal of our study was to investigate the effects of using erroneous examples in addition to WEs and TPS in SQL-Tutor. Previously, the AEP condition was found to be superior to using WEs or TPS alone [1, 15]. In this study, we compared the best condition from that previous study, AEP, to a new instructional strategy which presented a fixed sequence of worked example/problem pairs and erroneous example/problem pairs (WPEP) to support learning. Our hypotheses are that the addition of erroneous examples to WEs and TPS would be beneficial for learning overall (H1), and that their effect would be more pronounced for advanced students (H2).

## 2 SQL-Tutor

For this study, we modified SQL-Tutor [16], a constraint-based ITS for teaching the Structured Query Language (SQL) by developing three distinct modes to correspond to TPS, WEs and ErrExs. Figure 1 shows the screenshot of the problem-solving interface we used in this study. The left pane shows the structure of the database schema, which

the student can explore to gain additional information about tables and their attributes, as well as to see the data stored in the database. The middle pane is the problem-solving environment. At the start of a problem, this pane shows only the input areas for the *Select* and *From* clauses; the student can click on the other clauses to get the input boxes for the remaining clauses as necessary. The right pane shows the feedback once the student submits his/her solution.

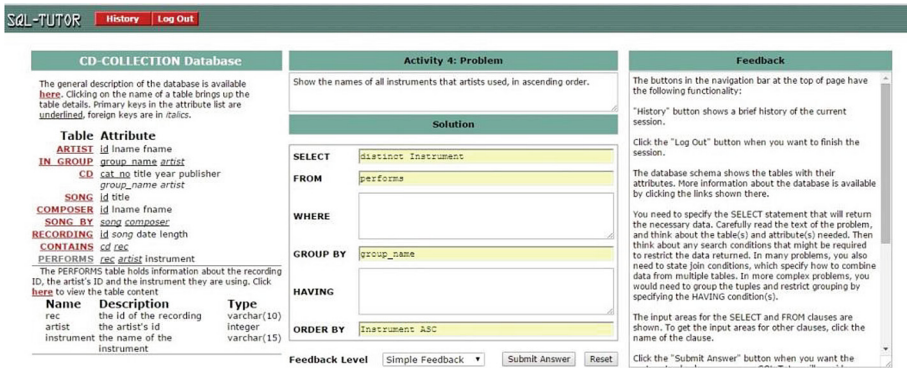


Fig. 1. The student interface of the problem-solving mode of SQL-Tutor

Figure 2 presents the screenshot of the WE mode. An example problem with its solution and explanation is provided in the center pane. A student can confirm that s/he has completed studying the example by clicking the *Continue* button.

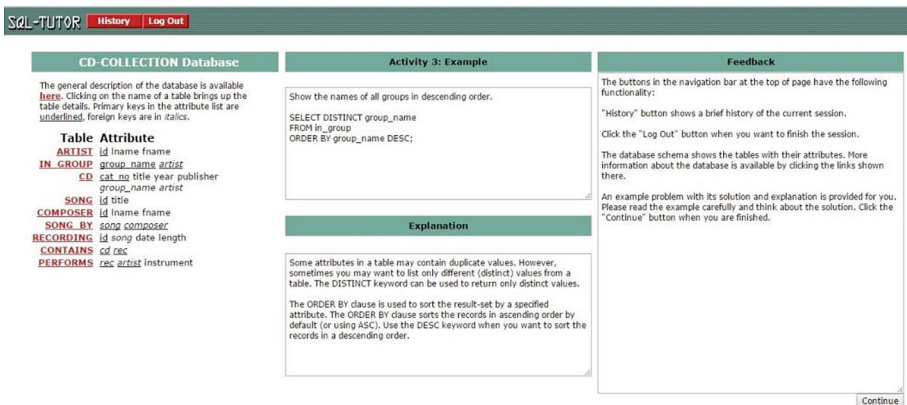


Fig. 2. The student interface of the worked example mode of SQL-Tutor

The ErrEx mode is illustrated in Fig. 3. An incorrect solution is provided for each problem, and the student’s task is to analyze the solution, find errors and correct them. The student can submit the solution to be checked by SQL-Tutor multiple times,



The screenshot shows the SQL-Tutor interface. At the top, there are links for 'History' and 'Log Out'. The main content is divided into three panels:

- CD-COLLECTION Database:** Contains a general description and a table with columns: CD (cat\_no, title, year, publisher, group\_name, artist), SONG (id, title), COMPOSER (id, name, name), RECORDING (id, song, date, length), CONTAINS (cd, cat), and PERFORMS (cat, artist, instrument).
- Activity 3: Erroneous Example:** Shows the task: 'Show the names of all groups in descending order.' The student's incorrect query is: `SELECT group_name FROM in_group`. The 'Solution' section shows the correct query: `SELECT group_name FROM in_group GROUP BY group_name ORDER BY group_name DESC`.
- Feedback:** Provides instructions on how to use the 'Submit Answer' and 'Reset' buttons and explains the feedback mechanism.

Fig. 3. The student interface of the erroneous-example mode of SQL-Tutor

similar to the problem-solving mode. In the situation illustrated in Fig. 3, the student has identified the SELECT clause as being incorrect, and is defining the new version of it. The student has also added the *Group by* and *Order by* clauses.

Previous research has shown the importance of self-explanation for learning [17, 18]. Providing Self-Explanation (SE) prompts is a common method to encourage students to self-explain. It was found in previous work that WEs help improve conceptual knowledge more than procedural knowledge, whereas problem solving results in higher levels of procedural knowledge [8, 19]. As a consequence, Najjar and Mitrovic [1] developed Conceptual-focused Self-Explanation (C-SE) prompts that support students to self-explain relevant domain concepts after problem solving, and Procedural-focused Self-Explanation (P-SE) prompts that supports students to self-explain solution steps after WEs. A C-SE prompt is presented after a problem is solved in order to aid the student in reflecting on the concepts covered in the problem they just completed (e.g. *What does DISTINCT in general do?*). On the other hand, a P-SE prompts are provided after WEs to assist learners in focusing on problem-solving approaches (e.g. *How can you specify a string constant?*). C-SE and P-SE prompts were used in the previous study [1] to increase learning. In order to keep our experimental design consistent with that of [1], our participants received C-SE prompts after problems, and P-SE prompts after WEs, to complement learning activities so that both conceptual and procedural knowledge is supported. Since ErrExs contain both properties of problems and WEs, we provided P-SE and C-SE prompts alternatively after ErrExs.

### 3 Experimental Design

The study was conducted with 60 students enrolled in an introductory database course at the University of Canterbury, in regular labs scheduled for the course (100 min long). Prior to the study, the students learned about SQL in lectures, and had one lab session. The version of SQL-Tutor used in this study had two conditions: Alternating Examples and Problems (AEP), the most effective learning condition from the previous study [15], and the experimental condition consisting of Worked example/Problem

pairs and Erroneous example/Problem pairs (WPEP). In both conditions, the order of tasks was the same, with the only difference being whether tasks were presented as problems to be solved, WEs or ErrExs. After providing informed consent, the participants were randomly assigned to either AEP or WPEP. The pre-test was administered online, followed by the 20 learning tasks. After completing all tasks, the participants completed the online post-test, which was similar in complexity and length to the pre-test. Figure 4 illustrates the study design.

AEP	WPEP
Pre-test	
20 problems and WEs (10 isomorphic pairs)	10 problems/WEs (5 isomorphic pairs), and 10 problems/ErrEx (5 isomorphic pairs), presented in alternation
Post-test	

**Fig. 4.** Study design with two conditions (AEP and WPEP)

## 4 Results

Our study was conducted at a time when the participants had assessment due in other courses they were taking. Since participation was voluntary, not all participants completed the study. Twenty-six students completed all activities and the post-test. In the following section, we present the results of analyses performed on the data collected for those 26 students (15 in the AEP and 11 in the WPEP condition).

More than half of the participants have not completed the study. Such a big attrition rate necessitated a further investigation. We compared the incoming knowledge (i.e. the pre-test scores) of the participants who completed or abandoned the study, in order to identify whether they were comparable or whether it was the weaker students who have not completed the study.

The pre/post-test consisted of 11 questions each. Questions 1–6 measured conceptual knowledge and were multi-choice or true-false questions (with the maximum of 6 marks). Questions 7–9 focused on procedural knowledge; question 7 was a multi-choice question (one mark), followed by a true-false question (one mark), while question 9 required the student to write a query for a given problem (4 marks). The last two questions presented incorrect solutions to two problems, and required the student to correct them, thus measuring debugging knowledge (6 marks). Therefore, the maximum mark on each test was 18.

The pre-test scores are given in Table 1. There were no significant differences between the two subsets of participants on overall pre-test scores. There were also no significant differences on the scores for declarative, procedural and debugging questions. Therefore, the 26 remaining participants are representative of the class.

**Table 1.** Pre-test scores (in %) for all students, and for participants who completed/abandoned the study (standard deviations shown in parentheses)

	All participants (60)	Completed (26)	Abandoned (34)
Overall	65.14 (14.09)	65.81 (13.14)	64.62 (14.96)
Conceptual	55.28 (17.76)	53.85 (17.19)	56.37 (18.36)
Procedural	81.67 (23.26)	85.26 (16.72)	78.92 (27.16)
Debugging	58.47 (23.19)	58.33 (24.15)	58.58 (22.79)

#### 4.1 Do the Conditions Differ on Learning Outcomes?

We used the Mann-Whitney U test to analyze the differences between the two conditions (Table 2). There was no significant difference between AEP and WPEP in both the pre-test and post-test scores. The students in both the AEP ( $p = .001$ ) and the WPEP condition ( $p = .003$ ) improved significantly between pre-test and post-test, as confirmed by a statistically significant median increase identified by the Wilcoxon signed-rank test (shown in the *Improvement* row of Table 2). The effect sizes (Cohen's  $d$ ) are high for both groups, with the WPEP group having a higher effect size. For both groups, the pre-test and post-test scores are positively correlated, but only the correlation for AEP is significant.

**Table 2.** Basic statistics for the two conditions

	AEP (15)	WPEP (11)
Pre-test (%)	67.22 (15.37), med = 66.67	63.89 (9.7), med = 61.11
Post-test (%)	91.11 (12.92), med = 97.22	93.94 (6.67), med = 94.44
Improvement	$W = 120, p < .005, d = 1.29$	$W = 66, p < .005, d = 1.73$
Pre/post-test correlation	$r = .58, p < .05$	$r = .52, ns$
Interaction time (min)	65.64 (16.96)	67.09 (10.22)

On average, the participants spent 66 min interacting with the learning tasks. There was no significant difference on the total interaction time between the two conditions. The students in both groups solved the same number of problems (10). The AEP group had 10 WEs, while the WPEP group had five WEs and five ErrExs. We expected erroneous examples to take more time in comparison to WEs, but the difference was not significant.

**Table 3.** Detailed scores on pre/post-tests

Group	Questions	Pre-test %	Post-test %	W, p
AEP (15)	Conceptual	57.78 (17.67)	94.44 (10.29)	120, .001**
	Procedural	80.56 (18.28)	97.78 (5.86)	36, .011**
	Debugging	63.33 (24.56)	81.11 (29.46)	73, .054*
WPEP (11)	Conceptual	48.48 (15.73)	91 (8.7)	66, .002**
	Procedural	91.67 (12.36)	97.73 (7.54)	ns
	Debugging	51.51 (22.92)	93.18 (15.28)	45, .007**

Table 3 shows the scores on different question types. There were no significant differences on pre-test scores for the two conditions. In the AEP condition, there were significant differences between pre- and post-test scores on conceptual and procedural questions, as well as a marginally significant difference on the score for debugging questions. In the WPEP condition, the students' scores on conceptual and debugging questions increased significantly between pre- and post-test, but there was no significant difference on the scores on procedural questions. The WPEP group started with a very high level of procedural knowledge, and that explains no significant difference on this type of questions.

In order to identify whether the two conditions affected students' problem solving differently, we analyzed the log data. As explained previously, ten learning tasks were problems to be solved. Table 4 reports the number of attempts (i.e. solution submission), as well as the number of errors (i.e. the number of violated constraints) for the ten problems. Overall, the AEP group made significantly more attempts ( $U = 37.5$ ,  $p = .018$ ) and more mistakes ( $U = 44$ ,  $p = .047$ ) on the ten problems.

**Table 4.** Analysis of attempts and errors for the two conditions

	All problems		Problems 4, 8, 12, 16, 20		Problems after WEs	
	Attempts	Errors	Attempts	Errors	Attempts	Errors
AEP	4.54 (1.7)	12.87 (8.31)	5.67 (2.14)	17.44 (11.12)	3.41 (1.89)	8.29 (8.09)
WPEP	3.08 (1.06)	7.73 (6.75)	3.49 (1.43)	9.64 (10.47)	2.67 (1.21)	5.82 (7.1)
p	<.02**	<.05**	<.01**	<.05**	ns	ns

The table also reports the two measures for various subsets of problems, identified on the basis of the previous learning task. We wanted to investigate whether correct and erroneous examples prepare students differently for problem solving. Problems 4, 8, 12, 16 and 20 were presented in the WPEP condition after ErrEx, whereas in the AEP condition after WEs. For those five problems, there were significant differences between the two conditions on both attempts ( $U = 30$ ,  $p = .005$ ) and errors ( $U = 41$ ,  $p = .032$ ). On the other hand, problems 2, 6, 10, 14 and 18 were presented to both conditions after WEs. For those problems, we found no significant differences between the two groups on either attempts or errors on this subset of problems. These findings show that erroneous examples prepare students better for problem solving in comparison to worked examples, which confirms our hypothesis H1. This is important, as some of the previous studies (as discussed in the Introduction) have found that worked examples are superior to other types of learning tasks.

## 4.2 Comparing Novices and Advanced Students

We were also interested in the effectiveness of the two conditions on students with different levels of pre-existing knowledge. We classified students into novices and advanced students based on their pre-test scores (Table 5). The participants whose

pre-test scores are lower than 66 % (the overall median pre-test score for our sample) are considered to be novices, and the rest as advanced students.

The Mann-Whitney U test revealed no significant differences between novices/advanced students in the two conditions, on pre- and post-test scores. The Wilcoxon signed-rank test showed that novices and advanced students in both conditions improved significantly between the pre- and post-test ( $p < 0.05$ ). A deeper analysis of the pre/post-test scores revealed that in the WPEP condition, the score for debugging questions improved significantly for novices ( $p < .05$ ) and marginally significantly for advanced students ( $p = .059$ ), while only advanced students from the AEP condition improved their score on debugging questions ( $p = .01$ ). The novice AEP students did not improve their debugging knowledge. The normalized gain on debugging questions only for novices from the AEP condition was 0.15 ( $sd = 0.71$ ), while for novices from the WPEP group it was 0.76 (0.3); the difference is marginally significant ( $U = 29.5$ ,  $p = .063$ ,  $d = 0.96$ ). The fact that both advanced and novice WPEP students improved on debugging questions rejects our second hypothesis; contrary to our expectations, both novices and advanced students benefitted from ErrEx.

**Table 5.** Comparing novices and advanced students

		Score (%)	Pre-test	Post-test	W, p
AEP (15)	Novices (6)	Overall	52.31 (7.94)	80.09 (13.77)	21, .028**
		Debug. questions	41.67 (20.41)	56.94 (34.73)	ns
	Adv. (9)	Overall	77.16 (9.8)	98.46 (3.7)	45, .008**
		Debug. questions	77.78 (14.43)	97.22 (5.89)	36, .01**
WPEP (11)	Novices (6)	Overall	56.94 (3.4)	91.2 (7.54)	21, .028**
		Debug. questions	38.89 (13.61)	87.5 (19.54)	15, .043**
	Adv. (5)	Overall	72.24 (7.85)	97.22 (3.93)	15, .041**
		Debug. questions	66.67 (23.57)	100 (0)	10, .059*

## 5 Discussion and Conclusions

Previous studies show that WEs are beneficial for novices in comparison to problem solving [6, 15, 20]. In a previous study, alternating WEs with problem solving was found to be the best strategy in SQL-Tutor [1]. However, the inclusion of ErrEx has not been studied before in this instructional domain. In this study, we compared students' performance in two conditions: alternating worked examples/problem (AEP), and worked example/problem pairs and erroneous examples/problem pairs (WPEP).

We found no significant difference between AEP and WPEP conditions on pre- and post-test performance, but the participants in both conditions improved significantly their scores on the post-test from the pre-test. Students in the WPEP condition acquired more debugging knowledge than those in the AEP condition. A possible explanation is that extra learning and additional time in the correcting phase of erroneous examples contribute to this benefit. Furthermore, students who learned with erroneous examples

showed higher performance on problem solving as measured by the number of attempts per problems and also the number of mistakes made. This suggests that the erroneous examples aid learning more than worked examples, which confirmed our hypothesis H1. The WPEP participants learned from both worked examples and erroneous examples. When students were asked to identify and correct errors in ErrEx, they engaged in deeper cognitive processing in comparison to when they engage with WEs. Therefore, they were better prepared for concepts required in the next isomorphic problem compared to the situation when they received WEs.

Although the present results suggest that ErrExs aid learning, an important issue concerns the benefit for students with different knowledge levels. Hypothesis H2, like in [12], was that advanced students would learn more from erroneous examples than novices. However, we did not find a difference between novices and advanced students in WPEP; both subgroups improved their debugging knowledge. Furthermore, novices from the WPEP group improved their debugging knowledge significantly more than their peers of similar abilities from the AEP group (with the effect size close to 1 sigma). Therefore, the students with any knowledge level benefitted from erroneous examples. One of the possible explanations for a different finding in comparison to [12] is in the instructional domains used in each study. The instructional task of the Große and Renkl study was probability (a well-defined instructional task), while the students were specifying SQL queries for ill-defined tasks in our study.

One of the limitations of our study is the small sample size. The timing of the study coincided with assignments in other courses the participants were taking, so many participants did not complete the full study. We plan to conduct the same study with a larger population. McLaren et al. [21] found that erroneous examples led to a delayed learning effect. However, our study did not include a delayed test. It would be interesting to see the results of the delayed learning effect.

Our study demonstrated that an improved instructional strategy, WPEP, resulted in improved problem solving, and that it also benefitted students with various levels of prior knowledge in SQL-Tutor. The results suggest that the students with different levels of prior knowledge may perform differently with worked examples, erroneous examples, and problem-solving. In addition, all students in our study learned SQL in the lectures before participating in our study. One direction for future work would be to develop an adaptive strategy that decides what learning activities (TPS, WE or ErrEx) to provide to the student based on his/her student model.

## References

1. Shareghi Najar, A., Mitrovic, A.: Examples and tutored problems: how can self-explanation make a difference to learning? In: Lane, H., Yacef, K., Mostow, J., Pavlik, P. (eds.) AIED 2013. LNCS, vol. 7926, pp. 339–348. Springer, Heidelberg (2013)
2. Sweller, J., Van Merriënboer, J.J., Paas, F.G.: Cognitive architecture and instructional design. *Educ. Psychol. Rev.* **10**(3), 251–296 (1998)
3. van Gog, T., Rummel, N.: Example-based learning: integrating cognitive and social-cognitive research perspectives. *Educ. Psychol. Rev.* **22**(2), 155–174 (2010)

4. Atkinson, R.K., Derry, S.J., Renkl, A., Wortham, D.: Learning from examples: instructional principles from the worked examples research. *Rev. Educ. Res.* **70**(2), 181–214 (2000)
5. Kalyuga, S., Chandler, P., Tuovinen, J., Sweller, J.: When problem solving is superior to studying worked examples. *Educ. Psychol.* **93**(3), 579 (2001)
6. van Gog, T., Kester, L., Paas, F.: Effects of worked examples, example-problem, and problem-example pairs on novices' learning. *Contemp. Educ. Psychol.* **36**(3), 212–218 (2011)
7. van Gog, T.: Effects of identical example-problem and problem-example pairs on learning. *Comput. Educ.* **57**(2), 1775–1779 (2011)
8. Schwonke, R., Renkl, A., Krieg, C., Wittwer, J., Alevin, V., Salden, R.: The worked-example effect: not an artefact of lousy control conditions. *Comput. Hum. Behav.* **25**(2), 258–266 (2009)
9. McLaren, B.M., Isotani, S.: When is it best to learn with all worked examples? In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) *AIED 2011*. LNCS, vol. 6738, pp. 222–229. Springer, Heidelberg (2011)
10. McLaren, B.M., Lim, S.-J., Koedinger, K.R.: When and how often should worked examples be given to students? New results and a summary of the current state of research. In: *Proceedings of 30th Annual Conference of the Cognitive Science Society*, pp. 2176–2181 (2008)
11. Durkin, K., Rittle-Johnson, B.: The effectiveness of using incorrect examples to support learning about decimal magnitude. *Learn. Instr.* **22**(3), 206–214 (2012)
12. Große, C.S., Renkl, A.: Finding and fixing errors in worked examples: can this foster learning outcomes? *Learn. Instr.* **17**(6), 612–634 (2007)
13. Tsovaltzi, D., McLaren, B.M., Melis, E., Meyer, A.-K.: Erroneous examples: effects on learning fractions in a web-based setting. *Technol. Enhanc. Learn.* **4**(3–4), 191–230 (2012)
14. Booth, J.L., Lange, K.E., Koedinger, K.R., Newton, K.J.: Using example problems to improve student learning in algebra: differentiating between correct and incorrect examples. *Learn. Instr.* **25**, 24–34 (2013)
15. Najar, A.S., Mitrovic, A.: Do novices and advanced students benefit differently from worked examples and ITS? In: *Proceedings of International Conference Computers in Education*, pp. 20–29 (2013)
16. Mitrovic, A.: An intelligent SQL-Tutor on the web. *Artif. Intell. Educ.* **13**, 173–197 (2003)
17. Weerasinghe, A., Mitrović, A.: Studying human tutors to facilitate self-explanation. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) *ITS 2006*. LNCS, vol. 4053, pp. 713–715. Springer, Heidelberg (2006)
18. Chi, M.T., Leeuw, N., Chiu, M.H., LaVancher, C.: Eliciting self-explanations improves understanding. *Cogn. Sci.* **18**(3), 439–477 (1994)
19. Kim, R.S., Weitz, R., Heffernan, N.T., Krach, N.: Tutoresd problem solving vs. “pure” worked examples. In: *Proceedings of 31st Annual Conference of the Cognitive Science Society*. Cognitive Science Society (2009)
20. McLaren, B.M., van Gog, T., Ganoë, C., Yaron, D., Karabinos, M.: Exploring the assistance dilemma: comparing instructional support in examples and problems. In: Trausan-Matu, S., Boyer, K.E., Crosby, M., Panourgia, K. (eds.) *ITS 2014*. LNCS, vol. 8474, pp. 354–361. Springer, Heidelberg (2014)
21. McLaren, B.M., Adams, D.M., Mayer, R.E.: Delayed learning effects with erroneous examples: a study of learning decimals with a web-based tutor. *Artif. Intell. Educ.* **25**(4), 520–542 (2015)

# Automatic Question Generation: From NLU to NLG

Karen Mazidi<sup>(✉)</sup> and Paul Tarau

Department of Computer Science and Engineering, University of North Texas,  
Denton, TX 76207, USA

karenmazidi@my.unt.edu, paul.tarau@unt.edu

**Abstract.** Questioning has been shown to improve learning outcomes, and automatic question generation can greatly facilitate the inclusion of questions in learning technologies such as intelligent tutoring systems. The majority of prior QG systems use parsing software and transformation algorithms to create questions. In contrast, the approach described here infuses natural language understanding (NLU) into the natural language generation (NLG) process by first analyzing the central semantic content of each independent clause in each sentence. Then question templates are matched to what the sentence is communicating in order to generate higher quality questions. This approach generated a higher percentage of acceptable questions than prior state-of-the-art systems.

## 1 Introduction

Rus et al. [33] describe automatic question generation as a dialogue and discourse task, drawing on both NLU (Natural Language Understanding) and NLG (Natural Language Generation). This description is particularly apt for a system such as the one outlined in this paper which takes raw text as input, performs NLU analysis [2], and then performs the NLG task of transforming the intermediate representation into English language questions. McDonald describes NLG as the process by which thought is rendered into language [16]. In generating questions from text, we are taking thoughts represented in declarative text and rendering them into another form of thought: questioning on the original thoughts. Piwek and Boyer [28] observe that question generation could be viewed as a search for algorithms to transform inputs to certain types of outputs, and that the combinations of inputs, outputs and algorithms are already quite varied in this developing research area. A caveat could be added to this apt observation, namely a concern that current state-of-the-art approaches at their core are essentially text processing approaches and there is much work to be done infusing AI techniques into automatic question generation. Until the day when we can state with the certainty provided by a Turing test that computers *understand* natural language, NLG systems are challenged to move beyond text processing and illusions of understanding as demonstrated in this exchange with Eliza [36]:

**Human:** What do you think of natural language generation?

**Eliza:** We were discussing you, not me.



Eliza was a chimera. Nevertheless it opened the door to exploring the limits of human-machine interaction through language. Educational technology is one of the most important applications of this interaction. Research demonstrates that students learn very little from reading textbooks, and need dialogue-based support, including questioning, to gain a deep understanding of material [35].

## 2 The Importance of Questions in Learning

That questioning improves learning has long been supported by research [3, 31], and recent studies reconfirm this important pedagogical principle [27, 29]. The IES Practice Guide for Teachers [26] discusses 7 evidence-based principles of learning, 2 of which explicitly refer to questioning. The APA and APS jointly conducted a symposium exploring principles of learning [10], and 6 of these 25 principles deal directly with questioning. Based on an exhaustive review of research providing data about which educational strategies work and which do not, Roediger and Pyc [30] have distilled this information into three principles: (1) the distribution (spacing and interleaving) of material and practice during learning, (2) the frequent assessment of learning (direct and indirect positive effects of quizzing and testing), and (3) explanatory questioning. Note that two of the three recommendations involve questioning, and the other one involves the timing and presentation of material and questions. Answering questions, whether presented by an instructor or self-testing while studying, has several positive benefits. First, retrieving information makes it more retrievable in the future and can transfer to other concepts. Testing helps students identify what they know and what they need to study further. Test potentiation studies show that students learn more restudying after taking a test than if they have not taken a test [30]. A foundation of factual questions is an underappreciated prerequisite to deeper conceptual questions. A document from the National Academy of Sciences [4] asserts the importance of students building a strong and deep foundation of factual knowledge. This conclusion is based on research that compares the performance of experts and novices.

Experts, regardless of the field, always draw on a richly structured information base; they are not just “good thinkers” or “smart people”. The ability to plan a task, to notice patterns, to generate reasonable arguments and explanations, and to draw analogies to other problems are all more closely intertwined with factual knowledge than was once believed.

## 3 Prior Work in Automatic Question Generation

Automatically generated questions have been shown to be as effective as human-authored ones in studies dating from the late 1970s [38] to 2015 [15]. Even for QG systems that tend to overgenerate, selecting among automatically generated questions saves significant amounts of time compared to manually generating them [13].

The past decade has witnessed a renaissance in the field of automatic question generation. Evidence is in the growth in both the number and diversity of recent approaches. Despite the early, promising work demonstrated in Wolfe’s AUTOQUEST [38], the field of automatic generation from text appears to have been relatively quiet in the closing decades of the 20th Century. In fact, up until the early 2000s and beyond, computer-based instructional systems continued to use frame-based methods in which all course content, including questions, was authored by hand [39]. The recent resurgence of interest in automatic question generation is motivated in part by the evolution of intelligent tutoring and computer-assisted learning systems, and the need for more rapid development of questions for these systems.

Apart from a very few outliers in specialized domains with limited results, the majority of question generation systems input a text source, automatically parse the sentences, and transform sentences into questions. Two major design decisions are: (1) selecting parsing software, and (2) deciding whether to use external templates or internal rules for sentence-to-question transformation. In a recent survey of question generation approaches for educational applications, Le et al. [19] observed that template-based approaches tended to perform better than systems that syntactically rearranged the source text. Our observation is that generating any question type is theoretically possible in any approach, but that some approaches make some question types easier to generate than others.

One of the most popular QG approaches involves parsing text with a phrase structure parser and then forming questions using templates [20, 21, 32, 40] or transformation rules and tree manipulation tools [1, 8, 9, 13]. Heilman notes [13] that these purely syntactic approaches do not allow higher-level abstractions that may be possible with more semantically informed approaches. Nevertheless, as these QG systems demonstrate, they can be robust and productive methods for generating fact-based questions.

An alternative to the phrase-structure parse is the semantic role label (SRL) parse which identifies for each predicate in a sentence, its associated arguments and modifiers, and specifies their semantic roles. A QG system can then extract arguments and modifiers for question construction [5, 22–24]. These systems are able to generate a wider variety of questions than the phrase structure approach and are not as closely bound to the sentence source text.

A third type of parse used in QG systems is the dependency parse, which connects words in a sentence in a graphical structure based on their grammatical and functional relations. Although the SRL parse is sometimes referred to as a shallow semantic parse, certain dependency relations give greater insight into semantics than the SRL parse. The italicized portions of the sentences in Table 1 were all parsed as **Arg1** by the SRL parser. In contrast, the labels provided by the dependency parser are quite varied, and provide opportunities to glean varied meanings from what is simply **Arg1** in the SRL parse.

Although the dependency parse had previously been used as an ancillary tool and for sentence simplification, Mazidi et al. [25] was the first to fully exploit dependency relations in question generation. The approach described in

**Table 1.** Arg1 versus dependency labels

Sentence	ArgN	Dep. Label	Meaning
1. John broke <i>the window</i>	Arg1	dobj	second entity in relation
2. John was <i>angry</i>	Arg1	acomp	property of subject
3. John felt <i>that everyone always ignored him</i>	Arg1	ccomp	proposition of subject
4. John is <i>an angry man</i>	Arg1	attr	definition of subject
5. John wanted <i>to make his presence heard</i>	Arg1	xcomp	purpose
6. John began <i>bleeding profusely</i>	Arg1	xcomp	action

this paper extends that earlier work by adding important observations that can be gleaned from text structure, as described below, thus infusing more AI into automatic question generation systems.

## 4 Approach: Infusing NLG with NLU

Examining the current state of the art of QG reveals that NLU is an underdeveloped prerequisite. Typical QG approaches, as described above, parse sentences then rearrange sentence constituents to transform the sentence into as many questions as allowed by English grammar rules. In contrast, our approach first classifies what the sentence is communicating by analyzing the type and arrangement of syntactic and semantic constituents. These *sentence patterns* fall into a surprisingly small number of categories. For each sentence, the QG system classifies its sentence pattern prior to the question generation phase. The sentence pattern is key to determining what type of question should be asked about that sentence. This analysis was based on text extracted from open source textbooks as well as Wikipedia. Each text passage consisted of the text of one chapter section, or Wikipedia text of equivalent length. Table 2 describes the distribution of sentence patterns in the test set, described in Table 3. In order to identify patterns to be included in the QG

**Table 2.** Pattern distribution in test sets A and B

Pattern	Meaning	Frequency
S-V-acomp	adjectival complement that describes the subject	8 %
S-V-attr	nominal predicative complement defining the subject	14 %
S-V-ccomp	clausal complement indicating a proposition of subject	7 %
S-V-dobj	indicates the relation between two entities	28 %
S-V-iobj-dobj	indicates the relation between three entities	< 1%
S-V-parg	phrase describing the how/what/where of the action	17 %
S-V-xcomp	non-finite clause-like complement	8 %
S-V	indicates an action of the entity	14 %
other	combinations of constituents	4 %

**Table 3.** Test sets A (Textbooks) and B (Wikipedia)

No	Topic	No. Sentences	% Generated	Grade Level	% Accept
A1	Lymphatic System	136	65 %	14	62 %
A2	Eukaryotic Cells	165	43 %	14	52 %
A3	Federalism	117	70 %	14	46 %
A4	International Trade	83	65 %	12	56 %
B1	Chemical Bonds	138	46 %	14	54 %
B2	Planned Economies	67	50 %	15	70 %
B3	Toledo War	83	63 %	13	50 %
B4	Tornadoes	151	38 %	13	48 %
	Average	118	53 %	14	<b>55%</b>

system, the following criteria was used: (1) Does the sentence pattern occur frequently across passages in different domains? (2) Is the semantic information conveyed by the sentence pattern consistent across different instances? and (3) Does the sentence pattern identify important content in source sentences so that generated questions will be meaningful and not trivial?

---

**Algorithm 1.** Sentence Object Formation
 

---

```

S ← set of parsed sentences
for each sentence s ∈ S do
  DIVIDEINDEPCLAUSES(s)
  for each indepClause ic ∈ s: do
    Step 1: Add predicate complex
    ic[pred.label] ← predicate
    icRoot = pred.index
    Step 2: Add constituents
    for each dep ∈ dependencies do
      if dep.gov == icRoot then
        ic[const.label] ← dp
    Step 3: Add ArgMs to IC
    for each AM in ArgMs for icRoot do
      ic[AM.label] ← AM (causative, locative, etc.)
    Step 4: Determine pp type
    for each pp in PPs do
      if pp == ArgN then
        pp.label = ppArg
      else
        pp.label = ppMod
    Step 5: Determine ic structure
    Determine ic type (passive, active, ...)
    Classify ic pattern
    Flag sentences with questionable parse
  
```

---

For each sentence in a text passage, the system gathers data from both a dependency parse and an SRL parse in order to form an intermediate representation from which questions can be generated. The steps involved in creating this intermediate representation, the *sentence object*, are given in Algorithm 1. After a sentence object is created for each independent clause of each sentence, the sentence pattern is compared against approximately 50 templates. If a template matches the pattern, a question can be generated. Templates are designed to ask questions related to the major point of the sentence as identified in the pattern (see Table 4). Templates also contain filter conditions which are checked. Filter conditions may check for the presence or absence of particular verbs (particularly be, do and have), whether the sentence is in active or passive voice,

**Table 4.** Sample questions by sentence type

Pattern and Sample
<p>1. <b>S-V-acomp</b> Adjectival complement that describes the subject.            S: Vacuoles are somewhat larger than vesicles.            Q: Indicate properties or characteristics of vacuoles.</p>
<p>2. <b>S-V-attr</b> Nominal predicative complement following copula, often defining the subject.            S: An antigen is a chemical structure that binds to T or B receptors.            Q: Define or describe an antigen.</p>
<p>3. <b>S-V-ccomp</b> Clausal complement indicates a proposition of or about the subject.            S: Seismic waves indicate that the outer core must be liquid and the inner core must be solid.            Q: What do seismic waves indicate?</p>
<p>4. <b>S-V-dobj</b> Indicates the relation between two entities.            S: The moon orbits the Earth.            Q: Describe the relation or interaction between the moon and the Earth.</p>
<p>5. <b>S-V-iobj-dobj</b> Indicates the relation between three entities.            S: The Bill of Rights gave the new federal government greater legitimacy.            Q: What gave the new federal government greater legitimacy?</p>
<p>6. <b>S-V-pparg</b> Prepositional phrase that is required to complete the meaning.            S: From outside to inside, the planet is divided into crust, mantle, and core.            Q: From outside to inside, into what is the planet divided?</p>
<p>7. <b>S-V-xcomp</b> Non-finite clause-like complement.            S: The lymphatics eventually merge to form larger lymphatic vessels.            A: For what purpose do the lymphatics eventually merge?</p>
<p>8. <b>S-V</b> May contain phrases that are not considered arguments.            S: Eventually, the general surrendered.            Q: What did the general eventually do?</p>

and other conditions that are documented in the template file. The source code, data, and generated questions are available<sup>1</sup> for those interested in implementation details.

Examples of source sentences and generated questions are provided in Table 4. An independent clause can be viewed as a proposition, and the predicate identifies the relationship, property or state of the entities participating in the proposition. The predicate determines the number of participants, or arguments, that are allowed [18]. In the S-V-iobj-dobj pattern, for example, there must be 3 entities identified in the sentence. The predicate is often the main verb but there are other constructions in which the predicate can be found in other syntactic categories. The *acomp* constituent follows a copula verb which has negligible semantic content in this construction. The meaning is carried by the *acomp*, which may be an adjective or a noun. Linguists often used the term *xcomp* to denote predicate complements of various syntactic categories [18]. In contrast, the universal dependency relations divide the complements into *acomp* for AP, *attr* for NP, *ccomp* for subordinate clauses, leaving *xcomp* for VP. It matters what syntactic category a complement belongs to because this provides important semantic indications of what the clause is saying. Take for instance a *ccomp* compared to a *dobj*. They differ syntactically in that the *ccomp* is a clause whereas the *dobj* is a phrase. Semantically, the *dobj* identifies the second entity in the predicate relation whereas the *ccomp* can be viewed as an independent proposition either indicated by or about the subject.

## 5 Results and Error Analysis

There is no standard way to evaluate automatically generated questions. Recent work in QG and other NLP applications favors evaluation by crowdsourcing which has proven to be both cost and time efficient and to achieve results comparable to human evaluators [14, 34]. A similar evaluation of the generated questions was conducted using Amazon’s Mechanical Turk Service. Workers were selected with at least 90% approval rating on their prior work and who were located in the US and proficient in US English. To monitor quality, work was submitted in small batches, manually inspected, and run through software to detect workers whose ratings did not correspond well with fellow workers. Each question was rated on a 1–5 scale by 4 workers. The four scores were averaged and a mean over 3.5 was considered acceptable. The first author gave a binary acceptable rating to each question. Questions were counted as acceptable if they received an acceptable rating from the first author *and* an average acceptable rating from the MTurk workers. Agreement between this rating and the MTurk workers’ acceptability score was  $\kappa = 0.67$ .

The QG system output 55% acceptable questions without any ranking component. The most frequently cited state-of-the-art QG system, Heilman and Smith [11], achieved 52% acceptable questions after a comparable evaluation, when considering only the top 20% of their automatically ranked questions.

<sup>1</sup> <http://www.karenmazidi.com/>.

Future work will add a component for automatically ranking generated questions in order to raise acceptability rate even further.

Analysis of questions that received unacceptable ratings reveal that 62% were unacceptable due to vagueness, 24% due to minor parsing errors such as including too many or too few words as dependents of the constituent head, and 14% indicate areas for additional work in the system. A key advance in improving the percentage of acceptable questions will be to eliminate vague questions through coreference resolution.

## 6 Discussion

A central observation of this paper is that sentence structure is a key aspect of natural language understanding and that advances in NLU can benefit question generation systems. To the degree that computer applications can “understand” natural language, it is with the aid of lexical resources such as WordNet and syntax parses. These different types of parses follow different linguistic traditions. The dependency grammar tradition is probably the oldest, having roots dating back to ancient Greek and Indian linguistic traditions. Analysis of semantic roles could be traced back to the Indian *karaka* theory of the 7th Century [17].

The question that arises in looking at these different approaches is: How well do any of these types of parses correspond to how humans parse sentences as we listen to a speaker or read text? Chomsky [6] proposed that we have an internal grammar in our minds that allows us to make sense of language. This idea has been controversial since its publication, but recent research in neuroscience has found some evidence that Chomsky was on the right track, although the research provides no insight as to whether these structures are innate or developed through experience. Using magnetoencephalography, researchers at NYU were able to identify distinct cortical activity that concurrently tracked auditory input (stripped of acoustic cues) at different hierarchical levels: words, phrases, sentences [7]. In other words, a hierarchy of neural processing underlies grammar-based internal construction of language. This exciting research may in the future be able to tease out what information these hierarchical processes actually encode. As that occurs, perhaps parsers could be developed in which sentence structure corresponds to our internally encoded structure. For example, it’s doubtful that we hear the beginning of sentence and think: that’s an NP. Rather, we think: that’s what we are talking about, i.e., the subject. And the rest of the sentence is just telling us what action surrounded that subject and possibly other entities. While we await further advances from neuroscience, it seems practicable to continue research in sentence representation forms that correspond to an intuitive understanding of sentence meaning, such as the sentence representation discussed in this work, rather than a particular linguistic tradition.

The question generation system presented here introduced a fresh approach to analyzing intrasentential structure and meaning that is both intuitive and practical. The pattern of the constituent structure indicates what meaning can

be inferred from the sentence. This enables generation of questions relevant to the central point of a sentence and avoids the overgeneration problem of prior work. The approach can be implemented with off-the-shelf parsers that provide both a dependency and an SRL parse. The QG system achieved a question acceptability rate of 55 %, a rate higher than the top 20 % of ranked questions from the most cited prior state-of-the-art system. This improvement is due in part to the internal NLU analysis of what the sentence is communicating which enabled the system to match to an appropriate template specific to the semantics of that sentence structure.

## References

1. Ali, H., Chali, Y., Hasan, S.: Automation of question generation from sentences. In: Proceedings of QG2010: The Third Workshop on Question Generation (2010)
2. Allen, J.: Natural Language Understanding. The Benjamin/Cummings Publishing Company, Redwood City (1995)
3. Anderson, R., Biddle, W.: On asking people questions about what they are reading. In: Bower, G. (ed.) Psychology of learning and motivation, vol. 9. Elsevier (1975)
4. Bransford, J., Brown, A., Cocking, R.R.: How People Learn. National Academy Press, Washington, DC (2004)
5. Chali, Y., Hasan, S.: Towards Topic-to-Question Generation. Computational Linguistics. MIT Press, Cambridge (2015)
6. Chomsky, N.: Syntactic Structures. Mouton, The Hague (1957)
7. Ding, N., Melloni, L., Zhang, H., Tian, X., Poeppel, D.: Cortical tracking of hierarchical linguistic structures in connected speech. *Nat. Neurosci.* **19**(1), 158–164 (2015)
8. Gates, D.: Automatically generating reading comprehension look-back strategy: questions from expository texts. DTIC Document (2008)
9. Gates, D.: Generating look-back strategy questions from expository texts. In: The Workshop on the Question Generation Shared Task and Evaluation Challenge, NSF, Arlington (2008)
10. Graesser, A., Halpern, D., Hakel, M.: Principles of Learning. Task Force on Lifelong Learning at Work and at Home, Washington, DC (2008)
11. Heilman, M., Smith, N.: Question generation via overgenerating transformations and ranking. DTIC Document (2009)
12. Heilman, M., Smith, N.: Good question! statistical ranking for question generation. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, ACL (2010)
13. Heilman, M.: Automatic Factual Question Generation from Text. Carnegie Mellon University, Pittsburgh (2011)
14. Heilman, M., Smith, N.: Rating computer-generated questions with Mechanical Turk. In: Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk, ACL (2010)
15. Huang, Y.-T., Mostow, J.: Evaluating human and automated generation of distractors for diagnostic multiple-choice cloze questions to assess children’s reading comprehension. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M.F. (eds.) AIED 2015. LNCS, vol. 9112, pp. 155–164. Springer, Heidelberg (2015)
16. Indurkha, N., Damerau, F.: Handbook of Natural Language Processing, vol. 2. CRC Press, Boca Raton (2010)



17. Jurafsky, D., Martin, J.: *Speech & Language Processing*. Pearson Education, Upper Saddle River (2008)
18. Kroeger, P.: *Analyzing Grammar: An Introduction*. Cambridge University Press, Cambridge (2005)
19. Le, N.-T., Kojiri, T., Pinkwart, N.: Automatic question generation for educational applications – the state of art. In: van Do, T., Thi, H.A.L., Nguyen, N.T. (eds.) *Advanced Computational Methods for Knowledge Engineering*. AISC, vol. 282, pp. 325–338. Springer, Heidelberg (2014)
20. Liu, M., Calvo, R.A., Rus, V.: Automatic question generation for literature review writing support. In: Alevin, V., Kay, J., Mostow, J. (eds.) *ITS 2010, Part I*. LNCS, vol. 6094, pp. 45–54. Springer, Heidelberg (2010)
21. Liu, M., Calvo, R., Rus, V.: G-Asks: an intelligent automatic question generation system for academic writing support. *Dialogue and Discourse* **3**(2), 101–124 (2012). Special Issue on Question Generation
22. Lindberg, D., Popowich, F., Nesbit, J., Winne, P.: Generating natural language questions to support learning on-line. In: *European Conference for Natural Language Generation* (2013)
23. Mannem, P., Prasad, R., Joshi, A.: Question generation from paragraphs at UPenn: QGSTEC system description. In: *Proceedings of QG2010: The Third Workshop on Question Generation* (2010)
24. Mazidi, K., Nielsen, R.D.: Pedagogical evaluation of automatically generated questions. In: Trausan-Matu, S., Boyer, K.E., Crosby, M., Panourgia, K. (eds.) *ITS 2014*. LNCS, vol. 8474, pp. 294–299. Springer, Heidelberg (2014)
25. Mazidi, K., Nielsen, R.D.: Leveraging multiple views of text for automatic question generation. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M.F. (eds.) *AIED 2015*. LNCS, vol. 9112, pp. 257–266. Springer, Heidelberg (2015)
26. Pashler, H., Bain, P., Bottge, B., Graesser, A., Koedinger, K., McDaniel, M., Metcalfe, J.: *Organizing Instruction and Study to Improve Student Learning*. IES Practice Guide, NCER (2007)
27. Peverly, S., Wood, R.: The effects of adjunct questions and feedback on improving the reading comprehension skills of learning-disabled adolescents. *Contemp. Educ. Psychol.* **26**(1), 25–43 (2001). Elsevier
28. Piwek, P., Boyer, K.: Varieties of question generation: introduction to this special issue. *Dialogue and Discourse* **3**, 1–9 (2012)
29. Roediger, H., Karpicke, J.: *The power of testing memory: basic research and implications for educational practice*. *Perspectives on Psychological Science*, vol. 1. SAGE (2006)
30. Roediger, H., Karpicke, J., Pyc, M.: Inexpensive techniques to improve education: applying cognitive psychology to enhance educational practice. *J. Appl. Res. Memory Cogn.* **1**(4), 242–248 (2012)
31. Rothkopf, E.: Learning from written instructive materials: an exploration of the control of inspection behavior by test-like events. *Am. Educ. Res. J.* (1966)
32. Rus, V., Cai, Z., Graesser, A.C.: Experiments on generating questions about facts. In: Gelbukh, A. (ed.) *CICLing 2007*. LNCS, vol. 4394, pp. 444–455. Springer, Heidelberg (2007)
33. Rus, V., Wyse, B., Piwek, P., Lintean, M., Stoyanchev, S., Moldovan, C.: A detailed account of the first question generation shared task evaluation challenge. *Dialogue Discourse* **3**(2), 177–204 (2012)
34. Snow, R., O’Connor, B., Jurafsky, D., Ng, A.: Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In: *Proceedings of the conference on empirical methods in natural language processing, ACL* (2008)

35. VanLehn, K., Graesser, A., Jackson, G., Jordan, P., Olney, A., Rose, C.: When are tutorial dialogues more effective than reading? *Cogn. Sci.* **31**(1), 3–62 (2007)
36. Weizenbaum, J.: ELIZA a computer program for the study of natural language communication between man and machine. *Commun. ACM* **9**(1), 36–45 (1966)
37. Wolfe, J.: Automatic question generation from text-an aid to independent study. *ACM SIGCUE Outlook*, vol. 10. ACM (1976)
38. Wolfe, J.: Reading Retention as a Function of Method for Generating Interspersed Questions. ERIC (1977)
39. Woolf, B.: *Building Intelligent Interactive Tutors: Student-centered Strategies for Revolutionizing e-Learning*. Morgan Kaufmann, Burlington (2010)
40. Wyse, B., Piwek, P.: Generating questions from openlearn study units. In: *Proceedings, The 2nd Workshop on Question Generation*, vol. 1 (2009)

# Using Eye-Tracking to Determine the Impact of Prior Knowledge on Self-Regulated Learning with an Adaptive Hypermedia-Learning Environment

Michelle Taub<sup>(✉)</sup> and Roger Azevedo

Department of Psychology, Laboratory for the Study of Metacognition  
and Advanced Learning Technologies, North Carolina State University,  
Raleigh, NC, USA

{mtaub, razeved}@ncsu.edu

**Abstract.** Recent research on self-regulated learning (SRL) includes multi-channel data, such as eye-tracking, to measure the deployment of key cognitive and metacognitive SRL processes during learning with adaptive hypermedia systems. In this study we investigated how 147 college students' proportional learning gains (PLGs), proportion of time spent on areas of interest (AOIs), and frequency of fixations on AOI-pairs, differed based on their prior knowledge of the overall science content, and of specific content related to sub-goals, as they learned with MetaTutor. Results indicated that students with low prior sub-goal knowledge had significantly higher PLGs, and spent a significantly larger proportion of time fixating on diagrams compared to students with high prior sub-goal knowledge. In addition, students with low prior knowledge had significantly higher frequencies of fixations on some AOI-pairs, compared to students with high prior knowledge. The results have implications for using eye-tracking (and other process data) to understand the behavioral patterns associated with underlying cognitive and metacognitive SRL processes and provide real-time adaptive instruction, to ensure effective learning.

**Keywords:** Metacognition · Self-regulated learning · Eye tracking · Prior knowledge · Adaptive hypermedia-learning environments · Process data

## 1 Introduction

Self-Regulated Learning (SRL) involves a student actively monitoring and regulating his or her cognitive, affective, metacognitive and motivational (CAMP) processes [1]. Although engaging in SRL can be effective for students, research has shown that students typically fail to monitor and regulate their CAMP process for a variety of reasons, including a lack of knowledge and ability to enact effective cognitive strategies, making inaccurate judgments following metacognitive monitoring, lack of interest in the topic, and inability to regulate negative emotions. Thus, researchers have designed advanced learning technologies (ALTs), such as adaptive hypermedia [1], and Intelligent Tutoring Systems (ITSs) [2, 3] to foster SRL for students of different ages and prior knowledge levels.

In this study, we make several assumptions regarding SRL as an event that involves the temporal deployment of CAMM processes that unfold during learning [1, 4] with an ALT, such as MetaTutor [1]. As such, we have used eye-tracking as an on-line trace method to capture the temporally unfolding events related to the underlying cognitive and metacognitive SRL processes during learning about the human circulatory system with MetaTutor. This study is also based on emerging research on the use of eye-tracking data to examine the cognitive strategies used during learning with ALTs. For example, see [5, 6].

Research on self-regulated learning (SRL) has primarily relied on self-report measures; however researchers are expanding on their measurement methods, and are including multi-channel data tools, such as eye-tracking, to measure how students self-regulate their learning with hypermedia-learning environments. For example, Bondareva et al. [7] showed that gaze behaviors in MetaTutor can be predictive of learning gains, with 78.3 % accuracy. They used a simple logistic regression model that uses fixation-related measures (e.g., duration, rate of fixation, etc.) and the proportion of transitions between AOIs, among other measures, to predict proportional learning gains. Moreover, Conati et al. [5] captured gaze data to determine the predictive power of attention patterns to adaptive hints in an educational learning game. Results of this research showed that eye data can effectively be used as a predictor of learning in ALTs [5]. Similarly, Jaques et al. analyzed eye-tracking data from MetaTutor that included tracking gaze transitions between AOI pairs, as a measure of engagement. This was shown to be an effective means of determining learners' boredom (69 % accuracy) and curiosity (73 % accuracy) [8]. Thus, there is evidence that we can use eye-tracking data to detect students' SRL behaviors as they learn with ALTs.

There are many factors that can impact how students self-regulate their learning as they interact with ALTs. For example, individual differences, such as prior knowledge, have been found to significantly impact how students deploy SRL processes as they learn with ALTs [9, 10]. These studies have reported that students with high prior knowledge of the circulatory system engaged in significantly more metacognitive SRL strategies, compared to students with low prior knowledge of the topic [9]; and when students were provided prompts and feedback from pedagogical agents, students with low prior knowledge of the circulatory system took significantly more notes than students with high prior knowledge [10]. Thus, it is evident from these studies, and others, that prior knowledge can significantly impact how students engage in SRL as they interact with ALTs.

The goal of this paper, therefore, is to determine if prior knowledge has an impact on students' fixations on different areas of interest (AOIs), as they learn with MetaTutor, an adaptive hypermedia-learning environment that teaches students about the circulatory system. As previously mentioned, studies have investigated SRL in ALTs using eye-tracking data, however the novelty of this paper is to include the impact of prior knowledge in using eye-tracking to investigate SRL in ALTs. In addition, this paper examines AOI pairs from text to other specific AOIs, which has not been examined in previous studies. Therefore, the results from this paper can contribute to the SRL and ITS communities by providing novel sets of data and challenges to understanding the complex nature of cognitive and metacognitive SRL processes.

## 2 Methods

### 2.1 Participants and Experimental Design

Participants in this study were 147 undergraduate students (52.4 % female) from three North American Universities. Their ages ranged from 18 to 38 ( $M = 20.61$ ,  $SD = 2.73$ ). Participants were compensated \$10 per hour for participating. Students were randomly assigned to one of two conditions prior to beginning the study, and on average, had low pre-test scores ( $M = 17.18$ ,  $SD = 4.52$ ) on the 30-item multiple-choice pre-test.

### 2.2 MetaTutor: An Adaptive Hypermedia-Learning Environment

In this study, students interacted with MetaTutor, an adaptive hypermedia-learning environment that teaches about the circulatory system [1]. In this version of MetaTutor, there are 47-pages of text and diagrams, which provide information on different sub-topics related to the circulatory system. As students learn with MetaTutor, they are provided with an overall learning goal, which is: learn all you can about the circulatory system. Specifically, be sure to learn about all the different organs and other components of the circulatory system, and their purpose within the system, how they work both individually and together, and how they support the healthy functioning of the body. Students are given 90 min to complete several self-set sub-goals (e.g., heartbeat), and thus achieve the overall learning goal. There are several elements that together make up the interface of the learning environment (see Fig. 1). On the top left corner, there is a timer that counts down the time remaining in the session. Students are given a total of 90 min to attempt to complete their sub-goals, and subsequently their overall learning goal. The table of contents is located beneath the timer, and provides students with a list of all the content pages, which they can access by clicking on the page title, which will link them to the given page. In the center, the content is presented, with the text on the left and the diagram on the right. The self-set sub-goals are placed just above the content, and the overall learning goal is located above the sub-goals. There are seven predetermined sub-goals, which students are guided to set during the initial sub-goal setting phase: (1) path of blood flow, (2) heartbeat, (3) heart components, (4) blood vessels, (5) blood components, (6) purposes of the circulatory system, and (7) malfunctions of the circulatory system. Participants are required to set two sub-goals during the sub-goal setting phase prior to beginning the learning session, however they are also able to add more sub-goals during the learning session. At the top right corner, one of the four pedagogical agents is displayed. Finally, the SRL palette is located under the agent, and is where students can indicate that they are going to engage in different SRL strategies, such as content evaluation or taking notes.

There are four pedagogical agents (PAs) that are embedded in MetaTutor. Each agent is responsible for promoting the different pillars of self-regulated learning. Gavin the Guide introduces students to the session, and administers self-report questionnaires, and the pre- and post-test. Pam the Planner represents the planning processes of SRL, and assists students with setting sub-goals and activating prior knowledge. Mary the Monitor exemplifies the monitoring processes of SRL, and assists students to engage in

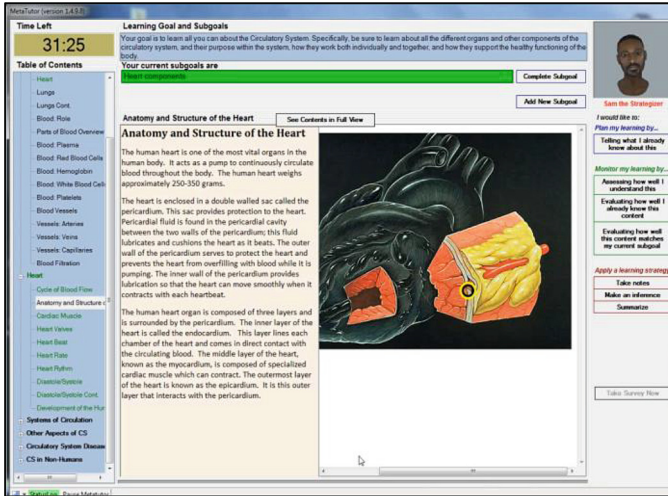


Fig. 1. Screenshot of the MetaTutor interface

metacognitive monitoring, such as judging how well they understand the material on the page, assessing how relevant the material is to the student's current sub-goal, and monitoring how far along the student is in completing his or her sub-goal. Sam the Strategizer represents the strategizing component of SRL, and emphasizes engaging in learning strategies, such as summarizing and coordinating informational sources. One agent is present at a time on screen, and this depends on what learning activity (e.g., taking notes, completing a quiz, or setting a new sub-goal) the student is performing.

### 2.3 Experimental Conditions

Before students interacted with MetaTutor, they were randomly assigned to one of two experimental conditions. In the **prompt and feedback condition**, students were provided with prompts to use SRL strategies, and received feedback on their performance on quizzes, from the pedagogical agents. During the sub-goal setting phase, Pam the Planner provided feedback regarding the student's proposed sub-goal, for example, if it was too broad. Mary the Monitor prompted students to engage in metacognitive monitoring strategies, such as judgment of learning [JOL], feeling of knowing [FOK], content evaluation [CE], and monitoring progress towards goals [MPTG]. In addition, she provided feedback on their performance using these strategies. For example, she informed them of their sub-goal quiz score, and whether or not they could proceed to the next sub-goal. Sam the Strategizer prompted students to create summaries, and provided them feedback on whether the summary was too long, too short, or done correctly. Thus, in this condition, students were heavily guided to use cognitive and metacognitive SRL strategies as they attempted to complete their self-set sub-goals and overall learning goal.

In the **control condition**, the agents were present on the screen, however they did not play an active role. For instance, when students set sub-goals, Pam the Planner simply suggested a sub-goal for them to set, instead of providing feedback, which would lead to them trying again. Mary did not prompt students to engage in metacognitive monitoring strategies, however students could choose to engage in them on their own. In addition, if students did choose to engage in these strategies, Mary did not provide them with any feedback. For example, if they completed a sub-goal quiz, she allowed them to move on to the next self-set sub-goal, regardless of their score, and without informing them of their performance on the quiz. Finally, Sam the Strategizer did not prompt students to create summaries, and if they chose to do so, he did not provide them feedback on the quality of the summary. Thus, in this condition, students acted as independent (self-regulating) learners, without guidance from the pedagogical agents. The conditions were similar with regards to the role of Gavin the Guide, who administered the same self-report measures and the same pre- and post-tests to participants. Thus, in both conditions, students were given access to the same science material and could set the same sub-goals, and so the only true difference between the conditions was the roles of the three pedagogical agents.

## 2.4 Experimental Procedure

The MetaTutor study was a 2-day experiment, which lasted for about three hours. On Day 1, students completed the consent form, and a demographics form. They were then presented with a series of self-report questionnaires, such as the Achievement Emotions Questionnaire; AEQ [11]. Finally, students were given the pre-test, which is a 30-item, four-choice option multiple-choice test on the circulatory system. The Day 1 session lasted between 30–60 min, and participants were compensated \$10 per hour. The Day 2 session involved students learning with MetaTutor. First, they were presented with an introductory video to the system, and to setting sub-goals. Next, they set two sub-goals with the assistance of Pam the Planner. They were then presented with a few more introductory videos, and when those were completed, they began the 90-minute learning session. During learning, students could have chosen which pages to navigate to, they could have read the text and viewed the diagrams, or they could have engaged in different cognitive (e.g. take notes, summarize) and metacognitive (e.g., JOL, CE) SRL processes. In addition, students could have completed their sub-goal, reprioritized their sub-goal, or added a new sub-goal. During the 90-min session, Gavin the Guide administered the Emotions and Values; EV questionnaire [11], where students reported on the emotions they were currently feeling (e.g., confusion). At the end of the session, students were presented with the post-test, which is a counterbalanced, 30-item, four-choice option multiple-choice test on the circulatory system. Students were then thanked and debriefed, and were compensated \$10 per hour for participating in this part of the study.

When students learned with MetaTutor, we collected several multi-channel data. Log-file data captured student-system interactions, such as time spent on different representations of information, scores on quizzes, pre-test and post-test scores, etc., at the millisecond level. We also collected eye-tracking data, using an SMI iView RED

120 Hz eye-tracker [12], which captured students' eye-tracking data at a rate of 120 frames per second. Students were seated 70 cm from the monitor, which was "47 × 30" cm, and has a screen resolution of 1680 × 1050 pixels. We calibrated students' eyes with a 9-point calibration, ensuring that the visual angle deviated less than 0.5° horizontally and less than 0.5° vertically. The eye-tracker captured students' gaze patterns and fixations on different areas of interest (AOI) throughout the interface.

## 2.5 Materials and Data Coding, Scoring, and Analysis

**Coding and Scoring.** To determine students' levels of prior content knowledge, we conducted a median split, based on pre-test scores, obtained from the log-files. The median score was .57 (17 out of 30), and so participants with scores higher than .57 were in the high prior content knowledge group ( $n = 73$ ), while participants with scores lower than .57 were in the low prior content knowledge group ( $n = 74$ ). To determine prior sub-goal knowledge groups, we performed another median split, based on participants' scores on the pre-test, which were relevant to the sub-goals they set during the sub-goal setting phase. Thus, students' scores differed based on their performance on the pre-test, and on the particular sub-goals they set during the sub-goal setting phase. The median score for sub-goal prior knowledge was .50, and so participants with scores higher than the median were in the high prior sub-goal knowledge group ( $n = 69$ ), and students with scores lower than the median were in the low prior sub-goal knowledge group ( $n = 78$ ). In addition, we omitted a total of 27 participants, since they had scores that were at the pre-test median (17) or the sub-goal pre-test median (.50).

To calculate participants' proportional learning gains (PLGs), we extracted their pre-test and post-test scores (i.e., total correctly answered questions out of a total of 30 questions) from the log files. To calculate proportional learning gain, we used the formula from [6]. To calculate proportional learning gains based on scores relative to sub-goals set during the initial sub-goal setting phase, we extracted the same pre-test and post-test scores from the log files, however these scores were adapted to create pre-test scores that only included questions relevant to the sub-goals, which were set during the initial sub-goal setting phase. Thus, for example, if question #2 was relevant to the first sub-goal, but not the second, the score on that question would only be included in the score for the first sub-goal. Therefore, an additional seven adapted pre-test scores were created for each participant. For our analyses, we created an average score of the sub-goal pre-test scores, based on the two sub-goals that were set during the initial sub-goal setting phase. To calculate this proportional learning gain, we used the same formula mentioned above [6], but used the average pre-test score, adapted based on the sub-goals set.

**Eye-Tracking and AOIs in MetaTutor.** To analyze students' eye-tracking, data for each eye were recorded, on an event basis, using the Dispersion-Threshold Identification algorithm, as outlined in [13], as either: (1) fixations, when the eye gaze is focused on a single point or area for more than 80 ms, and does not disperse more than 100 pixels; or (2) saccades, which are rapid eye movements between fixation points,



and thus disperse more than 100 pixels. We conducted these analyses for participants' first self-set sub-goal, which they completed. We chose the first sub-goal because the eye-tracker can only collect data up to two hours, and participants did not complete two sub-goals in two hours. As such, we wanted to ensure that we had a full data set for each participant. For this analysis, however, we will focus only on fixations, and we will not be analyzing saccades.

Using Experiment Center, version 3.4 [12], we marked nine areas of interest (AOI), based on the different interface elements of the MetaTutor environment. We coded the times participants spent on different interface layouts (e.g., normal layout, full view layout), by analyzing their screen recordings of their sessions with MetaTutor (see Fig. 2). These AOIs include: The Timer, Table of Contents, Learning Goal, currently set Sub-Goals, the Text Content, Diagram (both maximized and minimized), the Agent, the SRL Palette, and the Notes overlay. Experiment Center generated a data file for each participant, which contained the beginning and end time of each fixation, and which AOI the participant fixated on. Frequencies and mean durations for each AOI were computed from these data files, and were then converted into time periods proportional to the session, in order to control for unequal session times between participants ( $M = 77.057$  min; range: 37.15 to 121.38 min). Students may have different session times because the timer will stop when a student engages in SRL processes, thus resulting in different total session times, depending on how many processes he or she engages in. In addition, we computed the frequencies and durations for pairs of AOIs, all of which began with the content AOI, and was matched with each other AOI. Thus, we developed nine AOI pairs, and calculated the frequency of fixations of the first AOI and the second AOI together; and the proportion of time spent on the fixation pair. We chose AOI pairs (compared to triplets, etc.) due to the limit in the session time that we analyzed; i.e., we only analyzed students' eye-tracking as they completed their first self-set sub-goal, which resulted in analyzing only 56 % of the session. In addition, in extracting the AOI pairs, 324 pairs were generated. We chose to focus on the AOI pairs that began with content only, because, text is the foundation for learning about a complex science topic, and reading usually triggers monitoring, and the subsequent use of a learning strategy, etc.

### 3 Results

#### 3.1 Research Question 1: Are There Significant Differences in Proportional Learning Gains Between Prior Knowledge Groups, While Controlling for Condition?

We performed a MANCOVA, with prior content knowledge and prior sub-goal knowledge as the two independent variables, and proportional learning gain and proportional learning gain based on the self-set sub-goals, set during the sub-goal setting phase, as the two dependent variables. We used experimental condition as a covariate.

Results indicated, while controlling for condition, a significant main effect for prior sub-goal knowledge; Wilks'  $\lambda = .94$ ,  $F(2, 141) = 4.93$ ,  $p = .009$ ,  $\eta_p^2 = .07$ , but not for prior content knowledge; Wilks'  $\lambda = .98$ ,  $F(2, 141) = 1.47$ ,  $p = .23$ ,  $\eta_p^2 = .02$ .

In addition, there was no significant interaction effect; Wilks'  $\lambda = 10$ ,  $F(2, 141) = .12$ ,  $p = .88$ ,  $\eta_p^2 = .002$ . Subsequent between-subjects analyses indicated a significant effect for proportional learning gains based on sub-goals set in the initial sub-goal setting phase;  $F(1, 142) = 5.25$ ,  $p = .023$ ,  $\eta_p^2 = .036$ .

These results revealed that there is a significant difference in proportional learning gains based on sub-goals set during the sub-goal setting phase between sub-goal prior knowledge groups. More specifically, students with low prior knowledge of their sub-goals had significantly higher proportional learning gains based on their originally set sub-goals ( $M = 32.73$ ,  $SD = 29.52$ ) compared to students with high prior knowledge of their sub-goals ( $M = 11.55$ ,  $SD = 58.65$ ), while controlling for experimental condition.

### **3.2 Research Question 2: Are There Significant Differences in the Proportion of Time Spent on Different AOIs Between Prior Knowledge Groups, While Controlling for Condition?**

We performed our analyses on a subset of the dataset with eye-tracking data, since we had full eye-tracking data sets for these participants, thus yielding an  $n$  of 30 participants. For the first set of AOIs, the AOIs associated with learning, we performed a MANCOVA, with prior content knowledge and prior sub-goal knowledge as independent variables, and the proportion of time spent on the learning AOIs: text, diagram, table of contents, goals, and notes; as the dependent variables. Our covariate was experimental condition.

Results indicated, while controlling for condition, a non-significant main effect for prior content knowledge; Wilks'  $\lambda = .69$ ,  $F(5, 21) = 1.87$ ,  $p = .14$ ,  $\eta_p^2 = .31$ ; however, we did find a significant main effect for prior sub-goal knowledge; Wilks'  $\lambda = .58$ ,  $F(5, 21) = 3.00$ ,  $p = .034$ ,  $\eta_p^2 = .42$ . There was no significant interaction effect; Wilks'  $\lambda = .73$ ,  $F(5, 21) = 1.56$ ,  $p = .21$ ,  $\eta_p^2 = .27$ . Between-subjects analyses (see Fig. 2 for descriptive statistics) revealed a significant effect for the proportion of time spent on diagrams between prior sub-goal knowledge groups;  $F(1, 25) = 14.25$ ,  $p = .001$ ,  $\eta_p^2 = .36$ ; however no other between-subjects results were significant. Thus, our results revealed that students with low prior sub-goal knowledge spent a significantly higher proportion of time on the diagram AOI ( $M = .042$ ,  $SD = .031$ ), compared to participants with high prior sub-goal knowledge ( $M = .024$ ,  $SD = .018$ ), while controlling for experimental condition.

For the second set of AOIs, related to SRL strategies, we ran a MANCOVA with prior content knowledge and prior sub-goal knowledge as the independent variables, and the four AOIs related to SRL: SRL palette, timer, learning goal, and sub-goals; as the dependent variables. Again, we used experimental condition as a covariate. Results indicated no significant main effect for prior content knowledge; Wilks'  $\lambda = .83$ ,  $F(4, 22) = 1.16$ ,  $p = .36$ ,  $\eta_p^2 = .17$ ; no significant main effect for prior sub-goal knowledge; Wilks'  $\lambda = .89$ ,  $F(4, 22) = .69$ ,  $p = .61$ ,  $\eta_p^2 = .11$ ; and no significant interaction effect; Wilks'  $\lambda = .94$ ,  $F(4, 22) = .37$ ,  $p = .83$ ,  $\eta_p^2 = .063$ . Thus, there were no significant differences in the proportion of time spent fixating on the AOIs related to SRL, between prior content knowledge groups or prior sub-goal knowledge groups, while controlling for experimental condition.

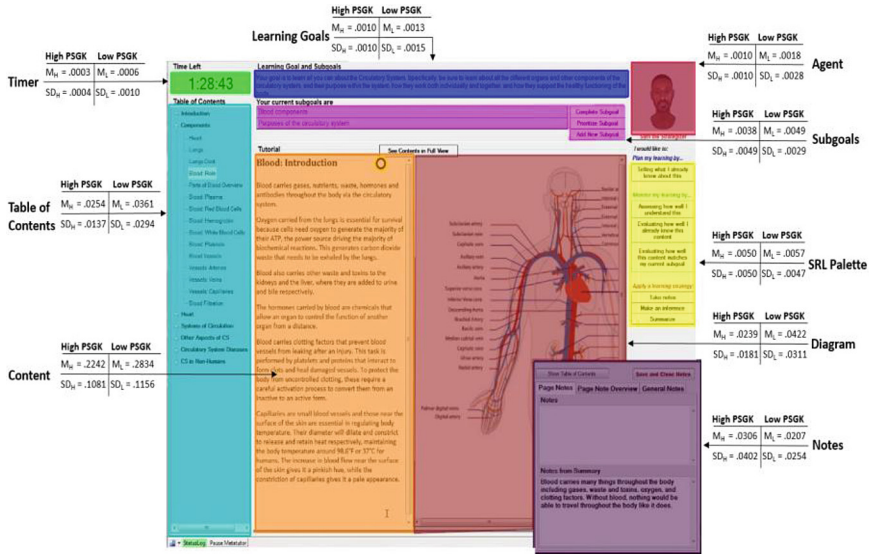


Fig. 2. Proportion of time spent on AOIs for Prior Sub-Goal Knowledge (PSGK) groups

Finally, to investigate the AOI related to the pedagogical agents, we conducted an ANCOVA, with prior content knowledge and prior sub-goal knowledge, once again, as the independent variables, the proportion of time spent on the AOI, where the pedagogical agent is located, as the dependent variable, and experimental condition as the covariate. Results revealed a non-significant main effect of prior content knowledge;  $F(1, 25) = .43, p = .52, \eta_p^2 = .017$ ; a non-significant main effect of prior sub-goal knowledge;  $F(1, 25) = 3.04, p = .094, \eta_p^2 = .11$ ; and a non-significant interaction effect;  $F(1, 25) = 1.39, p = .25, \eta_p^2 = .053$ . Therefore, these results indicated that there were no significant differences in the proportion of time spent on the agent AOI, based on prior content knowledge group or prior sub-goal knowledge group, while controlling for experimental condition.

### 3.3 Research Question 3: Are There Significant Differences in the Frequency Distribution of Fixations on AOI Pairs Between Prior Knowledge Groups?

We ran several chi-square analyses. It is important to note that due to the fact that each pair begins with the same onset, we are aware of the violation of independence for these analyses. Furthermore, we investigated frequencies of fixations on AOI pairs because it can be indicative of engaging in SRL; for example a fixation on the content to a fixation on the timer can be indicative of metacognitive monitoring. We did not assess the frequencies of fixations on a single AOI because it would not provide enough evidence of use of SRL.

We ran a total of eight chi-square tests ( $\alpha = .00625$ ) with prior content knowledge as our grouping variable, with eight different frequencies of AOI pairs. Results revealed significant differences for the content-timer AOI pair;  $\chi^2(1) = 15.11, p < .001$ ; the content-diagram AOI pair;  $\chi^2(1) = 43.99, p < .001$ ; the content-notes AOI pair;  $\chi^2(1) = 62.94, p < .001$ ; the content-agent AOI pair;  $\chi^2(1) = 8.40, p = .0038$ ; and the content-table of contents AOI pair;  $\chi^2(1) = 7.94, p = .0048$ . See Tables 1 and 2 for a summary of the frequencies and results, respectively. These findings revealed that participants with low prior content knowledge had significantly higher frequencies of fixations on the content-timer AOI pair, the content-diagram AOI pair, the content-agent AOI pair, and the content-table of contents AOI pair, compared to students with high prior content knowledge. In addition, students with high prior content knowledge had significantly higher frequencies of fixations on the content-notes AOI pair than students with low prior content knowledge.

Additionally, we ran eight other chi-square tests ( $\alpha = .00625$ ) with sub-goal prior knowledge as our grouping variable, and the same eight frequencies of AOI pairs. Results revealed a significant effect for the content-diagram pair;  $\chi^2(1) = 94.90, p < .001$ ; the content-notes AOI pair;  $\chi^2(1) = 24.001, p < .001$ ; the content-table of contents AOI pair;  $\chi^2(1) = 111.22, p < .001$ ; and the content-timer AOI pair;  $\chi^2(1) = 8.26, p = .0041$ . See Tables 1 and 2 for the frequencies and results, respectively. Overall, these findings suggest that students with low prior knowledge of their sub-goals had significantly higher frequencies of fixations on the content-diagram AOI pair, the content-table of contents AOI pair, and the content-timer AOI pair, compared to students with high prior knowledge of their sub-goals. Alternatively, participants with high prior sub-goal knowledge had significantly higher frequencies of fixations on the content-notes AOI pair than students with low prior sub-goal knowledge.

**Table 1.** Frequencies of AOI pairs based on prior content knowledge group.

	AOI pair							
	CT	CA	CD	CN	CToC	CSRL	CSG	CLG
<b>PKK</b>								
High	6	12	323	626	862	38	76	27
Low	29	31	515	375	983	65	104	31
<b>PSGK</b>								
High	9	23	278	578	696	39	77	26
Low	26	20	560	423	1149	64	103	32

*Note.* PKK = Prior Content Knowledge. PSGK = Prior Sub-Goal Knowledge. CT = Content-Timer. CA = Content-Agent. CD = Content-Diagram. CN = Content-Notes. CToC = Content-Table of Contents. CSRL = Content-SRL Palette. CSG = Content-Sub Goals. CLG = Content-Learning Goal.

**Table 2.** Chi-square results for AOI pairs based on prior knowledge.

AOI Pair	Prior content knowledge		Prior sub-goal knowledge	
	$\chi^2$	<i>p</i>	$\chi^2$	<i>p</i>
CT	15.11	.00**	8.26	.0041*
CA	8.40	.0038*	0.21	.65
CD	43.99	.00**	94.90	.00**
CN	62.94	.00**	24.001	.00**
CToC	7.94	.0048*	111.22	.00**
CSRL	7.078	.0078	6.068	.014
CSG	4.36	.037	3.76	.053
CLG	0.28	.60	0.62	.43

\**p* < .00625; \*\**p* < .001

Note. CT = Content-Timer.

CA = Content-Agent.

CD = Content-Diagram.

CN = Content-Notes.

CToC = Content-Table of Contents.

CSRL = Content-SRL Palette.

CSG = Content-Sub Goals.

CLG = Content-Learning Goal.

## 4 Implications for Designing Adaptive Hypermedia Environments

The findings from this study have implications for the design of adaptive hypermedia environments. Overall, the results revealed that there were significant differences in proportional learning gains, proportion of time spent on AOIs, and frequency of fixations on AOI pairs, between prior knowledge groups. More specifically, we found that students with low prior knowledge, of the sub-goals they set during the initial sub-goal setting phase, had significantly higher proportional learning gains and spent a significantly higher proportion of time fixating on diagrams, compared to participants with high prior knowledge of their sub-goals, set in the initial sub-goal setting phase. These findings indicate that Sam the Strategizer should engage low-prior knowledge students in a dialogue to understand the reasons for the prolonged fixations on diagrams. For example, do prolonged fixations reveal interest in the science topic (i.e., a motivational variable), additional processing time required to comprehend diagrams (compared to text), realization of a misconception, which may involve monitoring discrepancies between existing representations (in long-term memory) and diagrams presented in MetaTutor, potentially using cognitive strategies to deal with the complex nature of the diagrams, etc. In general, finding explanations for the prolonged fixations of low-prior knowledge students on diagrams is key to designing adaptive scaffolding enacted by the pedagogical agents.

Additionally, results indicated that students with low content prior knowledge and low sub-goal prior knowledge had significantly higher frequencies of fixations on some of the AOI pairs. However, the only AOI pair with a higher frequency for students with high prior content knowledge and high prior sub-goal knowledge was the content-notes AOI pair. One possible explanation of this effect is that students with high prior knowledge make more fixations to their notes because they already have knowledge of the science content, and therefore spend more time fixating on their notes either because they are taking, reviewing, or re-organizing their notes. In contrast, students with low prior knowledge, who spend more of their time fixating on the content, and other elements of the interface, such as the timer, and the diagrams, that require them to learn the science content to complete their sub-goals, and their overall learning goal. These findings have several design implications. For example, eye-tracking, as well as other process data (e.g., log-files, screen recordings of student-system interactions) need to be triangulated in order to determine what is causing high prior knowledge students to fixate from content to notes more frequently, and to differentiate between taking, reviewing, and (re-) organizing notes, since each of the note-taking behaviors have different implications for when and how pedagogical agents should facilitate, and perhaps intervene during note-taking [10]. In contrast, low prior knowledge students' more frequent fixations on several other AOI-pairs may be indicative of a lack of a strategic approach to monitoring and regulating their cognitive and metacognitive processes, and inadvertently induce extraneous cognitive load and therefore lead to inferior PLGs. As such, pedagogical agents could be designed to explicitly model and facilitate students' search and use of the various AOIs (i.e., interface elements) in order to facilitate the deployment of SRL processes and learning.

Despite the promise of eye-tracking data for enhancing adaptive instruction with ALTs [5, 6], results from this study raise several methodological and instructional issues that need to be addressed by interdisciplinary researchers, including members of the ITS community. First, it is evident that using eye-tracking data alone is not sufficient to fully capture and comprehend the underlying cognitive and metacognitive SRL processes deployed by learners during learning with ALTs, such as MetaTutor. As noted above, the inferences made from eye-tracking data alone are somewhat limited without additional process data (e.g., need to converge eye-tracking, screen recordings, etc.) to make accurate inferences regarding SRL processes to better inform design principles for improving complex learning with ALTs [16]. Second, data on AOI pairs (see Tables 1 and 2) are very interesting since they reveal significant differences on potential dyadic transitions from text to other relevant AOIs. For example, both groups made significant fixations from the science content to the diagrams, and this makes sense since learning about this topic involves the coordination of multiple representations of information [14]. Similarly, AOIs with low frequencies (e.g., content to SRL palette) may reveal that students self-regulate covertly without fixating on the SRL palette; however, we cannot determine this fully since we need other data to triangulate and improve the accuracy of this inference. In addition to including other trace data to augment eye-tracking data, we also need to explore methods from data mining and machine learning to understand the underlying processes, infer how pertinent variables (e.g., duration, sequences, and transitions of AOI transitions) are associated with other temporally aligned data (e.g., does frequent and prolonged fixation on notes allow the

differentiation between reviewing and re-organization of notes, or do we need the screen recording to provide context, does prolonged fixation duration on the agent's AOI reveal help-seeking behavior, etc.) [17]. In sum, we argue for the collection, integration, and alignment of multi-channel data to enhance our understanding of cognitive and metacognitive SRL processes as well as affective and motivational processes needed to provide students with effective adaptive instruction of complex science material [1, 15, 16].

**Acknowledgments.** This study was supported by funding from the National Science Foundation (DRL 1431552). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

## References

1. Azevedo, R., et al.: Using trace data to examine the complex roles of cognitive, metacognitive, and emotional self-regulatory processes during learning with multi-agent systems. In: Azevedo, R., Aleven, V. (eds.) *International Handbook of Metacognition and Learning Technologies*, pp. 427–449. Springer, Amsterdam (2013)
2. Graesser, A.C.: Evolution of advanced learning technologies in the 21st century. *Theor. Into Pract.* **52**, 93–101 (2013)
3. Lester, J.C., et al.: Supporting self-regulated science learning in narrative-centered learning environments. In: Azevedo, R., Aleven, V. (eds.) *International Handbook of Metacognition and Learning Technologies*, pp. 471–483. Springer, Amsterdam (2013)
4. Winne, P.H., Hadwin, A.F.: The weave of motivation and self-regulated learning. In: Schunk, D.H., Zimmerman, B.J. (eds.) *Motivation and Self-Regulated Learning: Theory, Research and Applications*, pp. 298–314. Erlbaum, New York (2008)
5. Conati, C., et al.: Understanding attention to adaptive hints in educational games: an eye-tracking study. *Int. J. Artif. Intell. Educ.* **23**, 136–161 (2013)
6. D'Mello, S.K., et al.: Gaze tutor: a gaze-reactive intelligent tutoring system. *Int. J. Hum. Comput. Stud.* **70**, 377–398 (2012)
7. Bondareva, D., Conati, C., Feyzi-Behnagh, R., Harley, J.M., Azevedo, R., Bouchet, F.: Inferring learning from gaze data during interaction with an environment to support self-regulated learning. In: Lane, H., Yacef, K., Mostow, J., Pavlik, P. (eds.) *AIED 2013. LNCS*, vol. 7926, pp. 229–238. Springer, Heidelberg (2013)
8. Jaques, N., Conati, C., Harley, J.M., Azevedo, R.: Predicting affect from gaze data during interaction with an intelligent tutoring system. In: Trausan-Matu, S., Boyer, K.E., Crosby, M., Panourgia, K. (eds.) *ITS 2014. LNCS*, vol. 8474, pp. 29–38. Springer, Heidelberg (2014)
9. Taub, M., et al.: Can the use of cognitive and metacognitive self-regulated learning strategies be predicted by learners' levels of prior knowledge in hypermedia-learning environments? *Comput. Hum. Behav.* **39**, 356–367 (2014)
10. Trevors, G., et al.: Note-taking within MetaTutor: interactions between an intelligent tutoring system and prior knowledge on note-taking and learning. *Educ. Technol. Res. Dev.* **62**, 507–528 (2014)
11. Pekrun, R., et al.: Measuring emotions in students' learning and performance: the achievement emotions questionnaire (AEQ). *Contemp. Educ. Psychol.* **36**, 36–48 (2011)

12. SMI Experiment Center 3.4.165 [Apparatus and Software]. SensoMotoric Instruments, Boston, Massachusetts, USA (2014)
13. Salvucci, D.D., Goldberg, J.H.: Identifying fixations and saccades in eye-tracking protocols. In: Duchowski, A.T. (ed.) *Eye-Tracking Research and Application*, pp. 71–78. ACM Press, Palm Beach Gardens (2000)
14. Mayer, R.E. (ed.): *The Cambridge Handbook of Multimedia Learning*, 2nd edn. Cambridge University Press, New York (2014)
15. Calvo, R.A., et al. (eds.): *The Oxford Handbook of Affective Computing*. Oxford University Press, New York (2015)
16. Azevedo, R.: Defining and measuring engagement and learning in science: conceptual, theoretical, methodological, and analytical issues. *Educ. Psychol.* **50**, 84–94 (2015)
17. Baker, R.S.: Educational data mining: an advance for intelligent systems in education. *IEEE Intell. Syst.* **29**, 78–82 (2014)



# Informing Authoring Best Practices Through an Analysis of Pedagogical Content and Student Behavior

Matthew Roy and Rohit Kumar<sup>(✉)</sup>

Raytheon BBN Technologies, Cambridge, MA, USA  
{mroy, rkumar}@bbn.com

**Abstract.** Among other factors, student behavior during learning activities is affected by the pedagogical content they are interacting with. In this paper, we analyze this effect in the context of a problem-solving based online Physics course. We use a representation of the content in terms of its position, composition and visual layout to identify eight content types that correspond to problem solving sub-tasks. Canonical examples as well as a sequence model of these tasks are presented. Student behaviors, measured in terms of activity, help-requests, mistakes and time on task, are compared across each content type. Students request more help while working through complex computational tasks and make more mistakes on tasks that apply conceptual knowledge. We discuss how these findings can inform the design of pedagogical content and authoring tools.

**Keywords:** Student behavior · Content development · Authoring · Online learning · Problem solving

## 1 Introduction

The study of student behavior in digital learning environments has been a prevalent topic in the intelligent tutoring systems research community. This line of investigation is based on the premise that improved understanding and modeling of student behavior informs the design of learning environments as well as interventions that encourage students to perform pedagogically beneficial behaviors and avoid counter-productive ones. Besides activity levels, specific behaviors like student mistakes, help-seeking [1], ‘gaming’ [2] and navigation style [3] have been the focus of research. Often factors like student’s prior knowledge, motivational disposition and affective state have been used to characterize their behavior [2, 4]. Other work has looked at the composition of learning environment [5] and nature of feedback provided by tutors embedded in those environments [6] to explain differences in behaviors.

---

This research was funded by the US Office of Naval Research (ONR) contracts N00014-12-C-0535 and N00014-16-C-0643

Our current work has focused on developing and evaluating a Physics course on a problem-solving based online learning platform described in the next section. In-class student observation and subsequent analysis of their actions indicated students spend a varying amount of time and effort while interacting with different types of pedagogical content included in the Physics course. In this paper, we investigate this observation to characterize how the content affects student behaviors. Section 3 presents an unsupervised clustering based approach for identifying different types of content authored in the course. Section 4 uses a dataset of student behavior logs to analyze differences in behaviors along various content types. Design principles for authoring problem-solving based online courses are derived from these analyses.

## 2 Problem-Solving Based Online Learning Platform

The research presented in this paper is conducted in the context of our ongoing work on the BBN Learning Platform (*Learnform*)[7], a domain-independent online learning platform used for the creation and delivery of problem-solving based learning tasks. Students learn by solving problems like the one shown in Fig. 1 below. A problem (learning task) starts out with the presentation of problem statement, shown on the left hand side of Fig. 1. While *Learnform* supports other forms of responses, all problem statements in our Physics course use a multiple choice question format.

**Equivalent resistance**

What is the equivalent resistance between points A and B?

- A. 3  $\Omega$
- B. 4  $\Omega$
- C. 10  $\Omega$
- D. 16  $\Omega$

**Choose the correct formula to calculate  $R_{eq}$**

Now that we know  $R_{eq1}$ , which is the correct formula to calculate the overall  $R_{eq}$ ?

- $R_{eq} = R_{eq1} + 4$
- $\frac{1}{R_{eq}} = \frac{1}{R_{eq1}} + \frac{1}{4}$

7/8

[Help](#)

**Fig. 1.** A problem-solving task on our learning platform

Students are allowed to solve a given problem without assistance or they can click on the help button in which case a decomposition of the problem into a sequence of steps is presented. The problem solving interface allows free-navigation through the steps i.e. the students are not required to strictly follow the steps. They can choose to skip the current step or revisit previous steps as they find it necessary to help them solve the problem. We consider this as a form of scaffolding achieved through intuitive user interface design. Furthermore, students are not forced to work through every element of every step. Rather, they are allowed free-exploration of the problem's solution to the extent allowed by the pre-designed steps. Feedback is provided for every problem-solving action and help in the form of hints are available upon request. The task concludes when the student inputs the correct answer to the problem.

The high-school level physics course available on this platform covers topics in Electricity and Magnetism (E&M). Two teachers, working part-time over four months, authored a total of 114 problems across six topics of E&M. The teachers used the workbench available on our learning platform to author these problems. The workbench comprises a WYSIWYG editor that is used to compose the problem statement and solutions steps. From an author's perspective, statements and steps are fixed sized *tiles*. In a manner akin to presentation editing software like PowerPoint and Keynote, these *tiles* are blank canvases that can be populated with elements like labels, images, text fields, combo-boxes and option boxes available from a palette. The tiles corresponding to solution steps are carefully designed to guide the students through an ideal solution to the problem. From a pedagogical point of view, the ideal solution is not the shortest or fastest solution, but one that exercises all the necessary conceptual knowledge and procedural skills along the way. As will see in the next section, this involves construction of various types of tiles that correspond to distinctive sub-tasks involved in solving Physics problems.

### 3 Analysis of Content Types

#### 3.1 Dataset

In our current work, we use content from three topics (96 problems) of our Physics course that have been thoroughly vetted through our content development process. The three topics include Electrostatics, Electric Fields and DC circuits (resistors only). The problems include a total of 644 tiles i.e. 96 problem statements and 548 solution steps. Each problem has at least two and up to thirteen solution steps. In this section, which focuses on identifying different types of content within these problems, we apply the same analysis approach on all 644 tiles.

#### 3.2 Representation

We start by representing each tile as a vector of 36 features which can be grouped into three types: Tile Position, Tile Composition and Tile Layout. Position features indicate the relative position of the tile within a problem. Statement tiles are positioned at the

front of tile deck followed by step tiles in the order they appear. Relative position feature is scaled between 0 and 100 per problem.

Elements are grouped into two types: Interactive (text fields, combo-boxes and option boxes) and Non-Interactive (labels, images). Composition features include counts and ratios of various elements available in the editor palette to construct a tile. Properties of interactive elements are further represented with features corresponding to number of choices available in combo-boxes and option boxes. Similarly, properties of non-interactive elements include number of characters included in labels.

Finally, layout feature capture geometric properties of elements such as the tile area occupied (coverage) and spread of elements on the tile. These features are indicative of visual appearance of the tile. We use entropy of element distribution among quadrants of a tile to measure spread. Low entropy indicates elements are concentrated in one corner of the tile.

### 3.3 Unsupervised Clustering

We use an off the shelf implementation [8] of an unsupervised clustering (Expectation Maximization) algorithm to identify 8 different types of content tiles. The number of clusters is automatically determined by the EM algorithm using the log-likelihood maximization criteria.

Table 1 shows the cluster distribution as well as average values of prominent position, composition and layout features across each cluster. The composition features listed in Table 1 shown proportion of text fields (TxtF), combo-boxes (Cmb) and option-boxes (Opt) out of all interactive elements on the tile. Similarly, proportions of labels (Lbl) and images (Img) out of all non-interactive elements are listed. Coverage (Cvrg) measure listed in Table 1 shows percentage of a tile’s space occupied by all (interactive as well as non-interactive) elements, where Spread measures the entropy of interactive elements.

**Table 1.** Distribution and properties of content (tile) type clusters

Content Type	Dist. %	Relative Position	Interactive			Non-Inter.		Layout	
			TxF	Cmb	Opt	Lbl	Img	Cvrg.	Spread
1. Statement	11.6	0.3	0.0	0.0	<b>100</b>	52.9	47	43.8	<i>0.04</i>
2. Annotate	10.0	29.5	0.0	<b>92.6</b>	7.4	34.6	<b>64.3</b>	35.1	0.27
3. Decide	15.0	47.6	0.0	0.0	<b>100</b>	62	38	38.8	0.11
4. Recall	<b>17.0</b>	51.0	0.0	0.0	<b>100</b>	95.4	0.9	<i>11.4</i>	0.17
5. Apply	5.7	45.2	0.0	16.8	83.2	51.0	49.0	38.9	0.21
6. Calculate	14.6	<b>81.6</b>	80.7	19.3	0	85.2	8.2	39.0	0.35
7. Compute	16.0	61.8	<b>82.3</b>	12.8	4.7	77.4	10.7	<b>59.0</b>	<b>0.44</b>
8. Interpret	10.1	66.7	50.1	4.4	45.4	77.8	20.2	50.0	0.32

### 3.4 Examination of Content Type Clusters

As listed in Table 1, each automatically identified content type was manually named post-hoc by visually examining the tiles that constitute the corresponding cluster. The cluster names identify problem solving strategies being exercised within the tile.

Hand-picked canonical examples of tiles that illustrate these content types are shown in Fig. 2. Almost all (73 out of 74) tiles clustered under the first content type are problem statements. They are the first tile of each problem. 76 % of the problem statements tiles in our dataset map to this cluster. *Statement* type tiles only use option boxes as interactive elements since they are multiple choice questions which are concentrated (low spread) towards the bottom of the tile. Consistent with standardized tests, there are usually 4 options (N for mode = 66). These tiles comprise approximately equal number of labels and images which occupy most of the space on the tile as indicated by high density.

*Annotate* type content usually comprise an image and a number of combo-boxes (dropdown type). These tiles are used to exercise a student's ability to perform mapping of entities (symbols, variables, given values, units) involved in the problem being solved to a visual representation of the problem. In Fig. 2(2), students map the given values of resistances provided in the problem statement to a circuit diagram. *Annotate* type tiles occur towards the start of a problem's solution.

Tiles in the *Decide* cluster occur towards the middle of the problem's solution. Students are asked to make an informed decision by choosing from among 2 to 4 options presented. These steps exercise skills such as the ability to develop and recall strategies for solving problem in that learning domain. Since some of the strategies presented may be sub-optimal but not completely wrong, feedback authored with such content should not only inform the student of the correctness of their response, but also provide a reason.

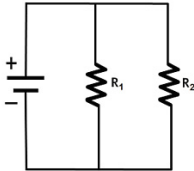
The most frequently occurring content type exercises a student's ability to *Recall* from their conceptual knowledge. This could include recalling a principle, equation or terminology. For example, in Fig. 2(4) students are asked to recall the formula for equivalent resistance in a series circuit. The recall is always facilitated by formulating it as multiple choice task, usually with 2 options (N for mode = 94 out of 109; Mean = 2.9). *Recall* tasks are visually sparse as they do not utilize a lot of area on the tile.

The least frequent content type on the other hand asks student to *Apply* their conceptual knowledge within the context of problem specific entities. These tiles can be considered to be complementary to the *Recall* type. In fact, through a visual inspect of a sample of *Apply* type tiles, we found that these tiles often comprise multiple vertically organized sections which consecutively recall and apply of conceptual knowledge.

The *Calculate* and *Compute* content types are similar in function and composition. Both correspond to procedural tasks such as reducing an equation or calculating a quantity given its relationship to other quantities. Tiles of both content types most often comprise text fields that elicit numeric values and combo-boxes that elicit physical units. The main difference between these two types is their layout. *Compute* type tiles have a denser coverage of the tile and interactive elements are more widely spread across the tile compared to any other content type.

**Equivalent resistance in a circuit**

In the circuit below,  $R_1$  has a lower resistance than  $R_2$ . The equivalent resistance of the circuit is:

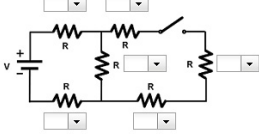


A. Less than  $R_1$   
 B. Greater than  $R_2$   
 C. The average of  $R_1$  and  $R_2$   
 D. The sum of  $R_1$  and  $R_2$

1. Statement

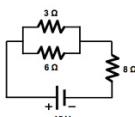
**Reduce the circuit in choice A**

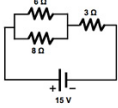
Given that the switch is open, choose Yes for the resistors that can be eliminated.

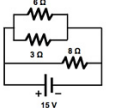


2. Annotate

**Choose the correct diagram for the problem**

A. 

B. 

C. 

3. Decide

**Choose the correct formula**

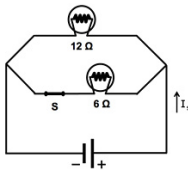
Choose the correct formula to calculate the equivalent resistance of a circuit in which the resistors are connected in series?

$R_{eq} = R_1 + R_2 + R_3 \dots$

$\frac{1}{R_{eq}} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} \dots$

4. Recall

**Calculate  $R_{eq}$  when switch  $S$  is closed**



Which is the correct option to calculate  $R_{eq}$  for this circuit when switch  $S$  is closed?

$R_{eq} = 6 + 12$

$\frac{1}{R_{eq}} = \frac{1}{6} + \frac{1}{12}$

Therefore,  
 $R_{eq} =$    $\Omega$

5. Apply

**Calculate the  $R_{eq}$  of Circuit B**

Fill in the missing values and calculate  $R_{eq}$

$$\frac{1}{R_{eq}} = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2}} + \frac{1}{\frac{1}{R_3} + \frac{1}{R_4}}$$

$R_{eq} =$    $\Omega$

6. Calculate

**Calculate the potential difference across  $R_1$ ,  $R_2$  and  $R_3$**

Fill in the missing values and calculate  $\Delta V_1$ ,  $\Delta V_2$  and  $\Delta V_3$

$V_1 = (I \text{ [ ] } \Omega) (R \text{ [ ] } \Omega)$   
 $V_1 =$   V

$V_2 = (I \text{ [ ] } \Omega) (R \text{ [ ] } \Omega)$   
 $V_2 =$   V

$V_3 = (I \text{ [ ] } \Omega) (R \text{ [ ] } \Omega)$   
 $V_3 =$   V

Fill in the known value in the table below

	$\Delta V$ (V)	I (A)	R ( $\Omega$ )	P (W)
$R_1$	<input type="text"/>	1.0	4.0	<input type="text"/>
$R_2$	<input type="text"/>	1.0	6.0	<input type="text"/>
$R_3$	<input type="text"/>	1.0	10.	<input type="text"/>
Circuit	20.	1.0	20.	<input type="text"/>

7. Compute

**Calculate potential difference of the battery**

We know the current is 1.5 A when the switch is open. Using Ohm's law, let's calculate the potential difference across the battery:

$\Delta V = I \cdot R$

$\Delta V =$   A  $\cdot$    $\Omega$

$\Delta V =$   V

Now, can we use this potential difference to solve for the current when the switch is closed?

Yes, because the potential difference across the battery remains the same regardless of switch position

No, because the potential difference across the battery would be different depending on the position of the switch

8. Interpret

Fig. 2. Canonical examples of content types identified by clustering

Finally, as the name suggests, the *Interpret* cluster helps students interpret the results of parts of the problem's solutions. Usually designed to reinforce student's intuition about a concept, these steps occur closer to end of the solution steps along with *Calculate* tiles. The interpretation task itself is often phrased as a True or False question which may be split over one or more vertically organized sections. It is not uncommon to have part of the results being interpreted be calculated within these tiles as indicated by a substantial proportion of text fields.

### 3.5 Sequence Analysis of Content Types

We apply a state sequencing algorithm developed in our previous work [9] to model the transitions between the eight content types identified previously. This algorithm sequences states based on the most frequent (i.e. mode) position of occurrence. Figure 3 illustrates this model. State size represents corresponding cluster type frequency and connector size indicates transition frequency between clusters.

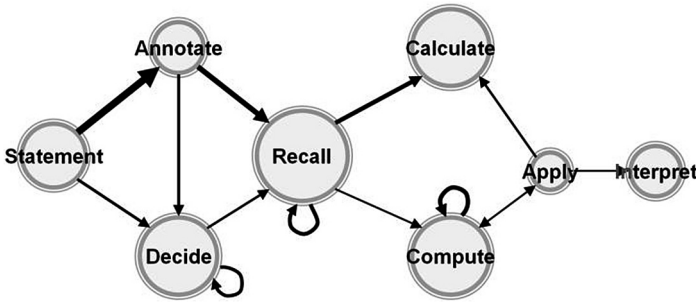


Fig. 3. Illustration of frequent sequence of content types

Starting with the problem statement, decomposition of its solutions usually follows an *Annotate*  $\rightarrow$  *Recall*  $\rightarrow$  *Calculate* pattern. *Calculate* steps are replaced by *Compute* steps in some cases. Other alternate paths taken include *Decide*  $\rightarrow$  *Recall*  $\rightarrow$  *Calculate* and augmentation of *Apply*  $\rightarrow$  *Interpret* steps in some problems after complex computation steps. In general, the above model provides a template for authoring structured problem solving tasks in the Physics domain.

## 4 Analysis of Student Behaviors

### 4.1 Dataset

We use student behavior logs collected from an in-class system evaluation conducted in April 2015 to extract measures of student behavior. 56 students (33 males, 26 females, 3 unreported; mean age = 17.3 years) enrolled in a Southern California based high school used *Learnform* as part of their regular classroom instruction of a Physics course. The students worked through problems from the Electrostatics topic during one

period, and the following week spent another period working through problems from the DC circuits topic. During these two periods, the students worked through up to 71 problems used in the content analysis in the previous section. Also, as part of our evaluation, students were administered a pre-test and post-test on the topic of DC circuits before and after their use of learning platform in the second week. We are not using the learning outcomes data from the tests in this paper.

The dataset contain different types of student actions performed during problem solving including responses to interactive elements on tiles, navigation between step tiles and help requests. 17,244 student actions were logged in the dataset used here. Student responses to interactive elements are automatically evaluated against valid answers encoded in an example-tracing tutor model which is used to determine the number of mistakes made by the students on each tile.

For the measure and analysis presented next, we have filtered this dataset to only include student actions performed during problem solving tasks that were visited by at least 5 students. After the filtering, we are left with 12,854 student actions across 36 problems (223 tiles).

## 4.2 Measures of Student Behaviors

In this work, we report four measures of student behaviors: Activity, %Help Requests, %Mistakes and Time on task. *Activity* is measured in terms of number of interactions each student has with a tile across all available interactive elements on the tile. *%Help Requests* measures the percentage of students who requested help on each tile. *%Mistakes* is measured as percentage of incorrect responses (mistakes) on each tile. Finally, *Time On-Task* measures, in seconds, the amount of time spent by each student on interacting with a tile. Specifically, it is calculated as the time spent between interactions with elements on a tile.

Table 2 presents averages of these measures for the content type clusters identified in the previous section. Also listed is the average number of students who interacted with tiles of each content type.

**Table 2.** Measures of student behaviors averaged across content types

Content type	#Students	Activity	%Help Requests	% Mistakes	Time on-task
1. Statement	41.6	2.5	27.2	48.8	8.7
2. Annotate	13.9	3.3	5.0	8.6	21.6
3. Decide	16.6	1.7	6.6	16.6	15.2
4. Recall	17.1	1.6	9.4	22.3	16.2
5. Apply	17.1	2.2	13.5	25.4	14.7
6. Calculate	5.6	5.5	1.8	23.5	52.3
7. Compute	8.8	7.2	50.0	28.2	69.3
8. Interpret	7.6	4.2	10.5	24.4	35.8



### 4.3 Observations

Using the number of students who interact with problem statements as the benchmark, we notice a steep drop off in the number of students who interact with solution steps offered to them. However, of the students who do interact with the decomposed solution offered on our system, many of them choose to work on sub-tasks that help them *Decide* on a problem solving strategy, *Recall* relevant conceptual knowledge and *Apply* those concepts. This is indicative of the types of steps that students find most helpful in solving the given problem.

Intuitively, most students appear to not interact with steps involving simple calculations which many students may be able to perform “in-their-head” also indicated by the least number of students requesting help on those tiles even though they make a substantial number of mistakes. However, we also notice the students who do choose to interact with *Calculate* type steps spend a lot of time (52 s) and effort (5.5 responses per tile) on those tiles, while making a substantial number of mistakes (23 %). This pattern of high activity (7.2 responses), large amount of time spent (69 s) and mistakes (28 %) is amplified in *Compute* cluster which are essentially a more complex variant of the *Calculate* steps. However, the complexity is sufficient to change the students’ metacognitive disposition to request more help. From an authoring point of view, help, in the form of hints, should be offered on these sub-tasks to encourage increased task completion [10].

In terms of the other mistakes made by the students, we see that almost half (49 %) of the interactions with problem statements are mistakes which indicates that the subject matter (electricity and magnetism) was a good match for the students enrolled in this course i.e. it was neither too easy that the students did not need any help, nor completely insurmountable without resorting to guessing.

The small number of help request and mistakes made on *Annotate* type tiles while spending a noticeable amount of time on those tiles might be indicative of the limited pedagogical merit of using this type of sub-tasks in solution decomposition. While it may be necessary to incorporate them occasionally to ground facts related to the problem before proceeding down to the rest of the solution, it could be worthwhile to design less time consuming ways of achieving the same effect.

## 5 Conclusion

Content analysis of problem solving tasks in the Physics domain has been conducted in prior work [11] which took a qualitative approach to content categorization. In contrast to that, the quantitative nature of the feature-based content analysis approach described in this work makes it applicable to other domains where problem solving based learning tasks are employed. In addition to offering insights about the type of sub-tasks that subject matter experts who authored solution decompositions generated, sequence analysis presented here informs best practices. Specifically, templates based on each of the content type and their sequences can be included within authoring tools to accelerate the content development process. Furthermore, clustering based automated

content categorization and sequence model developed with the methodology presented in this paper could guide diagnostics build into authoring tools [12].

This paper also studies the *missing link* of how content and its layout influences student behavior in terms of their interaction with learning systems. While this is a novel extension to contemporary work on student modeling, historically, similar investigation was pursued in terms of observable measures of cognitive load like *task completion time* while performing paper based problem solving tasks [13]. The obvious next step is analyzing the effect of learner's interactions with different content types on learning outcomes. While that analysis is beyond the scope of this paper, the analysis presented here identifies problem solving sub-tasks that elicit many misconceptions and help request (or lack thereof). Quantified differences in student behavior across these content types inform authoring practices. Specifically, *Apply* and *Compute* type content should offer hints on every interactive element whereas *Recall*, *Calculate* and *Interpret* steps should provide not only informational but also corrective feedback when students make mistakes.

## References

1. Alevan, V., Roll, I., McLaren, B.M., Koedinger, K.R.: Automated, unobtrusive, action-by-action assessment of self-regulation during learning with an intelligent tutoring system. *Educ. Psychol.* **45**(4), 226–233 (2010)
2. Baker, R., Walonoski, J., Heffernan, N., Roll, I., Corbett, A., Koedinger, K.: Why students engage in “Gaming the System” behavior in interactive learning environments. *J. Interact. Learn. Res.* **19**(2), 185–224 (2008)
3. Cha, H.J., Kim, Y.S., Park, S.H., Yoon, T.B., Jung, Y.M., Lee, J.H.: Learning styles diagnosis based on user interface behaviors for the customization of learning interfaces in an intelligent tutoring system. In: Ikeda, M., Ashley, K.D., Chan, T.W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 513–524. Springer, Heidelberg (2006)
4. Beal, C.R., Qu, L., Lee, H.: Mathematics motivation and achievement as predictors of high school students' guessing and help-seeking with instructional software. *J. Comput. Assist. Learn.* **24**(6), 507–514 (2008)
5. Walonoski, J.A., Heffernan, N.T.: Prevention of off-task gaming behavior in intelligent tutoring systems. In: Ikeda, M., Ashley, K.D., Chan, T.W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 722–724. Springer, Heidelberg (2006)
6. Koedinger, K.R., Alevan, V.: Exploring the assistance dilemma in experiments with cognitive tutors. *Educ. Psychol. Rev.* **19**(3), 239–264 (2007)
7. Kumar, R., Chung, G.K., Madni, A., Roberts, B.: First evaluation of the physics instantiation of a problem-solving-based online learning platform. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M.F. (eds.) AIED 2015. LNCS, vol. 9112, pp. 686–689. Springer, Heidelberg (2015)
8. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. *SIGKDD Explor.* **11**(1), 10–18 (2009)
9. Kumar, R., Roy, M., Roberts, R.B., Makhoul, J.I.: Towards automatically building tutor models. In: International Conference on Intelligent Tutoring Systems (2014)
10. Stamper, J., Eagle, M., Barnes, T., Croy, M.: Experimental evaluation of automatic hint generation for a logic tutor. *Int. J. Artif. Intell. Educ.* **22**, 3–17 (2013)

11. Chi, M.T.H., Feltovich, P.J., Glaser, R.: Categorization and representation of physics problems by experts and novices. *Cogn. Sci.* **5**, 121–152 (1981)
12. Kumar, R., Roy, M.E, Pattison-Gordon, E., Roberts, R.B.: General purpose ITS development tools. Workshop on Intelligent Tutoring System Authoring Tools, 12th International Conference on Intelligent Tutoring Systems (ITS 2014), Honolulu, HI (2014)
13. Sweller, J.: Cognitive load during problem solving: effects on learning. *Cogn. Sci.* **12**, 257–285 (1988)

# Timing Game-Based Practice in a Reading Comprehension Strategy Tutor

Matthew E. Jacovina<sup>1</sup>(✉), G. Tanner Jackson<sup>2</sup>, Erica L. Snow<sup>3</sup>,  
and Danielle S. McNamara<sup>1</sup>

<sup>1</sup> Institute for the Science of Teaching and Learning, Arizona State University,  
Tempe, AZ 85287, USA

{Matthew.Jacovina, Danielle.McNamara}@asu.edu

<sup>2</sup> Cognitive Science, Educational Testing Service, Princeton, NJ 08541, USA  
gtjackson@ets.org

<sup>3</sup> SRI International, Menlo Park, CA 94025, USA  
erica.snow@sri.com

**Abstract.** Game-based practice within Intelligent Tutoring Systems (ITSs) can be optimized by examining how properties of practice activities influence learning outcomes and motivation. In the current study, we manipulated *when* game-based practice was available to students. All students ( $n = 149$ ) first completed lesson videos in iSTART-2, an ITS focusing on reading comprehension strategies. They then practiced with iSTART-2 for two 2-hour sessions. Students' first session was either in a game or nongame practice environment. In the second session, they either switched to the alternate environment or remained in the same environment. Students' comprehension was tested at pretest and posttest, and motivational measures were collected. Overall, students' comprehension increased from pretest to posttest. Effect sizes of the pretest to posttest gain suggested that switching from the game to nongame environment was least effective, while switching from a nongame to game environment or remaining in the game environment was more effective. However, these differences between the practice conditions were not statistically significant, either on comprehension or motivation measures, suggesting that for iSTART-2, the timing of game-based practice availability does not substantially impact students' experience in the system.

**Keywords:** Game-based learning · Intelligent Tutoring Systems · Comprehension · Motivation

## 1 Introduction

Intelligent Tutoring Systems (ITSs) have produced positive outcomes for students across a number of domains [1]. The individualized instruction offered by ITSs is most successful when students engage in extended practice. Unfortunately, students often become disengaged and bored while using ITSs [2]. Enhancing students' motivation to persist in their use of these systems without sacrificing educational benefits has thus been an ongoing challenge for developers. Implementing educational games and game-like features is one method for increasing students' interest in practicing within

tutoring systems [3]. Games aim to leverage students' enjoyment to both increase persistence in practice and encourage deep and meaningful interactions with the content of the game [4]. Research on the addition of game features to nongame environments in order to improve user experience has become an increasingly hot field of study. Despite attracting attention from fields such as marketing and health, however, there are many gaps left to be filled in understanding the impact of game-features [5].

The study of ITSs has not reached a consensus on the efficacy of educational games and game-like features. Clearly, games are not a panacea for all educational goals and contexts, and their use must be tested broadly and with multiple implementations. For example, the influence of games has been studied in contexts ranging from classrooms [6] to military training [7], all with some degree of success. Unsurprisingly, though, not all game features are equally compelling or appropriate for different goals [3, 8]. Moreover, game features may serve to distract some students from the pedagogical goals of a system [9–11].

An important aspect of testing the effectiveness of educational games is determining which specific properties of the gaming experience are important for educational outcomes and motivation. This can allow developers to make informed decisions about how to implement game features. For example, one study examined the effect of making an educational game single-player or multiplayer, and found no differences on knowledge acquisition or perceptions of the activity [12]. Design analyses of popular games can also be conducted to extract key properties of positive gaming experiences. In an analysis of the puzzle game *Candy Crush Saga*, for example, Varonis and Varonis [13] identified several important aspects of the game, such as the requirement for iterative innovation, providing immediate feedback, giving bonuses for exemplary performance, and allowing players to engage in alternative activities in between engaging with the main game.

In addition to game *features*, the *timing* of game-based practice availability may be an important factor. Given the mixed results in the literature on the effectiveness of educational games [14], one possibility is that game-based practice best serves students at particular time points. For example, a game designed to teach the programming concept of *loops* was found to be more effective when played before a more traditional assignment on the topic than after the traditional assignment [15]. This game was tightly integrated with the learning material, potentially making it immediately effective. For systems that add game features to educational activities that may distract from learning, having all features available immediately may be undesirable.

## 1.1 iSTART-2

The current study was conducted using the Interactive Strategy Training for Active Reading and Thinking-2 (iSTART-2) system. iSTART-2 is a game-based tutoring system that provides reading comprehension instruction by teaching self-explanation strategy lessons and strategy practice games [16, 17]. iSTART-2 provides 8<sup>th</sup> grade through college students with strategies designed to help them construct deep and meaningful text representations. This is an important academic skill and one that is difficult for many readers [18]. Although the strategy lessons and practice activities are

the driving forces in helping students improve, other system features (e.g., game-based practice) may help to motivate students and indirectly improve comprehension. However, the game features do not directly teach self-explanation skills. Thus, a key goal for iSTART-2 is to include game features that increase motivation but do not distract from practicing self-explanation strategies.

Previous work has compared a game-based version of iSTART to a nongame based system, and found that students equally benefitted from the two versions of the system [19]. Another study showed that across time, a game-based version of iSTART yielded higher enjoyment and motivation than a nongame version [16]. This research suggests iSTART-2's game-based practice may be appropriately tuned to enhance motivation without decreasing learning. However, these findings do not confirm that learning and motivation have been optimized. Varying the availability of game and nongame activities may further enhance outcomes. Specifically, early exposure to nongame practice followed by access to game-based practice may afford students an uninterrupted introduction to practice activities, and then introduce motivational features that motivate their continued effort.

## 1.2 Current Study

In this study we aimed to determine *when* to make game-based practice available to students within the iSTART-2 practice environment. All students in this study began by watching lesson videos and answering checkpoint questions for each. During two subsequent study sessions, students practiced within iSTART-2 for two hours. Students were randomly assigned to begin their first practice session in either a game or nongame environment. During students' second practice session, they either continued in the same environment or switched to the alternate environment. This created a total of four conditions across the 2 (Initial Practice: Game or Nongame)  $\times$  2 (Practice Consistency: Switch or Stay) experimental design.

The game and nongame practice environments differed primarily on the presence of game features within the practice activities. In both environments, students had access to one generative activity and one identification activity (see Fig. 1).

In generative activities, students read science texts and write self-explanations in response to predefined target sentences. After submitting their self-explanation, students receive an automated score for the quality of their response [20]. *Map Conquest*, in the game environment, allowed students to use the points they earned through their self-explanations to attempt to "conquer" a game board against computer opponents. *Coached Practice*, in the nongame environment, assigned scores to students' self-explanations, but these scores did not relate to a game activity. However, Coached Practice did offer additional feedback and suggestions to improve the quality of students' self-explanation in the form of verbal responses from a pedagogical agent.

In identification activities, students read self-explanations that are ostensibly written by other students. The students' task is to identify which iSTART-2 strategy was used to generate that self-explanation. All students receive feedback on the accuracy of their choices. *Bridge Builder*, in the game environment, also gives points to students, with point bonuses for consecutive correct answers. A simple narrative also plays out as

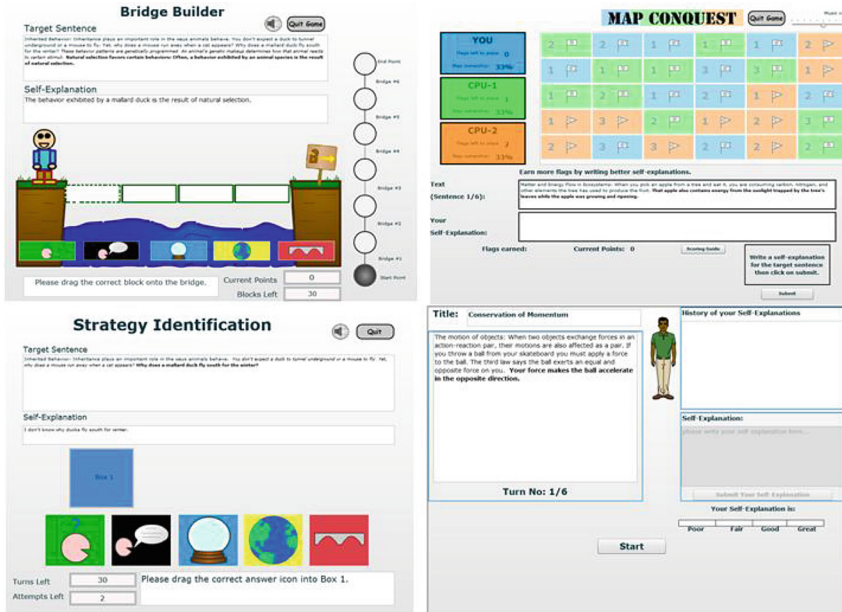


Fig. 1. The practice activities in the game (top row) and nongame (bottom row) environments

students give correct answers, allowing an explorer to cross a bridge in search of treasure. *Strategy Identification*, in the nongame environment, only gives accuracy feedback. Within the game environment, students can use the points they earned to modify the background color of the site, purchase new hair styles and colors for an avatar, and track their achievements through a list of trophies that they win in the games. These features are not available in the nongame environment.

Both practice environments were thus nearly identical in terms of the educational content, but the game environment includes features intended to enhance students' experience. Learning was measured comparing pretest and posttest performance on open-ended comprehension questions for a science text. Half of the comprehension questions were textbase questions and half were bridging inference questions. Textbase questions require readers to remember information that was directly stated in one sentence of the text, whereas bridging inference questions require that readers integrate information across multiple sentences in a text.

Our first hypothesis (H1) was that students would perform better at posttest than pretest across both question types and across all practice conditions. Improvement from pretest to posttest on the comprehension measure is indicative of the benefits of iSTART on students' ability to comprehend challenging content-area texts.

Two alternative hypotheses center on how the timing of availability for these features might influence learning. Hypothesis 2a (H2a) was that practice in the game environment would on average be more beneficial than practice in the nongame environment. This hypothesis is plausible given that past research has shown benefits for game-based practice [6, 7]. Hypothesis 2b (H2b) was that the timing of game-based

practice would influence pretest to posttest gain. This hypothesis is based on findings that game-based practice may impede performance [9–11]. Specifically, this leads to the prediction that larger benefits would be observed when students switch from a nongame to game environment, because students receive unadulterated practice early in the learning process, and then obtain access to motivating game features. Smaller benefits would be observed, however, when switching from a game to nongame environment, because during the second session, the system fails to meet students' expectations for game-based practice.

Our third and fourth hypotheses center on how the timing of availability for game-based features influenced dimensions of motivation, which should be related to posttest performance. Three dimensions of motivation were measured at posttest: students' reported effort exerted while using iSTART-2, their perception of their performance quality, and their emotional state at posttest. To confirm that these dimensions were related to performance, hypothesis 3 (H3) was that each dimension would correlate with posttest performance. Hypothesis 4 (H4) was that an interaction would emerge between initial practice environment and practice consistency. Specifically, we predicted that students would report higher scores on the motivational dimensions when their second session was a game environment, and report the lowest scores when switching from a game to a nongame environment. Thus, our prediction was that beginning in a nongame environment and switching to a game-environment would, overall, be the optimal condition. This condition initially provides practice without the distraction of games, and follows with game-based practice in the second practice session when students' motivation may have decreased.

## 2 Method

### 2.1 Participants

This study included 149 high school students and recent high school graduates from the Southwest United States. These students were, on average, 16.22 years of age (range: 13–20 years), with the majority of students reporting their grade level as high school seniors or sophomores. Of the 149 students, 55 % self-identified as female; 43.6 % self-identified as Caucasian, 32.2 % as Hispanic, 8.7 % as African-American, 7.4 % as Asian, and 8.1 % as another ethnicity. Seven students dropped out of the study before the final session and their data were not included in these analyses; one additional student's data were removed from analyses due to technical problems with the pretest survey.

### 2.2 Materials

The pretest and posttest included measures of reading comprehension skill and motivation. Reading comprehension skill was assessed through comprehension questions based on two science passages. The presentation order of the texts (pretest or posttest) was counterbalanced across students. The texts and questions were modified from those used in previous research [16, 21]. The texts were selected for their similar length (311



and 283 words), Flesch-Kincaid grade level (8 and 9), and linguistic features as measured by the natural language processing tool, Coh-Metrix [22]. While reading each text, students were prompted to self-explain 9 sentences. For each text, there were 8 open-ended questions, including 4 textbase questions and 4 bridging inference questions. The text was not on screen while students answered these questions. The answers to textbase questions were found within a single sentence of the text, whereas the answers to bridging inference questions required students to integrate information between two or more sentences. Each question could receive a maximum of 1 point, with some questions allowing for partial credit. Two coders independently scored at least 14 % of the responses for each question, resolved discrepancies, and iterated on this process until they achieved 95 % exact agreement (all kappa values above 0.8). After achieving agreement on a question, one coder completed the scoring.

Pretest motivation was assessed using the learning intentions, self-efficacy, and emotional state dimensions of a modified version of the Online Motivation Questionnaire [OMQ; 23]. Posttest motivation was also assessed using an adapted version of the OMQ, and included the dimensions of reported effort, result assessment, and emotional state.

### 2.3 Procedure

This project was part of a 5-session study, which lasted approximately 8.5 h in total. Each session was completed on a different day to avoid fatigue. In session 1, students completed demographic surveys and a writing task that is unrelated to the current study. During session 2, students completed pretest measures, including the reading comprehension questions and the pretest OMQ questions. Students then completed the iSTART-2 lesson videos. During both sessions 3 and 4, students engaged with the iSTART-2 practice interface for 2 h. These practice sessions were controlled for time and not the activities with which students engaged. The initial practice environment (game or nongame) and practice consistency (whether the environment switched or stayed the same between session 3 and 4) varied depending on students' randomly assigned condition. During session 5, students completed a posttest, which included the reading comprehension test and the OMQ questions.

## 3 Results

Analyses were conducted to examine the effects of initial practice (game or nongame) and practice consistency (switch or stay) on comprehension scores and motivation measures.

### 3.1 Comprehension Scores

To determine the effects of iSTART-2 training and practice condition on comprehension scores, a mixed ANOVA was conducted with test (pretest, posttest) and question type (textbase, bridging) as within-participant factors, and initial practice (game,

nongame), and practice consistency (switch, stay) as between-participant factors. Comprehension scores are reported as the percentage of total possible points that a student achieved (see Table 1). A main effect of question type emerged such that students scored higher on textbase questions than on bridging questions [ $F(1, 137) = 26.99, p < .001, \eta_p^2 = .165$ ]. This finding serves as a confirmation that the bridging questions were more difficult to answer. A main effect of test also emerged such that students scored higher at posttest than at pretest [ $F(1, 137) = 5.02, p = .027, \eta_p^2 = .035$ ]. This finding thus supported H1.

**Table 1.** Pretest and posttest means (and *SD*) for textbase and bridging questions.

	Pretest ( <i>SD</i> )	Posttest ( <i>SD</i> )	Mean ( <i>SD</i> )
Textbase questions	48.2 % (29.9)	51.6 % (31.0)	49.9 % (26.3)
Bridging questions	39.8 % (24.7)	44.3 % (25.1)	42.1 % (21.6)
Mean ( <i>SD</i> )	44.0 % (24.2)	47.9 % (25.0)	

No main effects or interactions involving the two practice condition factors, initial practice or practice consistency, were significant, failing to support H2a or H2b. The lack of interactions involving test, question type and practice conditions suggests that gains for both textbase and bridging questions were similar across conditions. Table 2 displays the pretest and posttest mean scores for each condition as well as the effect size of the pretest to posttest improvement. Although an interaction did not emerge between the conditions, the pretest to posttest gain were highest when students remained in a game environment (i.e.,  $\eta_p^2 = .081$ ) or switched from a nongame environment to a game environment (i.e.,  $\eta_p^2 = .069$ ), and lowest when students switched from a game environment to a nongame environment (i.e.,  $\eta_p^2 = .003$ ). This pattern is partially consistent with H2b in that switching from a game to a nongame environment led to a lower gain while switching from a nongame to game environment led to a higher gain. However, this may be attributable to pretest differences.

**Table 2.** Partial eta squared values for the pretest to posttest gain for each of the four conditions.

	Initial practice: game			Initial practice: nongame		
	Pretest	Posttest	Effect size	Pretest	Posttest	Effect size
Switch practice environments	47.1 %	48.4 %	$\eta_p^2 = .003$	46.0 %	51.2 %	$\eta_p^2 = .069$
Stay in practice environment	40.0 %	46.0 %	$\eta_p^2 = .081$	43.4 %	46.3 %	$\eta_p^2 = .025$

### 3.2 Motivation Measures

Table 3 displays correlations between posttest comprehension scores and the pretest and posttest OMQ dimensions of interest. All OMQ motivation dimensions were significantly correlated with posttest performance, supporting H3. To test the effects of practice condition on posttest motivation, between-participant ANCOVAs were conducted with the

three posttest OMQ dimensions serving as dependent variables: reported effort, performance assessment, and emotional state. Initial practice (game, non-game) and practice consistency (switch, stay) were between-participant factors. Pretest OMQ dimensions (learning intentions, self-efficacy, and emotional state) served as covariates to control for pretest differences across conditions that emerged despite random assignment. However, no main effects or interactions emerged for initial practice or practice consistency (all  $F_s < 2.6$ ,  $p_s > .10$ ). This suggests that practice condition did not influence these dimensions of posttest motivation, failing to support H4.

**Table 3.** Correlations between comprehension scores and motivation measures.

Measure	1	2	3	4	5	6	7
1. Post comprehension	-						
2. Pre learning intentions	.28**	-					
3. Pre self-efficacy	.28**	.35**	-				
4. Pre emotional state	.17*	.22**	.27**	-			
5. Post reported effort	.39**	.28**	.23**	.16	-		
6. Post result assessment	.33**	.25**	.42**	.22**	.57**	-	
7. Post emotional state	.32**	.32**	.34**	.43**	.39**	.37**	-

\* $p < .05$ , \*\*  $p < .01$ .

## 4 Conclusions

In this study we examined how the timing of game-based practice availability influenced performance and motivation. After completing instructional videos, students spent 2 two-hour sessions in iSTART-2 practice environments. Students were randomly assigned to begin in a game-based or nongame environment; half of the students stayed in the same environment during the second practice session and half switched to the other environment. Overall, we found that students’ scores on comprehension questions improved from pretest to posttest, supporting H1. Consistent with past work, these results support the notion that iSTART-2 benefits students’ reading comprehension. No effects of initial practice or practice consistency emerged, failing to support H2a or H2b. Students’ overall benefits were approximately equivalent regardless of whether they began in a game or nongame practice environment, or whether they switched or stayed in the same environment. However, the effect sizes of the pretest to posttest gain were partially consistent with H2b, in that switching from a game to a nongame environment was least effective, while switching from a nongame to game environment was more effective. Remaining in a game environment also led to a large effect size. All motivation dimensions were positively correlated with posttest comprehension performance, supporting H3. This suggests that testing for effects of practice condition on these motivation measures is worthwhile. However, the dimensions of motivation were not influenced by condition, failing to support H4. Reports of effort and performance quality, and posttest emotional state did not seem to be influenced by the timing of game-based feature availability.

These results align with a past study using iSTART-2 that compared students' self-explanation quality after 45 min of practicing in a game-like or less game-like activity, and found no overall difference [24]. For iSTART-2, one possibility is that the impact of individual game features is small compared to the overall impact of a system that affords students *agency* over their learning through choices of practice activities [17]. An additional possibility is that the outcome measures included in this study were not sufficiently sensitive. Future analyses examining interaction patterns within iSTART-2 may uncover differences between practice conditions. Moreover, posttest motivation was measured during a separate session to capture students' overall experience. Testing students' motivation more frequently, perhaps during and at the conclusion of each session, may capture changes in motivation over time that this study could not. In classrooms, behavioral measures may serve as proxies for motivation, such as how frequently students practice outside of class assignments.

Overall, the findings in the current project provide support for the effectiveness of iSTART-2. Although the results do not provide strong evidence for when game-based practice should be made available in iSTART-2, the pretest to posttest gains across conditions suggest that students should either be provided consistent access to games or should begin with nongame practice and then transition to game-based practice. Future work will continue to explore the features of game-based practice and its timing, perhaps over longer periods of time that include the gradual release of more than two games, in order to optimize students' experience within iSTART-2.

**Acknowledgments.** This research was supported in part by the Institute for Educational Sciences (IES R305A130124). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the IES. We thank the many colleagues and students who have contributed to this work, and extend a special thanks to Tricia Guerrero for her help in coding data for this project.

## References

1. Steenbergen-Hu, S., Cooper, H.: A meta-analysis of the effectiveness of intelligent tutoring systems on college students' academic learning. *J. Educ. Psychol.* **106**, 331–347 (2014)
2. D'Mello, S., Olney, A., Williams, C., Hays, P.: Gaze tutor: a gaze-reactive intelligent tutoring system. *Int. J. Hum Comput Stud.* **70**, 377–398 (2012)
3. McNamara, D.S., Jackson, G.T., Graesser, A.C.: Intelligent tutoring and games (ITaG). In: Baek, Y.K. (ed.) *Gaming for Classroom-Based Learning: Digital Roleplaying as a Motivator of Study*. IGI Global, Hershey (2010)
4. Gee, J.P.: *What Video Games Have to Teach Us About Learning and Literacy*. Palgrave Macmillan, New York (2003)
5. Richter, G., Raban, D.R., Rafaei, S.: Studying gamification: the effect of rewards and incentives on motivation. In: Reiners, T., Wood, L. (eds.) *Gamification in Education and Business*, pp. 21–46. Springer International Publishing, Switzerland (2015)
6. Papastergiou, M.: Digital game-based learning in high school computer science education: impact on educational effectiveness and student motivation. *Comput. Educ.* **52**, 1–12 (2009)
7. Belanich, J., Orvis, K.L., Sibley, D.E.: PC-based game features that influence instruction and learner motivation. *Mil. Psychol.* **25**, 206–217 (2013)

8. Amory, A., Naicker, K., Vincent, J., Adams, C.: The use of computer games as an educational tool: identification of appropriate game types and game elements. *Br. J. Educ. Technol.* **30**, 311–321 (1999)
9. Adams, D.M., Mayer, R.E., MacNamara, A., Koenig, A., Wainess, R.: Narrative games for learning: testing the discovery and narrative hypotheses. *J. Educ. Psychol.* **104**, 235–249 (2012)
10. Rieber, L.P., Noah, D.: Games, simulations, and visual metaphors in education: antagonism between enjoyment and learning. *Educ. Media Int.* **45**, 77–92 (2008)
11. Jackson, G.T., Dempsey, K.B., McNamara, D.S.: Game-based practice in reading strategy tutoring system: showdown in iSTART-ME. In: Reinders, H. (ed.) *Computer games*, pp. 115–138. Multilingual Matters, Bristol (2012)
12. Tsai, F.H., Tsai, C.C., Lin, K.Y.: The evaluation of different gaming modes and feedback types on game-based formative assessment in an online learning environment. *Comput. Educ.* **81**, 259–269 (2015)
13. Varonis, E.M., Varonis, M.E.: Deconstructing candy crush: what instructional design can learn from game design. *Int. J. Inf. Learn. Technol.* **32**, 150–164 (2015)
14. Wouters, P., van Nimwegen, C., van Oostendorp, H., van der Spek, E.D.: A meta-analysis of the cognitive and motivational effects of serious games. *J. Educ. Psychol.* **105**, 249–265 (2013)
15. Eagle, M., Barnes, T.: Evaluation of a game-based lab assignment. In: *Proceedings of the 4th International Conference on Foundations of Digital Games (FDG 2009)*, pp. 64–70. ACM, New York, NY (2009)
16. Jackson, G.T., McNamara, D.S.: Motivation and performance in a game-based intelligent tutoring system. *J. Educ. Psychol.* **105**, 1036–1049 (2013)
17. Snow, E.L., Allen, L.K., Jacovina, M.E., McNamara, D.S.: Does agency matter?: exploring the impact of controlled behaviors within a game-based environment. *Comput. Educ.* **26**, 378–392 (2014)
18. McNamara, D.S., Magliano, J.P.: Towards a comprehensive model of comprehension. In: Ross, B. (ed.) *The Psychology of Learning and Motivation*, vol. 51. Elsevier Science, New York (2009)
19. Jackson, G.T., Varner (Allen), L.K., Boonthum-Denecke, C., McNamara, D.S.: The Impact of individual differences on learning with an educational game and a traditional ITS. *Int. J. Learn. Technol.* **8**, 315–336 (2013)
20. Jackson, G.T., Guess, R.H., McNamara, D.S.: Assessing cognitively complex strategy use in an untrained domain. *Top. Cogn. Sci.* **2**, 127–137 (2010)
21. McNamara, D.S., O'Reilly, T., Best, R., Ozuru, Y.: Improving adolescent students' reading comprehension with iSTART. *J. Educ. Comput. Res.* **34**, 147–171 (2006)
22. McNamara, D.S., Graesser, A.C., McCarthy, P., Cai, Z.: *Automated Evaluation of Text and Discourse with Coh-Metrix*. Cambridge University Press, Cambridge (2014)
23. Boekaerts, M.: The on-line motivation questionnaire: a self-report instrument to assess students' context sensitivity. *New Dir. Measures Methods* **12**, 77–120 (2002)
24. Jacovina, M.E., Snow, E.L., Jackson, G., McNamara, D.S.: Game features and individual differences: interactive effects on motivation and performance. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M. (eds.) *AIED 2015. LNCS*, vol. 9112, pp. 642–645. Springer, Heidelberg (2015)

# Evaluation of the Formal Models for the Socratic Method

Nguyen-Thanh Le<sup>(✉)</sup> and Nico Huse

Humboldt-Universität zu Berlin, Berlin, Germany  
nguyen-thinh.le@hu-berlin.de, nico.huse@googlemail.com

**Abstract.** In this paper, we report results of an evaluation study that investigate the applicability and usefulness of the formal models of the Socratic Method. Nelson suggested that the Socratic Method, which is employed in teaching consists of three phases: searching for examples, searching for attributes and generalizing the attributes. These formal models are intended to serve in a computerized learning environment where users can train with a chatbot to stimulate their critical thinking. This paper demonstrates the applicability and the usefulness of the formal models and shows its effectiveness in group discussion where the chatbot acts as a discussion leader who applies the Socratic Method. The contribution of this paper is two-fold. First, in the dialogue models, we integrated critical questions using the question taxonomy of Paul and Elder in the three phases of the Socratic Method. Second, the formalization of the three phases of the Socratic Method using state diagrams is a new innovation.

**Keywords:** Socratic questioning · Socratic method · Critical thinking · Dialogue models

## 1 Introduction

Socratic dialogue is considered an effective teaching approach that is used to stimulate critical thinking (Paul and Elder 2007). Paul defined “critical thinking” as: “Thinking explicitly aimed at well-founded judgement, utilizing appropriate evaluative standards in an attempt to determine the true worth, merit, or value of something.” Based on the definition of “critical thinking” Paul argued that “critical thinking provides us with definitive and specific tools”. He linked the relationship between critical thinking and the Socratic Method as follows: “Critical thinking [...] is the key to Socratic questioning because it makes the intellectual moves used in Socratic dialogue explicit and accessible to anyone interested in learning it, and willing to practice it.” There exists a huge body of literature about Socratic dialogues and the Socratic teaching approach. However, the research gap is how to formalize the process of Socratic teaching. In this paper, we present our attempts to formalize the three steps of the Socratic Method: searching for examples, searching for attributes, and generalizing the attributes. The formal models of the three steps of the Socratic Method has been implemented in a chatbot application and evaluated. This paper reports the evaluation results of these formal models.

## 2 State of the Art of Computer Modeling of Critical Thinking

### 2.1 The Socratic Method

The Socratic Method is rooted back to the philosopher Socrates about whom we know from the books of Platon, one of the Socrates' disciplines. "He [Socrates] is best known for his association with the Socratic method of question and answer, his claim that he was ignorant (or aware of his own absence of knowledge)"<sup>1</sup>. It may be true that Socrates was highly skilled at questioning. However, it is not easy to emulate the types of questions he asked at any given point in a discussion. By studying the Socratic dialogues, Paul and Elder (2007) explicated the components and processes that came to be known as the Socratic Method. They identified a classification of six classes of Socratic questions: (1) questions that require clarification, (2) questions probing assumptions, (3) questions probing reasoning and evidence, (4) questions probing perspective, (5) questions probing implications, (6) questions about the question. This question taxonomy provides us with specific tools for critical questioning.

Nelson (1970) analyzed the questioning technique of Socrates and developed the so-called Socratic Method. A Socratic dialogue, according to Nelson, starts with a self-experienced example. Nelson identified in the habit of Socrates' dialogues that Socrates used observations of daily life as examples and pre-conditions of certain judgment to lead the dialogue partner to the less certain judgment. This is referred to as the abstraction process. Horster (1994) investigated the theoretical assumptions of the Socratic Method, modified the abstraction process proposed by Nelson and extracted the Socratic dialogue in the following steps.

Given a discussion topic (e.g., freedom, happiness, sense of life, ect.), the first step is to ask the dialogue partner to give self-experienced examples for the specified topic. The attributes and features of the topic should be contained in each example (i.e., to be sure that the example is relevant to the topic). The second step is to collect those attributes and features. As the third step, the collected attributes and features will be summarized. These three steps of the Socratic Method has been being applied widely not only in dialogues, but also in a group discussion in the sense of Socratic teaching.

### 2.2 Applications that Support the Socratic Method

Several educational applications support tutorial dialogues. Olney and colleagues (2012) presented a method for generating questions for tutorial dialogue. This involves automatically extracting concept maps from textbooks in the domain of Biology. Five question categories were deployed: hint, prompt, forced choice question, contextual verification question, and causal chain questions. Also with the intention of supporting students using conversational dialogues, Person and Graesser (2002) developed an intelligent tutoring system that improves students' knowledge in the areas of computer literacy and Newtonian physics using an animated agent that is able to ask a series of deep reasoning questions according to the question taxonomy proposed by Graesser and

---

<sup>1</sup> <http://www.iep.utm.edu/socrates/>.

Person (1994). Lane and VanLehn (2005) developed PROPL, a tutor which helps students build a natural-language style pseudo-code solution to a given problem. All these educational applications deployed some kinds of dialogues, however, they did not apply the Socratic Method.

Hoeksema (2004) developed a group discussion environment that is intended to serve virtual Socratic dialogues. The author used the collaborative learning environment Cool Modes (“Collaborative Open Learning Modelling and Designing System”, Pinkwart et al. 2001) The Socratic dialogues using this discussion environment are intended to be held similarly in a usual face-to-face environment. Whereas this work focused on developing an environment for Socratic group discussions, our goal is to formalize the Socratic Method in order to help students develop critical thinking.

Otero and Graesser (2001) developed a computational mechanism for triggering questions. Based on the computational model of a given topic text, the computer should be able to decide for which situation which question can be posed to the student. For this purpose, a formalism for describing conditions of posing questions is required. Otero and Graesser (2001) proposed to use production rules. These rules specify which particular questions should be generated when particular elements or configurations of information in the text occur. Each production rule is an “if state, then action” expression. If a particular state (or Boolean configuration of states) exists, then an action or action sequence is performed (Anderson et al. 1995).

The goal of our work is to formalize the dialogue steps of the Socratic Method. Whereas Otero and Graesser used production rules as formalism, we propose to apply state diagrams, because the dialogue steps could be mapped to the dialogue states intuitively. To our best knowledge, no formalization for the Socratic Method has been developed yet. The formalization of the Socratic Method for dialogues is described in this paper contributes both to the Socratic teaching community (through the integration of critical questions of Paul and Elder in the Socratic Method) and the community of intelligent tutoring systems.

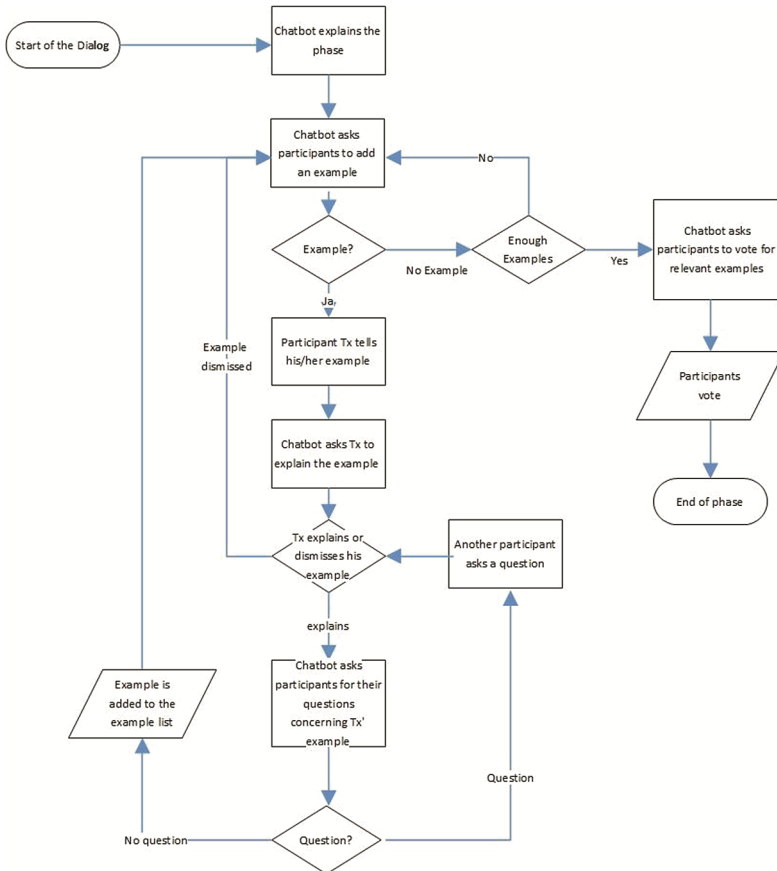
### 3 The Formal Models of the Socratic Method

In this section, we present the formal models for the Socratic Method that was suggested by Nelson (1970) and extended by Horster (1994) and is applied in dialogues or in group discussions. In the developed models, a computer agent is required in order to start conversation by imitating the role of the discussion leader. The computer agent is called “chatbot” in the developed models. He attempts to apply the Socratic Method in order to stimulate the discussion participants to critically think about the discussion topic by searching for examples, searching for attributes, and searching for generalized attributes. The dialogue starts when the chatbot explains the phases of the discussion (search for examples, search for attributes, and summarizing the attributes) and gives a discussion topic.

The first phase starts when the chatbot asks all participants to give examples (cf. Fig. 1). After a participant Tx gives an example, the chatbot asks Tx to explain more about the example by applying the question class “clarification”. At this time, Tx thinks



about the given example again and may dismiss it or elaborate on it. As the next step, the chatbot asks other participants if they have any concern about the example given by Tx. If any questions arise, Tx needs to explain his/her example again. Otherwise, the given example can be added to the collection of examples for the discussion topic and this collection will be shown to other participants. If the example collection is enough (the chatbot may determine the threshold for “enough”, for instance, the threshold is equal the number of participants minus 1), the chatbot asks the participants to vote the best example that will be worked out in the next phase (i.e., searching for attributes). After this vote, the first dialogue phase is finished.



**Fig. 1.** Search for an example

The best voted example is used in the second phase to elicit relevant attributes. First, the chatbot repeats all collected examples (cf. Fig. 2). The chatbot indicates the list of elicited attributes as if the discussion leader uses a whiteboard or flipchart to collect attributes given by the discussion participants (at the beginning, the flipchart is empty). Then, the chatbot asks the participants to elicit attributes from the examples. If a

participant Tx gives feedback (i.e., he/she wants to name an attribute), the chatbot asks Tx to explain the attribute using the question class “clarification” (cf. Paul and Elder 2007). After Tx has elaborated on the named attribute, the chatbot asks other participants about their opinion or whether they have any question concerning the named attribute. If any participant Ty has a question, the chatbot asks him/her about his/her question using the class “questions about the question”. If any participant Ty has an opinion, the chatbot asks about his/her opinion using the class “questions about viewpoints and perspectives”. The question (of the class “questions about the question” or of the class “questions about viewpoints and perspectives”) scrutinizes the question/opinion of Ty and helps him/her to think about his/her question/opinion again. In case, the participant Ty has a question, the chatbot forwards this question to the participant Tx and requests him/her to answer the Ty’s question.

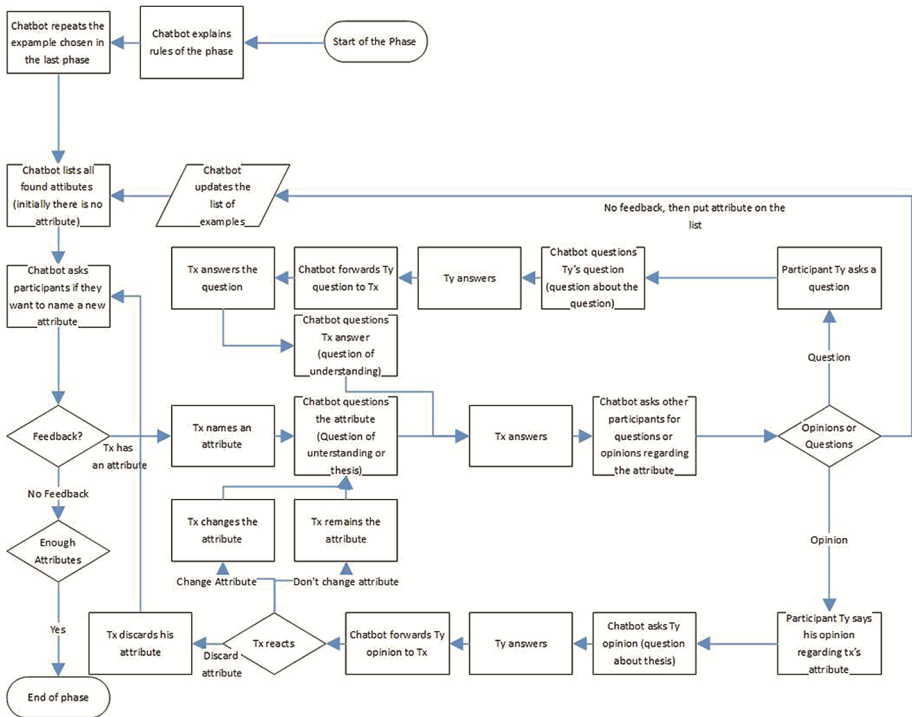


Fig. 2. Search for attributes

This cycle is repeated until no questions from other participants arise. In case, the participant Ty has an opinion, the chatbot forward this opinion to Tx. Tx may dismiss his/her named attribute, change it or remain with that attribute. If neither questions nor opinions arise, the named attribute of the participant Tx will be added to the list of collected attributes. The chatbot decides to finish this discussion phase if the list of collected attributes is “enough” (e.g., the threshold can be determined by the number of participants minus 1). Similar to the threshold for searching

examples, here, the chatbot may determine that the threshold for attributes is equal the number of participants minus 1.

Similar to the previous two phases, at the beginning of the third phase (cf. Fig. 3), the chatbot takes the role of the discussion leader and explains the rules of this phase. Then, the chatbot presents the list of collected attributes for the discussion topic and asks the participants to generalize two or more attributes. If a participant Tx notifies that he/she wants to generalize the attributes, the chatbot asks him/her which attributes on the list he/she wants to generalize and which rule he/she wants to apply.

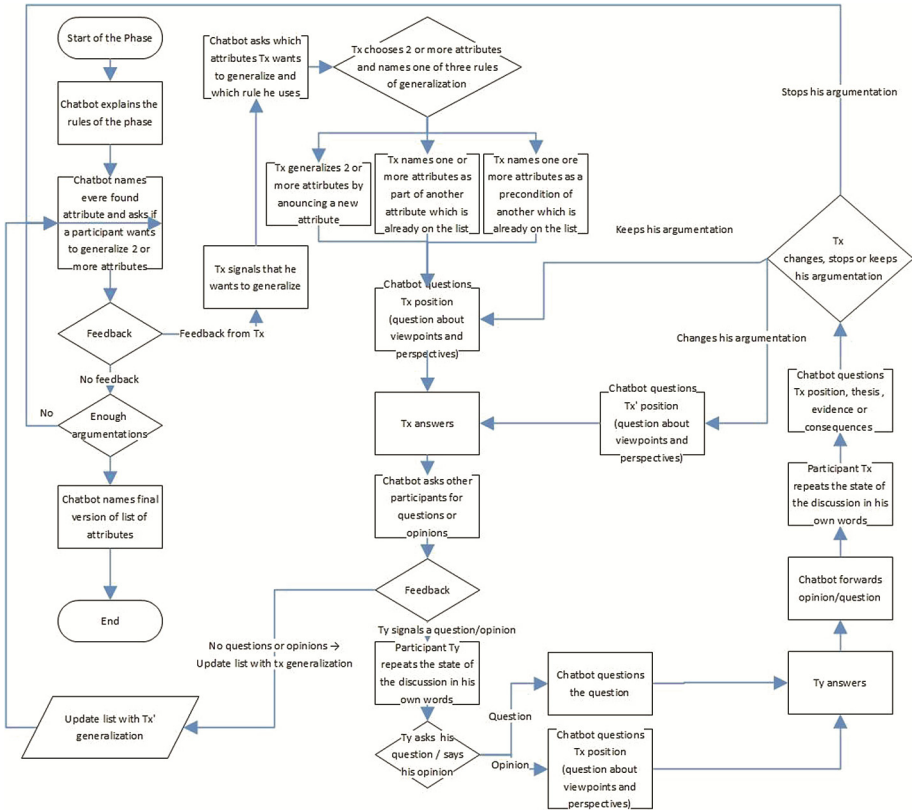


Fig. 3. Generalizing the attributes

Horster (1994) suggested three ways of summarizing the attributes: (1) an attribute is more general than another, (2) a hypernym is used for two or several attributes, and (3) an attribute is a pre-condition for another attribute. The participant Tx will have the option to select two or more attributes from the attribute list and select one of three generalization rules. After Tx has selected a generalization rule, the chatbot will apply the class of “questions that probe reason and evidence” to ask the participant Tx to explain his/her decision. Then, the chatbot asks other participants about their opinions or whether they have any questions. If a participant Ty notifies that he has a question,

first he needs to summarize the status of the discussion in his/her own words, and then asks a question or gives his/her opinion. If the participant Ty asks a question, the chatbot asks about his/her question using the class “questions about the question”. If the participant gives an opinion, the chatbot asks a question about this opinion using the question classes “questions that probe reason and evidence”. After the participant Ty answers the question of chatbot, his/her answer is forwarded to the participant Tx. Then, the chatbot asks the participant Tx to summarize the state of the discussion in his/her own words. Then, the chatbot asks a question about his/her position. Upon on this question, the participant Tx may withdraw his/her argument for generalization of attributes, change or remain the argument (for generalizing the attributes). In case, all other participants agree with the argument for the generalization of attributes (i.e., no question, no opinion), the chatbot updates the list of generalized attributes. If the list of generalized attributes is “enough”, the chatbot can stop the discussion. Horster (1994) proposed an extension of the Socratic Method that the discussion leader goes back to the phase 2 (searching for attributes) and take the next best example to be discussed in the phases 2 and 3.

## 4 Evaluation

In order to verify the applicability of the formal models for the three phases of the Socratic Method, we have decided for an application that is designed to support group discussion. The application provides a chatbot who acts as a discussion leader applying the Socratic Method to stimulate the discussion participants thinking about a discussion topic. In order to show the scalability of the three models of the Socratic Method, in this group discussion scenario, the number of participants will be more than two. The architecture of the application consists of three modules: the user interface is WhatsApp<sup>2</sup>, the message handling module is implemented using Yowsup<sup>3</sup> (a Python program), and the chatbot module which is the main module of the application, implements the developed formal models of the Socratic Method (cf. Sect. 3). This modular architecture allows us to scale to another other platform, e.g., we can substitute the WhatsApp frontend by a web-based user interface if necessary. The detailed description of the architecture of this system is referred to Huse and Le (2016). The goal of our evaluation is to investigate the research objective: Can the Socratic Method be applied in a computerized educational environment by a computer agent effectively? This objective is broken down into the following research questions:

1. Does the chatbot influence the dialogues positively or negatively
2. Is the dialogue structure, which is dictated by the chatbot, helpful or disturbed?
3. Can the discussion participants benefit from the Socratic Method?
4. Does the Socratic aspect of the dialogue have effect on the discussion participants?

---

<sup>2</sup> <http://blog.whatsapp.com/615/WhatsApp-kostenlos-und-n%C3%BCtzlicher-machen> (Access: 07/03/2016).

<sup>3</sup> <https://github.com/tgalal/yowsup/wiki>.

5. Which question class of the critical question taxonomy (Paul and Elder 2007) can support which phase of the Socratic Method better?

In this study, we only investigate the first and second research questions.

#### 4.1 Design and Data

For the evaluation study, we invited six persons in the cycle of our friends who are between 26 and 51 years old. All of them are employed. Only one female participant knows the Socratic Method. We told the study participants in advance that they should plan about two or three hours for participating in the group discussion and what they will need to discuss with other persons in a group discussion using the application. The introduction into the phases of the Socratic Method will be given by the chatbot. Then, the chatbot asks the participants to discuss about the topic “happiness”. Since the participants use the frontend WhatsApp to join the discussion, we did not give any specification about the environment where the participants should join the discussion. If any participant wants to leave the discussion, he/she should acknowledge other participants. If someone who has left the discussion and wants to participate in the on-going discussion, this is allowed. The discussion will be interrupted and closed by the chatbot as long as the last phase is finished. This is important because the rules of the Socratic Method in the last phase are most complex.

The application logs the whole dialogue. We plan to use the log data in order to analyze the quality of questions the chatbot poses to discussion participants, whether participants held the discussion rules according to the Socratic Method, how participants behave to the chatbot during the discussion, and whether the dialogue follows the three phases of the Socratic Method.

In addition to dialogue protocols, after the group discussion is finished, participants are asked to answer a questionnaire. Each question is answered by indicating a value on the Likert scale between 1 (poor) and 10 (good). The questionnaire consists of four sections. The first section consists of general questions about the attitude of participants with respect to the whole Socratic dialogue and the role of the chatbot who adopted the Socratic Method. The second, third and fourth sections of the questionnaire focus on the phases searching examples, searching attributes, and generalizing attributes, respectively.

#### 4.2 Results and Analysis

All participants participated in the group discussion within around three hours. In the first phase of the Socratic dialogue, the participants found five examples for the topic “Happiness”: (1) “Happiness means for me to be near to the beloved ones.”, (2) “The birth of my daughter brought me happiness.”, (3) “I felt my happiness at my wedding.”, (4) “I had happiness when I got my Berliner cup.”, (5) “I felt my happiness, as we got our house.”

One example “I felt my happiness when I met my partner.” was posed, but then removed by one participant. Among the five examples, the one “The birth of my daughter

brought me happiness.” was voted by the participants as the best one to be worked out in the second phase. On average, about 1.5 questions for each example were posed by the participants.

In the second phase, five attributes were given by the participants: (1) “When kids are born, this is the greatest happiness.”, (2) “At the birth of our kids, we forget our suffer of daily lives.”, (3) “Love”, (4) “Satisfaction”, (5) “Protective instinct”.

In addition to the five attributes, the participants posed five questions and one opinion. These attributes are generalized by the participants in the following ones: “At the birth of our kids, we forget our suffer of daily lives.”, “Satisfaction”, “Love”.

From the number of the examples, attributes and generalized attributes given by the participants, at the first sight, we can conclude that the participants participated in the group discussion led by the chatbot actively and that they followed the dialogue structure according to the Socratic Method.

Table 1 shows the results of the general questions of the questionnaire. Due to the page limit, we do not investigate the evaluation results for each dialogue phase.

**Table 1.** Results of general questions about the socratic chatbot

Question	Mean (s.d.)
Which impression do you have about the socratic dialogue that you have participated in?	7.8 (0.7)
Do you think that the questions of the chatbot helped you to rethink about your contribution (example, attribute, question, opinion) more critically?	6.5 (2.1)
How did you find the structure of the dialogue that was led by the chatbot?	8.2 (1.2)
How do you rate the moderation of the chatbot with respect to the discussion flow?	6.5 (1.5)
How do you rate the explanation and the instruction given by the chatbot for each dialogue phase?	7.8 (1.1)

In general, the participants found the dialogue positively ( $m = 7.8$ ). With respect to the impact of the Socratic questions on the participants, not all participants did agree ( $m = 6.5$ ). Two of the participants found the Socratic questions that were posed by the chatbot not helpful. On the contrary, two other participants found the Socratic questions helpful. The dialogue structure that was led by the chatbot did not disturb the discussion ( $m = 8.2$ ). However, the discussion moderation of the chatbot was rated by the participants with a value above average ( $m = 6.5$ ). The explanation and the instruction of the chatbot each dialogue phase were useful for the participants ( $m = 7.8$ ).

Although the mean values of these questions (Table 1) are above average (5), with respect to the first and second research questions, we can say that the chatbot influences the group discussion positively and the dialogue structure given by the chatbot did not disturb the discussion. Since the number of participants in this study is low and the participants came from the cycle of our friends, this evaluation study is biased and limited, and thus, we cannot conclude about the statistical significance.

## 5 Conclusions and Future Work

In this paper, we have presented three models for three dialogue phases of the Socratic Method by applying state diagrams. The models are intended to help students develop critical thinking for a given discussion topic. These models have been validated and evaluated in a group discussion scenario using an application. The evaluation showed that the models can be integrated in an educational system for training Socratic dialogues. The evaluation also demonstrated the tendency that a chatbot who is implemented with these models can be useful. Since no formal models for the Socratic Method has been proposed until now, to our best knowledge, the introduced models for three dialogue phases are the contribution of this paper. In addition, the integration of critical questions using the question taxonomy developed by Paul and Elder (2007) in the steps of the Socratic Method can be considered a second contribution in the paper. In the future, we will analyze data collected from the Sects. 2, 3 and 4 of the questionnaire and from the logged dialogues.

## Referneces

- Anderson, J.R., Corbett, A.T., Koedinger, K.R., Pelletier, R.: Cognitive tutors: lessons learned. *J. Learn. Sci.* **4**, 167–207 (1995)
- Graesser, A.C., Person, N.K.: Question asking during tutoring. *Am. Educ. Res. J.* **31**(1), 104–137 (1994)
- Hoeksema, K.: Virtuelle sokratische gespräche - umsetzung einer idee aus dem philosophieunterricht. In: Proceedings on Modellierung als Schlüsselkonzept in Intelligenten Lehr-/Lernsystemen (2004)
- Horster, D.: Das Sokratische Gespräch in Theorie und Praxis. Springer, Opladen (1994)
- Huse, N., Le, N.T.: The formal models for the socratic method. In: Proceedings of the 4th International Conference on Computer Science Applied Mathematics and Applications. Springer, Heidelberg (2016)
- Lane, H.C., Vanlehn, K.: Teaching the tacit knowledge of programming to novices with natural language tutoring. *J. Comput. Sci. Educ.* **15**, 183–201 (2005)
- Nelson, L.: Die sokratische methode. In: ders., Gesammelte Schriften in neun Bänden, Band 1, Hamburg (1970)
- Olney, A.M., Graesser, A., Person, N.K.: Question generation from concept maps. *Dialogue Discourse* **3**(2), 75–99 (2012)
- Otero, J., Graesser, A.C.: PREG: elements of a model of question asking. *Cogn. Instr.* **19**(2), 143–175 (2001). Lawrence Erlbaum Associates
- Paul, R., Elder, L.: Critical thinking: the art of socratic questioning, part I. *J. Dev. Educ.* **31**(1), 36–37 (2007). ProQuest Education Journal
- Person, N.K., Graesser, A.C.: Human or computer? Autotutor in a bystander turing test. In: Cerri, S.A., Gouardères, G., Paraguaçu, F. (eds.) ITS 2002. LNCS, vol. 2363, pp. 821–830. Springer, Heidelberg (2002)
- Pinkwart, N., Hoppe, U., Gaßner, K.: Integration of domain-specific elements into visual language based collaborative environments. In: Proceedings of 7th International Workshop on Groupware, pp. 142–147. IEEE Computer Society (2001)

# Stealth Assessment in ITS - A Study for Developmental Dyscalculia

Severin Klingler<sup>1</sup>(✉), Tanja Käser<sup>1</sup>, Alberto-Giovanni Busetto<sup>2</sup>,  
Barbara Solenthaler<sup>1</sup>, Juliane Kohn<sup>3</sup>, Michael von Aster<sup>3,4,5</sup>,  
and Markus Gross<sup>1</sup>

<sup>1</sup> Department of Computer Science, ETH Zurich, Zürich, Switzerland  
kseverin@inf.ethz.ch

<sup>2</sup> Department of Electrical and Computer Engineering,  
University of California, Santa Barbara, USA

<sup>3</sup> Department of Psychology, University of Potsdam, Potsdam, Germany

<sup>4</sup> Center for MR-Research, University Children's Hospital Zurich, Zürich, Switzerland

<sup>5</sup> Department of Child Adolescent Psychiatry,  
DRK Kliniken Berlin Westend, Berlin, Germany

**Abstract.** Intelligent tutoring systems are adapting the curriculum to the needs of the student. The integration of stealth assessments of student traits into tutoring systems, i.e. the automatic detection of student characteristics has the potential to refine this adaptation. We present a pipeline for integrating automatic assessment seamlessly into a tutoring system and apply the method to the case of developmental dyscalculia (DD). The proposed classifier is based on user inputs only, allowing non-intrusive and unsupervised, universal screening of children. We demonstrate that interaction logs provide enough information to identify children at risk of DD with high accuracy and validity and reliability comparable to traditional assessments. Our model is able to adapt the duration of the screening test to the individual child and can classify a child at risk of DD with an accuracy of 91 % after 11 min on average.

**Keywords:** Automatic assessment · Feature processing · Bayesian network · Pairwise clustering · Computer-based screening · Dyscalculia

Intelligent tutoring systems (ITS) are gaining importance in education. A lot of research has been conducted to represent and model student knowledge accurately, design effective curricula and develop optimal instructional policies. A large body of work has focused on mining the data logs collected from ITS. Important topics in this area are automatic stealth assessments such as the evaluation of student learning or detection of student properties (e.g. intelligence, learning disabilities) [31]. Traditional assessments are often time consuming and have to be supervised by an expert, rendering them expensive in practice. Hence, this approach does not scale and is therefore not suitable in many cases, such as MOOCs, large university courses, or widespread screenings in elementary schools to enable early detection of learning disabilities.



Previous work has investigated stand-alone automatic digital assessments, including research on automatic scoring [5], item generation [18] and game-based assessment [20]. Furthermore, digital screening programs replacing traditional neuropsychological tests, for example for dyscalculia [10] or dyslexia [12], have been developed. Ideally, such computer-based screening programs are seamlessly integrated into an ITS. This enables not only automatic and non-intrusive assessment of students, but also analysis and detection of student traits that allow for a better adaptation of the curriculum to the individual needs. Despite these advantages only few work have addressed such ITS systems with fully integrated assessment. One step in this direction are integrated behavior detectors identifying students gaming the system [6], finding wheel-spinning students [9] or modeling engagement, e.g. [1, 8, 14]. Other work used clustering and classification approaches to detect students' mathematical characteristics [23].

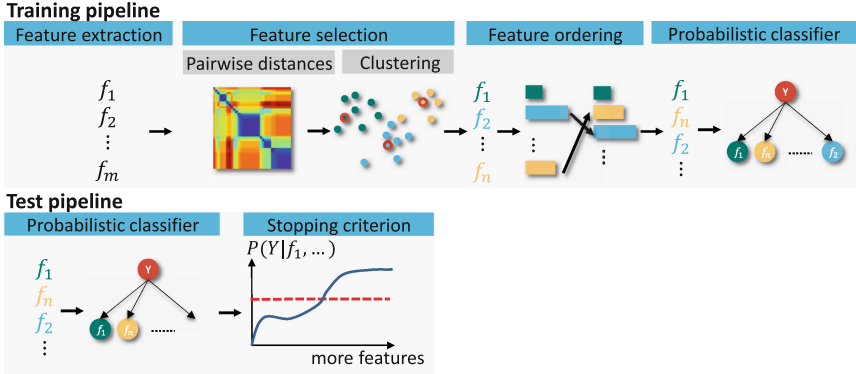
In this paper, we propose a pipeline for integrating automatic assessment, i.e. detectors of student traits, directly into the tutoring system. We validate our approach for the case of developmental dyscalculia (DD) (a specific learning disability affecting the acquisition of arithmetic skills [2]) and the game-based training environment *Calcularis* [22].

Our pipeline leverages the potential of machine learning algorithms. Its data-driven nature features several advantages. First, since it builds upon a large set of student training data, the costs for model building are low and the accuracy of the classifier can be continuously improved as more student data is added over time. The test duration can be adapted to each child individually, which reduces the average test duration substantially. Second, our classifier can be seamlessly embedded into an ITS (in our case *Calcularis* [22]), where the assessment runs continuously and non-intrusively in the background. This integration reduces testing expenses and emotional stress imposed to children is kept at a minimum. The embedding allows the ITS to leverage the information from the stealth assessment during the training. Third, our pipeline has the potential to be applied to a different ITS and be used for the assessment of different student traits.

We extensively evaluate the accuracy, practicability, and validity of our approach on data logs from 68 children. Our results demonstrate that we can identify children at risk of DD with a high accuracy (91 % sensitivity, 91 % specificity) within a short time (11 min on average). We conclude from our results that recorded user inputs alone could potentially allow for a detailed reconstruction of student traits and that the integration of stealth assessments may refine the adaptation of the curriculum that ITS are currently providing.

## 1 Adaptive Classification Algorithm

Our adaptive classification is based on the training environment *Calcularis* [22], a computer-based system for learning mathematics designed for children with DD. The program is structured into different instructional games, which are



**Fig. 1.** Processing pipeline: pairwise distances of features  $f$  serve as input for the clustering. We select the representative feature per cluster and determine an optimal feature ordering. A Naive Bayes model is trained on the selected features. The probabilistic output of the classifier is used to adapt the test duration to each child.

designed based on current neuro-cognitive theory. *Calcularis* consists of ten different games representing 100 different skills that are essential for learning mathematics. Our model building process consists of four steps (see Fig. 1). We first extract a large set of candidate features and then perform feature selection based on common similarity measures. Next, we build our adaptive classifier by first sorting the selected features and then defining a Naive Bayes model.

**Feature Extraction.** We identified a set of recorded features that describe different mathematical properties of the user. These features can be classified into *skill-* and *game dependent* features, and are summarized in Table 1. *Skill dependent* features provide information about tasks associated with a specific skill. The performance  $\mathbf{P}$  for a skill measures the ratio of correctly solved tasks for a given number of tasks. We expect children without DD to outperform children with DD on these tasks, since mathematical abilities of children with DD are at a level comparable to the level of children without DD of lower age [3]. Answer time  $\mathbf{AT}$  is measured for all skills as children with DD tend to have longer answer times compared to children without DD [17]. They often show deficits in fact retrieval and tend to have difficulties to acquire arithmetic procedures [28] which increases answer times for simple arithmetic tasks. We count typical mistakes  $\mathbf{TM}$  for a subset of games where such a measure is meaningful.  $\mathbf{TM}$  are extracted by matching the erroneous result to a set of error patterns. As an example switching the digits of the result in an arithmetic task is considered a typical mistake (e.g.  $15 + 9 = 42$ ). The complete set of error patterns is described in [22]. Additional *game dependent* features were chosen related to specific games. The estimation game feature  $\mathbf{E}$  measures the relative number of overestimates when estimating the number of points in a point cloud. Whether children with DD are less sensitive to differences in this number representation is not consistently supported by recent work [27]. The feature  $\mathbf{SN}$  for the

**Table 1.** Extracted features and abbreviations (**bold**) used in the screener.

Feature	Description
<i>Skill dependent features (extracted at specific skills)</i>	
<b>Performance</b>	Ratio of correctly solved tasks.
<b>Answer Time</b>	Average answer time.
<b>Typical Mistakes</b>	Number of typical mistakes committed.
<i>Game dependent features</i>	
<b>Estimation</b>	Estimating the number of displayed points.
	<b>E</b> is the ratio between number of overestimates and task count.
<b>Secret Number</b>	Guessing a number in as few steps as possible.
	<b>S</b> is the ratio by which the remaining search interval is reduced.
<b>Ordering</b>	Is a number sequence ordered ascending?
	<b>O</b> is the ratio of false positive and incorrectly solved tasks.
<b>Landing</b>	Positioning a number on a number line.
	<b>L</b> is the distance to the correct position of the given number).

secret number game measures the reduction of the search interval while repeatedly guessing the same number. This feature quantifies common problem-solving strategies such as bisection of the search interval or linear search. The ordering game feature **O** measures the ratio of false positives when assessing whether numbers are in ascending order. Children with DD are shown to be less efficient when processing numbers [26], therefore we hypothesize that they will perform worse when comparing numbers. The landing game feature **L** measures the error of the number estimate. Deficits in spatial number representation as often shown by children with DD [25] are obstructive to this task, thus we expect children with DD to perform significantly worse compared to peers without DD.

**Feature Selection.** Our feature extraction yields a few hundred features, each corresponding to a set of tasks the user has to solve. Therefore, the number of features directly influences the test duration. To limit the test duration and to remove possible correlations between features, we only use a subset of features for classification. We cluster the features into groups based on their similarity and select one representative feature per cluster. As the different feature types have different domains (e.g., **P**  $\in [0, 1]$ , **AT** seconds  $> 0$ ) a direct comparison between the features is not meaningful. We therefore process the features to make them comparable.

In a first step, we compute a similarity matrix  $\mathbf{K}_i \in [0, 1]^{S \times S}$  for each feature  $f_i$ , where  $S$  denotes the number of children. Therefore,  $\mathbf{K}_i$  contains the pairwise similarities between each pair of children regarding feature  $f_i$ . We design the matrices based on the nature of each feature and in particular exploiting invariance of the feature types. For example, for the answer time **AT** we combine a Gaussian kernel with a log transform to obtain

$$\mathbf{K}_i(s, u) = \exp\left(-\frac{\|\log(f_i^s) - \log(f_i^u)\|^2}{2\sigma^2}\right), \quad (1)$$

where  $f_i^s$  and  $f_i^u$  denote the respective feature values for children  $s$  and  $u$ . We incorporate a cumulative beta distribution to design the similarity matrix for the performance features  $\mathbf{P}$ . For the  $\mathbf{SN}$  feature, we designed an exponential kernel. All other features ( $\mathbf{TM}, \mathbf{E}, \mathbf{O}, \mathbf{L}$ ) apply a standard Gaussian kernel. Further details regarding the design of the different kernels can be found in [24].

In a second step, we cluster the features using pairwise-clustering [21] based on the pairwise distances  $d_{ij} = \|\mathbf{K}_i - \mathbf{K}_j\|_F$  between all feature pairs using the Frobenius norm. We then compute an optimal matrix  $\mathbf{T}$ , which contains the pairwise Hamming distances between child labels, i.e.,  $\mathbf{T}(s, u) = 0$  if  $s$  and  $u$  belong to the same group  $\in \{DD, CC\}$ , with  $CC$  referring to control, and  $\mathbf{T}(s, u) = 1$  otherwise. For each cluster, we select one representative feature, which is the one with the smallest distance  $dt_i = \|\mathbf{K}_i - \mathbf{T}\|_F$  to matrix  $\mathbf{T}$ .

**Probabilistic Classifier.** Based on the selected features, we develop a probabilistic model that adapts the test duration to the individual child. The classification task is solved using an adapted Naive Bayes model, which assumes conditional independence of all the features  $f_i$  given the group label  $Y$  ( $Y = 0$  child with DD,  $Y = 1$  CC), but was shown to perform optimally even if the independence assumption is violated [34]. Correlations between features are low in our case (average  $\rho=0.07$ ,  $<1\%$  significant correlations at  $\alpha=0.001$ ) because of our feature selection step. The posterior probability of the group label  $Y$  for a child given  $N$  observed features is proportional to

$$p(Y|f_1, \dots, f_N) \propto \prod_{i=1}^N p(f_i|Y) \cdot p(Y), \quad (2)$$

where for every feature we choose the density  $p(f_i|Y)$  from a set of standard distributions that best models the data according to the BIC score. We assume a normal distribution for the features  $\mathbf{E}$ ,  $\mathbf{SN}$ ,  $\mathbf{O}$  and  $\mathbf{L}$ , and a Beta, Gamma, and Poisson distribution for  $\mathbf{P}$ ,  $\mathbf{AT}$ , and  $\mathbf{TM}$ , respectively. The prior probability  $p(Y)$  is set to the estimated prevalence of DD [30]. Due to the independence assumption, we can deal with cases where we only observe a subset of all features. After observing the first feature  $f_1$ , we can compute  $p(Y = 1|f_1)$ . Having observed  $f_2$ , we infer  $p(Y = 1|f_1, f_2)$  etc. For any threshold  $\tau \in [0, 1]$ , the predicted group label  $\hat{Y}$  can then be computed as

$$\hat{Y} = \begin{cases} 1 & p(Y = 1|f_1, \dots, f_n) > \tau \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

**Feature Ordering.** To determine the optimal ordering of the tasks in the test, we compute the amount of group information contained in each feature. We prefer features where the feature values differ substantially across the groups (DD and CC) and are similar within the group. To assess the quality of each feature

$f_i$ , we use an unpaired t-test for a difference in means of the two independent groups. We then order the features by sorting the calculated p-values in ascending order, *i.e.*, the feature with the smallest p-value is asked first.

**Stopping Criterion.** The optimal point in time to stop the test is heuristically determined. After observing the first  $t$  features the classifier has a current belief about the group label of a child and predicts the label based on  $p(Y|f_1, \dots, f_t) > \tau$  (see Eq. (2)). Intuitively, we stop the test if observing the next feature would not contradict our current belief about the group label. As the next feature value  $f_{t+1}$  is unknown, the feature value in the training data  $\hat{f}_{t+1}$  that contradicts the model’s current belief the most is taken instead. We stop if observing  $\hat{f}_{t+1}$  is not changing the current belief, *i.e.*, if  $p(Y = 1|f_1, \dots, f_t) > \tau$  and  $p(Y = 1|f_1, \dots, f_t, \hat{f}_{t+1}) > \frac{\tau}{2}$ .

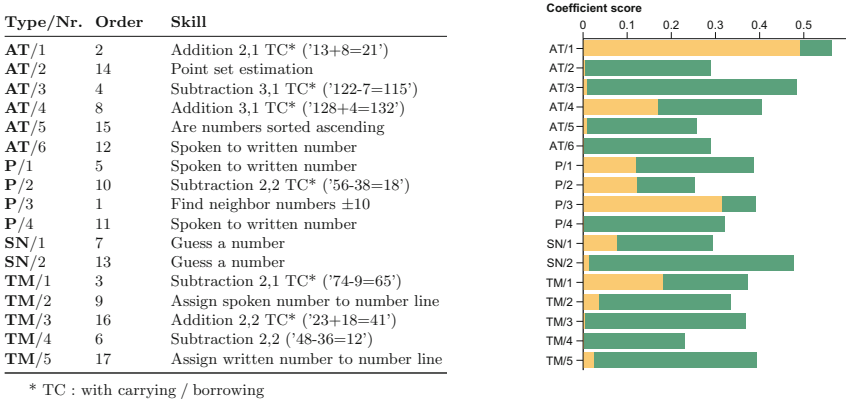
## 2 Experimental Evaluation

The experimental evaluation of our method was based on log files from 68 participants (32 DD, 36 CC) of a multi-center user study conducted in Germany and Switzerland [32]. During the study, children trained with *Calcularis* at home for five times per week during six weeks and solved on average 1551 tasks. There were 28 participants in the 2<sup>nd</sup> grade (9 DD, 19 CC) and 40 children in the 3<sup>rd</sup> grade (23 DD, 17 CC). The diagnosis of DD was based on standardized neuropsychological tests [4, 16, 19].

We calculated the accuracy, the specificity and the sensitivity of our model based on the predicted and the true label of the students (either DD or CC). All results were computed on unseen students in the test set. Training and test sets were created using .632 bootstrap with resampling ( $B = 300$ ). All parameter estimates are based on maximum likelihood estimation using Nelder-Mead simplex direct search. The optimization stops when the improvement in the likelihood is  $< 10^{-4}$  or after 400 iterations. Hyper parameters (parameters for kernels and features) and features (including feature ordering) were selected using nested cross validation, employing .632 bootstrap with resampling ( $B = 300$ ) on top of 10-fold cross validation. The optimal number  $k^*$  of clusters in the feature selection step was heuristically determined by limiting the maximal test duration to  $< 35$  min. Since we required five recorded tasks per feature (average recorded task time: 0.39 min), this test duration results in  $k^* = 17$  clusters (which leads to 85 tasks in the test).

**Content Validity.** 17 features were automatically selected based on the recorded data alone. For all features we calculated Pearson’s correlation coefficient  $\rho^2$  and the maximal information coefficient (MIC) [29] between the feature and the test score to measure the linear and non-linear relationships, respectively. For most features the relationship is highly non-linear, which prohibits the use of simple prediction methods such as linear regression. The feature ordering yields the optimal task sequence in the test as listed in Fig. 2.

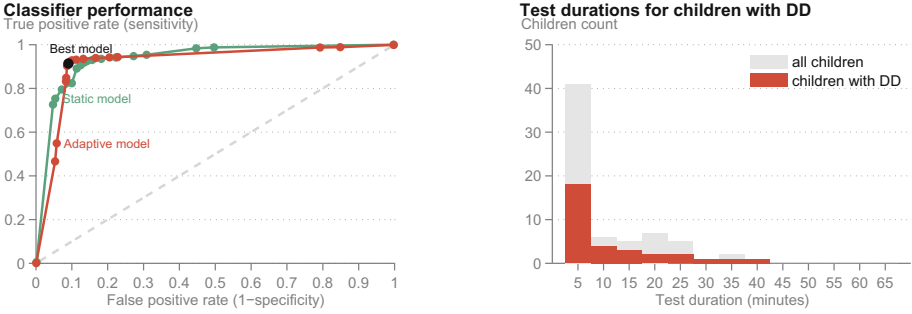
The automatically selected features agree well with findings in previous work on DD. Deficits in number comparison that are shown by children with DD [26]



**Fig. 2.** Selected features and their corresponding skills and ordering in the test. The relationship between a feature and the test score is shown on the right, using Pearson's correlation coefficient (yellow) and the maximal information coefficient MIC (green). (Color figure online)

are captured by considering temporal and performance values (AT/5, P/3). Children with DD exhibit deficits in number processing [13]. Number processing skills are captured in various features and include again temporal and performance information (AT/2, AT/6, P/1, P/4). The features extracted from the number line game (TM/2, TM/5) capture typical mistakes in spatial number representation [11]. Furthermore, different problem solving strategies are analyzed based on the Secret Number game (SN/1, SN/2). Finally, difficulties acquiring simple arithmetic procedures and deficits in fact retrieval that are frequently shown by children with DD [28] are captured measuring answer times for various arithmetic procedures in AT/1, AT/3. Interestingly, no features from tasks associated with subitizing are selected, although subitizing is considered one of the basic functions often impaired for children with DD [26]. Most of the selected features correspond well with the type of tasks used in standardized tests for DD such as counting, number comparison, number representation and simple arithmetical tasks [4]. Note that the screener includes some features such as typical mistakes and problem solving strategies that are not captured by paper tests. The type of the selected features agrees well with other screening tools that measure answer time, performance and typical mistakes on tasks such as dot enumeration, number comparison, single digit arithmetic (Dyscalculia Screener Digital [10]) or recognizing reading and writing of natural number (DyscaliUM [7]).

**Criterion-Related Validity.** In Fig. 3, left, we compare the performance of the static and adaptive Bayesian network model with ROC curves. In the static case (green line), we used all features, *i.e.*, all tasks, while in the adaptive case (red line) we used early test abortion based on our stopping criterion. Every point on the curves corresponds to a different threshold  $\tau$  for the probabilistic classifier.



**Fig. 3. Left.** Performance comparison of the classifiers using ROC curves. The adaptive approach with reduced test duration (red) shows comparable performance to the classifier using all features (green). Points on the curves correspond to different probability thresholds  $\tau$  at which the model decides if a child has DD. **Right.** Test durations for all children (grey) and DD (red). Our adaptive screener requires on average 11 test minutes to classify a child. Around 40 % can be classified already after 5 min. (Color figure online)

Our best classifier (selected by cross validation) exhibits a high sensitivity and specificity of 0.91 for a threshold  $\tau = 0.3$  (black dot).

There is no significant decrease in performance when we stop the test early with our adaptive model, i.e., on average, children are not misclassified more frequently. In fact, the adaptive classifier that is based on partial data is outperforming the static approach for a specificity in the range  $[0.05, 0.15]$ . As the features are ordered based on how much information they carry about the group label, it can be advantageous to neglect those with little information since they tend to have more noisy information. Our classifier achieves a higher sensitivity compared to the stand-alone digital screening test DyscalculiUM; no comparison can be done with the Dyscalculia Screener Digital as it was standardized independent of traditional tests for DD.

**Construct Validity.** Construct validity of our method was assessed by correlating the probabilistic output of our screener with a series of tests measuring different cognitive aspects of all participants. We performed standardized tests to assess convergent validity and discriminant validity as listed on the right. We observe moderate to high correlation coefficients for all measures capturing related cognitive concepts and weak correlations to the set of tests measuring unrelated concepts. These results are comparable to construct validity analysis of standardized neuropsychological tests that assess mathematical abilities. Correlations for these tests range from 0.22 to 0.73 [15, 33].

Test	$\rho$	$p$ -value
<i>Convergent validity</i>		
Non verbal intelligence [16]	0.44	$<10^{-3}$
Math anxiety test [26]	0.42	$<10^{-2}$
Cognitive competence [1]	0.63	$<10^{-7}$
<i>Discriminant validity</i>		
Working memory [19]	0.19	0.13
Verbal intelligence [16]	0.23	0.06
Sport competence [1]	-0.17	0.18
Peer acceptance [1]	0.08	0.51
Attentional performance [43]	0.25	0.10

**Reliability.** Classical notion of test reliability in terms of measures such as Cronbach’s alpha do not apply for our adaptive test due to non tau-equivalence of the measurements and the fact that our test output is a non-linear function of item scores. We therefore investigate the split-half reliability of our proposed model as an approximation to the standard notion of test reliability. We observe a reliability of 0.87. This is comparable to other mathematical tests where a reliability in the range of 0.7 to 0.92 is reported [15,16].

**Test Duration.** Due to our stopping criterion, the test duration is adapted to the individual child. Figure 3, right, shows the test duration for all children (grey) and for DD (red). On average, our adaptive screener classifies a child as DD or CC after only 11 min (at which point the test is stopped). This is notably shorter than screener durations reported in previous work. In comparison, the test duration of the Dyscalculia Screener Digital is reported to be between 15 and 30 min [10]. For Higher Education, a test duration of 48 min was reported using the computer-based screener for DD DyscalculiUM. With our adaptive screener, roughly 40% of children are already classified after five test minutes. Our static screener test takes 26.6 min on average, which emphasizes the importance of the adaptivity. The adaptive stopping criterion is important to retain classification accuracy as for 43% of the children the initial classification changed until the stopping criterion was met.

### 3 Discussion and Conclusion

We developed a fully data-driven pipeline for the automatic detection of student traits that can be seamlessly embedded into an ITS. We validated the method for the case of DD, allowing for non-intrusive and unsupervised screening of children while they are training with the ITS. The automatically selected features are covering a broad range of different characteristics of the children and are in accordance with the literature on DD. The classifier exhibits high sensitivity (0.91) and specificity (0.91) and adapts the test duration to each child individually, resulting in an average duration of as little as 11 min. Further, our method exhibits good construct validity (high correlations to tests measuring mathematical abilities, low correlations to tests assessing dissimilar abilities). These findings demonstrate that student traits can be effectively learned from user inputs alone. This knowledge about student traits allows an ITS to further adapt the curriculum to the specific needs of the students. In the future we would like to investigate potential intervention strategies based on the inferred knowledge about student traits. While this work evaluates the proposed model only for the screening of children at risk of DD, there is nothing inherently DD specific in the method. As such, our framework can be applied for the unobtrusive detection of other student traits and using different learning environments.

**Acknowledgments.** This work was supported by ETH Grant ETH-23 13-2.



## References

1. Arroyo, I., Woolf, B.P.: Inferring learning and attitudes from a bayesian network of log file data. In: Proceedings of AIED, pp. 33–40 (2005)
2. von Aster, M.G., Shalev, R.: Number development and developmental dyscalculia. *Dev. Med. Child Neurol.* **49**, 868–873 (2007)
3. von Aster, M.G.: Developmental cognitive neuropsychology of number processing and calculation: varieties of developmental dyscalculia. *Eur. Child Adolesc. Psychiatry* **9**, S41–S57 (2000)
4. von Aster, M., Zulauf, M.W., Horn, R.: Neuropsychologische Testbatterie für Zahlenverarbeitung und Rechnen bei Kindern: ZAREKI-R. Pearson, Frankfurt (2006)
5. Attali, Y.: Reliability-based feature weighting for automated essay scoring. *Appl. Psychol. Meas.* **39**(4), 303–313 (2015)
6. Baker, R.S., Corbett, A.T., Koedinger, K.R.: Detecting student misuse of intelligent tutoring systems. In: Lester, J.C., Vicari, R.M., Paraguaçu, F. (eds.) ITS 2004. LNCS, vol. 3220, pp. 531–540. Springer, Heidelberg (2004)
7. Beacham, N., Trott, C.: Screening for dyscalculia within HE. *MSOR* **5**, 1–4 (2005)
8. Beck, J.E.: Engagement tracing: using response times to model student disengagement. In: Proceedings of AIED, pp. 88–95 (2005)
9. Beck, J.E., Gong, Y.: Wheel-spinning: students who fail to master a skill. In: Lane, H.C., Yacef, K., Mostow, J., Pavlik, P. (eds.) AIED 2013. LNCS, vol. 7926, pp. 431–440. Springer, Heidelberg (2013)
10. Butterworth, B.: *Dyscalculia Screener*. Nelson Publishing Company Ltd., London (2003)
11. Butterworth, B., Varma, S., Laurillard, D.: Dyscalculia: from brain to education. *Science* **332**(6033), 1049–1053 (2011)
12. Cisero, C., Royer, J., Marchant, H., Jackson, S.: Can the computer-based academic assessment system (CAAS) be used to diagnose reading disability in college students? *J. Educ. Psychol.* **89**(4), 599–620 (1997)
13. Kadosh, R.C., Kadosh, K.C., Schuhmann, T., Kaas, A., Goebel, R., Henik, A., Sack, A.T.: Virtual dyscalculia induced by parietal-lobe TMs impairs automatic magnitude processing. *Current Biol.* **17**, 689–693 (2007)
14. Cooper, D.G., Muldner, K., Arroyo, I., Woolf, B.P., Burleson, W.: Ranking feature sets for emotion models used in classroom based intelligent tutoring systems. In: De Bra, P., Kobsa, A., Chin, D. (eds.) UMAP 2010. LNCS, vol. 6075, pp. 135–146. Springer, Heidelberg (2010)
15. Desoete, A., Grégoire, J.: Numerical competence in young children and in children with mathematics learning disabilities. *Learn. Individ. Differ.* **16**(4), 351–367 (2006)
16. Esser, G., Wyschkon, A., Ballaschk, K.: *BUEGA: Basisdiagnostik Umschriebener Entwicklungsstörungen im Grundschulalter*. Hogrefe, Göttingen (2008)
17. Geary, D.C., Brown, S.C., Samaranayake, V.A.: Cognitive addition: a short longitudinal study of strategy choice and speed-of-processing differences in normal and mathematically disabled children. *Dev. Psychol.* **27**(5), 787–797 (1991)
18. Graf, E.A., Fife, J.H.: Difficulty modeling and automatic generation of quantitative items: recent advances and possible next steps. In: Gierl, M.J., Haladyna, T.M. (eds.) *Automatic Item Generation: Theory and Practice*, pp. 157–179. Routledge, London (2013)

19. Haffner, J., Baro, K., Parzer, P., Resch, F.: Heidelberger Rechentest (HRT): Erfassung mathematischer Basiskompetenzen im Grundschulalter. Hogrefe Verlag, Goettingen (2005)
20. Hao, J., Shu, Z., Davier, A.: Analyzing process data from game/scenario- based tasks: an edit distance approach. *JEDM* **7**, 33–50 (2015)
21. Hofmann, T., Buhmann, J.M.: Pairwise data clustering by deterministic annealing. *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(1), 1–14 (1997)
22. Käser, T., Baschera, G.M., Kohn, J., Kucian, K., Richtmann, V., Grond, U., Gross, M., von Aster, M.: Design and evaluation of the computer-based training program *calcularis* for enhancing numerical cognition. *Front. Dev. Psychol.* **4**, 489 (2013)
23. Käser, T., Busetto, A.G., Solenthaler, B., Kohn, J., von Aster, M., Gross, M.: Cluster-based prediction of mathematical learning patterns. In: Lane, H.C., Yacef, K., Mostow, J., Pavlik, P. (eds.) *AIED 2013*. LNCS, vol. 7926, pp. 389–399. Springer, Heidelberg (2013)
24. Käser, T.: Modeling and Optimizing Computer-Assisted Mathematics Learning in Children. Ph.D. thesis, Diss., ETH Zürich, Nr. 22145 (2014)
25. Kucian, K., Grond, U., Rotzer, S., Henzi, B., Schönmann, C., Plangger, F., Gälli, M., Martin, E., von Aster, M.: Mental number line training in children with developmental dyscalculia. *NeuroImage* **57**(3), 782–795 (2011)
26. Landerl, K., Bevan, A., Butterworth, B.: Developmental dyscalculia and basic numerical capacities: a study of 8-9-year-old students. *Cognition* **93**, 99–125 (2004)
27. Noël, M.P., Rousselle, L.: Developmental changes in the profiles of dyscalculia: an explanation based on a double exact-and-approximate number representation model. *Front. Hum. Neurosci.* **5**, 165 (2011)
28. Ostad, S.A.: Developmental differences in addition strategies: a comparison of mathematically disabled and mathematically normal children. *Br. J. Educ. Psychol.* **67**, 345–357 (1997)
29. Reshef, D.N., Reshef, Y.A., Finucane, H.K., Grossman, S.R., McVean, G., Turnbaugh, P.J., Lander, E.S., Mitzenmacher, M., Sabeti, P.C.: Detecting novel associations in large data sets. *Science* **334**(6062), 1518–1524 (2011)
30. Shalev, R., von Aster, M.G.: Identification, classification, and prevalence of developmental dyscalculia. *Encyclopedia of Language and Literacy, Development*, pp. 1–9 (2008)
31. Shute, V.J.: Stealth assessment in computer-based games to support learning. In: *Computer Games and Instruction* (2011)
32. Von Aster, M., Rauscher, L., Kucian, K., Käser, T., McCaskey, U., Kohn, J.: *Calcularis* - evaluation of a computer-based learning program for enhancing numerical cognition for children with developmental dyscalculia. In: 62nd Annual Meeting of the American Academy of Child and Adolescent Psychiatry (2015)
33. Woolger, C.: Wechsler intelligence scale for children-third edition (WISC-III). In: Dorfman, W.I., Hersen, M. (eds.) *Understanding Psychological Assessment. Perspectives on Individual Differences*, pp. 219–233. Springer, New York (2001)
34. Zhang, H.: The optimality of naive bayes. In: *Proceedings of FLAIRS* (2004)

# Mastery-Oriented Shared Student/System Control Over Problem Selection in a Linear Equation Tutor

Yanjin Long<sup>1</sup>(✉) and Vincent Alevén<sup>2</sup>

<sup>1</sup> Learning Research and Development Center, University of Pittsburgh,  
3939 O'Hara Street, Pittsburgh, PA 15213, USA  
ylong@pitt.edu

<sup>2</sup> Human-Computer Interaction Institute, Carnegie Mellon University,  
5000 Forbes Avenue, Pittsburgh, PA 15213, USA  
aleven@cs.cmu.edu

**Abstract.** Making effective problem selection decisions is a challenging Self-Regulated Learning skill. Students need to learn effective problem-selection strategies but also develop the motivation to use them. A mastery-approach orientation is generally associated with positive problem selection behaviors such as willingness to work on new materials. We conducted a classroom experiment with 200 6th – 8th graders to investigate the effectiveness of shared control over problem selection with mastery-oriented features (i.e., features that aim at fostering a mastery-approach orientation that simulates effective problem-selection behaviors) on students' domain-level learning outcomes, problem-selection skills, enjoyment, future learning and future problem selection. The results show that shared control over problem selection accompanied by mastery-oriented features leads to significantly better learning outcomes, as compared to fully system-controlled problem selection, as well as better declarative knowledge of a key problem-selection strategy. Nevertheless, there was no effect on future problem selection and future learning. Our experiment contributes to prior literature by demonstrating that with tutor features to foster a mastery-approach orientation, shared control over problem selection can lead to significantly better learning outcomes than full system control.

**Keywords:** Mastery-approach orientation · Problem selection · Self-Regulated Learning · Learner control · Classroom experiment · Intelligent Tutoring System

## 1 Introduction

Intelligent Tutoring Systems often are strongly system-controlled learning environments that adaptively select problems for students based on their knowledge level [13]. Recently, some ITSs have started to grant students control to select their own learning tasks to elicit higher motivation, which in turn may lead to better learning outcomes [6]. However, prior research has found that students are not good at making effective problem selection decisions [9]. Fully student-controlled problem-selection was found

to lead to worse learning outcomes than system-selected problems [2]. Hence some ITSs created shared student/system control over problem selection (e.g., letting the system pick problem types while the students select a specific problem from that type) to prevent students from making suboptimal decisions and achieved comparable learning outcomes to those achieved with full system control [6]. It is still an open question how ITSs can be designed to foster better learning outcomes and higher motivation with shared control over problem selection, as compared to full system control. In addition, theories of SRL emphasize the important role of motivation in promoting desirable SRL behaviors [15]. Yet little work with ITSs has adopted a motivational design (i.e., design to foster motivations) approach to foster appropriate motivations that will stimulate effective problem-selection behaviors. Most of the interventions that support SRL processes in ITSs use cognitive and metacognitive tools, such as prompts and feedback [3, 11]. Furthermore, few of these studies have measured the lasting effects of the interventions when they are not in effect [1].

We tackle these open questions by applying motivational design to extend an ITS for equation solving, *Lynnette*, to help students learn an effective problem-selection rule, i.e., the Mastery Rule, while fostering a mastery-approach orientation [8]. The Mastery Rule specifies that students should stop practicing a problem type once it is fully mastered. An ITS that implements this rule (in a system-controlled manner) led to better learning outcomes than a fixed curriculum [5]. Our prior classroom studies and interviews with students revealed that the lack of a mastery-approach orientation might be a main challenge that keeps students from applying the Mastery Rule when they can select problems for themselves [8]. A mastery-approach orientation is a type of achievement goal that is associated with positive learning behaviors such as perseverance and willingness to learn new materials [14]. It aligns with the desirable problem selection behaviors based on the Mastery Rule. It is likely, but unproven that students with a mastery-approach orientation will apply the Mastery Rule to select problems and achieve better learning outcomes in the tutor, as compared to full system control. We therefore added features to foster a mastery-approach orientation. We refer to these features as the mastery-oriented features.

The current paper describes our classroom experiment that investigated two research questions: **Research Question 1:** Compared to full system control over problem selection, does shared control, supported by mastery-oriented features enhance students' (a) problem-selection decisions in the tutor; (b) domain-level learning outcomes; (c) enjoyment and (d) knowledge of the Mastery Rule? **Research Question 2:** Do the mastery-oriented features enhance students' (a) *future* problem-selection decisions and (b) *future* domain-level learning outcomes in an environment with shared control but without mastery-oriented features, as compared to full system control?

## 2 Methods

### 2.1 Experimental Design

**The Learning Phase Versus the Future Learning Phase.** The classroom experiment used a two-phase design, with a Learning Phase and a Future Learning Phase, so that

we could investigate both immediate effects of mastery-oriented shared control (Research Question (1) and effects on future learning without the mastery-oriented features (Research Question (2)). We created three variations of *Lynnette* for different conditions in the two phases, *Lynnette-System*, *Lynnette-Shared* and *Lynnette-Shared-Mastery-Oriented*. *Lynnette-System* implements full system control over problem selection through Bayesian Knowledge Tracing (BKT) and Cognitive Mastery [5], as in standard ITS. Both *Lynnette-Shared* and *Lynnette-Shared-Mastery-Oriented* implement shared control over problem selection. As shown in Fig. 1, students are free to select any level they want to practice and decide how much practice they want for each level. Once the student selects a level, the tutor assigns a specific problem from the chosen level. Students are able to select problems even after they have fully mastered that level in these two versions (as calculated by the tutor’s BKT and displayed by the mastery bars for each level). Only *Lynnette-Shared-Mastery-Oriented* has the mastery-oriented features that we describe below. All three *Lynnette* versions have the element badges and mastery bars for each level (as seen in Fig. 1).

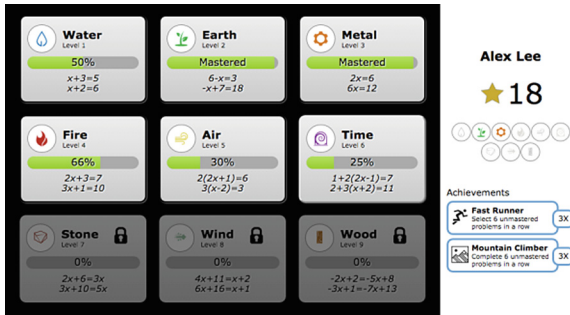


Fig. 1. Problem selection screen in *Lynnette-Shared-Mastery-Oriented* in the learning phase

The experiment started with two conditions in the Learning Phase, and only Levels 1 to 6 were unlocked in this phase. As shown in Table 1, the “Mastery Shared” condition used *Lynnette-Shared-Mastery-Oriented*, while the “Standard Tutor” used *Lynnette-System*. By comparing these two conditions, we can address Research Question 1, i.e., whether the mastery-oriented shared control leads to better problem-selection, learning and enjoyment as compared to full system control. In the Future Learning Phase, Levels 7 to 9 were also unlocked and the two conditions were split into four. Half of the participants from the “Mastery Shared” condition were assigned to use “*Lynnette-Shared*” and half use “*Lynnette-System*”. Similarly, half of the “Standard Tutor” condition switched to “*Lynnette-Shared*” and half continued using “*Lynnette-System*”. The four conditions in the second phase allowed us to investigate Research Question 2, i.e., the effects of the mastery-oriented features on students’ problem selection and learning outcomes when they are removed in new tutor units with shared control, compared to full system control.

**Table 1.** Conditions of the learning phase and the future learning phase

Learning Phase		Future Learning Phase	
Conditions	<i>Lynnette</i> Version	Conditions	<i>Lynnette</i> Version
Mastery Shared	<i>Lynnette-Shared-Mastery-Oriented</i>	Mastery to Shared	<i>Lynnette-Shared</i>
		Mastery to Standard	<i>Lynnette-System</i>
Standard Tutor	<i>Lynnette-System</i>	Standard to Shared	<i>Lynnette-Shared</i>
		Standard to Standard	<i>Lynnette-System</i>

**Mastery Oriented Features in *Lynnette*.** There are four mastery-oriented features in *Lynnette-Shared-Mastery-Oriented* that aim at helping students learn the Mastery Rule and foster a mastery-approach orientation [8]: **(1) Tutorial:** A tutorial is shown when students log in to the tutor for the first time. It introduces the concept of Mastery, the mastery bars, and how to apply the Mastery Rule to select problems. **(2) Achievements and Stars:** Two types of Achievements are implemented in the tutor to reward students' good problem selection decisions and perseverance with practicing new problems, as shown on the right panel of the screen in Fig. 1. Students earn the Achievements when they select or complete 6 problems in a row. In addition, the student earns a star each time s/he selects an unmastered problem. **(3) Instant Feedback Messages on Problem Selection Decisions:** Each time the student selects a problem, either a positive message (e.g., "Good problem selection decision! Water is still unmastered, so you can learn new skill from it. Don't be discouraged if you feel it is difficult. It is ok to make errors when you are learning!") or a negative message (e.g., "You've picked Earth but it is already mastered. Your equation solving skill will not grow if you repeat material you've already mastered.") will pop up and provide feedback on her/his choice. The language used in the messages emphasizes a mastery-approach orientation. **(4) Problem Selection Recap:** The problem selection recap screen (as shown in Fig. 2) is shown to the students after every 5th problem, in order to help students review and reflect on their recent problem selection decisions. The specific problem levels the student has selected are displayed with corresponding mastery bars showing the percentages of mastery at the time the student selected each level. The student also receives instant feedback on whether s/he has correctly clicked the unmastered levels. The names of the problem levels turn green or red when the student clicks. Green flags a correct click.

## 2.2 Procedure, Measurements and Participants

The experiment included 294 students from 5 local middle schools. The participants came from 16 classes, taught by 8 different teachers. Among the 16 classes, 4 were advanced 6th grade classes, 9 were mainstream 7th grade classes, and 3 were mainstream 8th grade classes. The participants were randomly assigned to one of the four



**Fig. 2.** The problem selection recap screen in *Lynnette-Shared-Mastery-Oriented*

conditions within each class before the experiment. All conditions followed the same procedure, summarized in Table 2, consisting of a Learning Phase and a Future Learning Phase. Three paper tests were given to measure different constructs before and after each phase of learning. Each equation on the three tests was graded from 0 to 1, with partial credit given where appropriate. The pre-test only had items from Levels 1 to 6. The mid-test and post-test had items that measure equation solving abilities for all 9 levels. The enjoyment questionnaire was adapted from the Enjoyment subscale of the Intrinsic Motivation Inventory (IMI). There were three check-box items on the mid-test to measure the students' declarative knowledge of applying the Mastery Rule. The first item tested the students' understanding of the concept of mastery. The second item described a scenario and tested whether the students would keep selecting problem levels that have been mastered. The third item also was scenario-based, and it tested whether the students were willing to challenge themselves with new problem types to learn new skills.

### 3 Results

200 students completed the pre-test and mid-test, and were present in all four class periods or mastered the first six levels during the Learning Phase. We refer to these 200 students as the Learning-Phase-Sample. 165 students completed the pre-test, mid-test and post-test. They were present during all 6 class periods (both the Learning and Future Learning Phases) or mastered all 9 levels. These students constitute the Future-Learning-Phase-Sample. We report Cohen's  $d$  for effect sizes. An effect size  $d$  of .20 is typically deemed a small effect, .50 a medium effect, and .80 a large effect. For all ANCOVAs, Teacher was used as a co-variate to account for the variances that reside within different teachers' classes.

#### 3.1 The Learning Phase: Research Questions 1.a – 1.d

We first analyzed data from the Learning Phase, to answer Research Questions 1.a-1.d. The Learning-Phase-Sample was used for all analyses.

**Table 2.** Overview of the procedure and measurements of the experiment

Pre-test	Learning phase	mid-test	Future learning phase	Post-test
<ul style="list-style-type: none"> <li>• 6 items on equation solving abilities of levels 1–6</li> </ul>	<ul style="list-style-type: none"> <li>• 4 41-min class periods</li> <li>• Learning the first 6 levels</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Mid-Test-Equations1:</b> 6 items on levels 1–6</li> <li>• <b>Mid-Test-Equations2:</b> 3 items on levels 7–9</li> <li>• 7 7-point Likert scale items on enjoyment of using the system</li> <li>• 3 items on declarative knowledge of applying the Mastery Rule</li> </ul>	<ul style="list-style-type: none"> <li>• 2 41-min class periods</li> <li>• All 9 levels were unlocked</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Post-Test-Equations1:</b> 6 items on levels 1–6</li> <li>• <b>Post-Test-Equations2:</b> 3 items on levels 7–9</li> </ul>

**Problem Selection Decisions (RQ 1.a).** To test the hypothesis that mastery-oriented features will help foster more consistent application of the Mastery Rule, we looked at the percentage of mastered problems the students selected in the “Mastery Shared” condition during the Learning Phase (under perfect application of the Mastery Rule, the students should not select any mastered problems). Twenty out of 102 students (19.61 %) in the “Mastery Shared” condition selected at least one mastered problem during the Learning Phase. On average 1.4 % of the problems (SD = 3.8 %) selected by each student in the condition were mastered problems, indicating good application of the Mastery Rule when the mastery-oriented features were present.

**Learning Outcomes (RQ 1.b).** To test the hypothesis that mastery-oriented shared control over problem selection will lead to greater learning gains than full system control, we compared the two conditions’ test performance on equation solving. As shown in Table 3, both conditions scored close to ceiling on the pre-test. An ANCOVA using the learning gain (Mid-Test-Equations1 minus Pre-Test) as the dependent variable revealed that the main effect of condition is significant ( $F(1, 192) = 4.486, p = .035, d = .30$ ). In other words, The “Mastery Shared” condition learned significantly more during the Learning Phase than the “Standard Tutor” condition. However, given the ceiling effect, the students did not improve significantly from pre-test to mid-test on solving the equations.

Given the ceiling effect on the pre-test, we split the sample based on the median of the pre-test score (median = .83) into two sub-groups: the Lower-Performing Group and the Higher-Performing Group. The Lower-Performing Group had 102 students (mean pre-test = 0.67, SD = 0.18), and the Higher-Performing Group had 98 students (mean pre-test = 0.98, SD = 0.05). ANCOVAs revealed that overall the two conditions improved significantly from pre-test to mid-test on Equations1 within the Lower Performing Group ( $F(1, 94) = 13.451, p < .000, d = .76$ ). The condition effect was marginally significant ( $F(1, 94) = 3.490, p = .065, d = .37$ ), with the “Mastery Shared” condition improving more than the “Standard Tutor” condition. On the other hand, there was a significant decrement of the two conditions’ performance from the pre-test to mid-test within the Higher-Performing Group ( $F(1, 90) = 25.704, p < .000$ ,



**Table 3.** Means and SDs for test performance of levels 1–6 equations on pre-test and mid-test

	All sample		Lower-performing		Higher-performing	
	Pre-test	Mid-test-Equations1	Pre-test	Mid-test-Equations1	pre-test	Mid-test-Equations1
Mastery shared	0.81 (0.21)	0.85 (0.20)	0.68 (0.20)	0.80 (0.22)	0.98 (0.04)	0.91 (0.14)
Standard tutor	0.84 (0.19)	0.81 (0.21)	0.66 (0.16)	0.70 (0.22)	0.98 (0.05)	0.91 (0.15)

$d = 1.07$ ), probably representing regression to the mean. No significant condition effect was found for the learning gains within the Higher-Performing Group.

**Enjoyment (RQ 1.c).** To test the hypothesis that mastery-oriented shared control over problem selection will lead to higher enjoyment of using the tutor than full system control, we compared students’ enjoyment ratings on the mid-test. The “Mastery Shared” condition reported higher enjoyment (mean = 4.63, SD = 1.59) than the “Standard Tutor” (mean = 4.52, SD = 1.36). However, an ANCOVA test found the difference was not statistically significant ( $F(1, 192) = .450, p = .530, d = .09$ ).

**Declarative Knowledge (RQ 1.d).** To test the hypothesis that the mastery-oriented features with shared control will lead to better knowledge of the Mastery Rule, compared to full system control, we analyzed the students’ responses to the three items on the mid-test. There were 12 options for all three items. The students were instructed to check all options that apply. We coded the students’ responses to each option as 0 or 1. On average those in the “Mastery Shared” condition (mean = 0.76, SD = 0.15) scored significantly higher ( $F(1, 184) = 8.263, p = .005, d = .59$ ) than those in the “Standard Tutor” condition (mean = 0.69, SD = 0.17). The “Mastery Shared” condition showed significantly better declarative knowledge of the Mastery Rule on the mid-test after the Learning Phase.

### 3.2 The Future Learning Phase: Research Questions 2.a and 2.B

We performed analyses on students’ problem selection decisions and equation solving performance. The Future-Learning-Phase-Sample was used for all analyses.

**Problem Selection Decisions (RQ 2.a).** We tested the hypothesis that the students exposed to the mastery-oriented shared control over problem selection in the Learning Phase will transfer and apply the Mastery Rule during the Future Learning Phase with the shared control. Specifically, we compared students’ problem-selection decisions between the “Mastery to Shared” condition and the “Standard to Shared” condition. In the “Mastery to Shared” condition, 15 out of 49 students (30.61 %) selected at least one mastered problem during the Future Learning Phase, whereas in the “Standard to Shared” condition, 7 out of 35 (20 %) students selected at least one mastered problem. Moreover, on average 2.7 % of the problems selected by the “Mastery to Shared” condition were mastered, while 1.6 % selected by the “Standard to Shared” condition were mastered. Nevertheless, an ANCOVA test revealed that the difference between the percentages of these two conditions was not statistically significant.

**Learning Outcomes (RQ 2.b).** To test the hypothesis that shared control over problem selection (without mastery-oriented features) will lead to better learning outcomes in the Future Learning Phase, compared to full system control, we performed ANCOVAs to analyze students' learning gains from the mid-test to post-test. Two independent variables were used in the ANCOVA analyses: (1) whether the students had mastery-oriented shared control or full system control over problem selection in the Learning Phase, and (2) whether they had shared versus system control during the Future Learning Phase. As shown in Table 4, the students' performance on Equations1 did not change much from mid-test to post-test. An ANCOVA revealed no significant improvement from the mid-test to post-test for Equations1 for the four conditions. Also, no significant main effects or interaction were found for Equations1 with the two independent variables. On the other hand, overall the four conditions improved significantly on Equations2 from mid-test to post-test ( $F(1, 155) = 37.028, p < .000, d = .98$ ), as well as the whole test (with Equations1 and Equations2 together,  $F(1, 155) = 16.839, p < .000, d = .66$ ). However, no significant main effects or interaction were found between the conditions for Equations2 or the whole test.

**Table 4.** Means and SDs for mid-test and post-test equation solving items

	Mid-test-equations1	Post-test-equations1	Mid-test-equations2	Post-test-equations2
Mastery to shared	0.82 (0.23)	0.80 (0.24)	0.38 (0.40)	0.58 (0.40)
Mastery to standard	0.86 (0.16)	0.85 (0.18)	0.36 (0.40)	0.59 (0.40)
Standard to shared	0.82 (0.20)	0.86 (0.16)	0.34 (0.41)	0.56 (0.43)
Standard to standard	0.84 (0.20)	0.86 (0.22)	0.46 (0.45)	0.59 (0.38)

## 4 Discussion, Conclusions and Future Work

Our classroom experiment investigated whether mastery-oriented shared control over problem selection would foster the learning of an effective problem selection strategy, students' learning outcomes and enjoyment, as well as future problem selection and future domain-level learning. We found that shared control over problem selection, while it was supported with mastery-oriented features, led to better learning outcomes as compared to full system control in an ITS. Specifically, during the Learning Phase, those in the mastery-oriented shared control condition improved significantly more than those in the system-controlled condition on equation solving. Although the two conditions overall did not improve significantly due to the ceiling effects on the pre-test. Within the lower-performing group, there were significant learning gains from pre-test to mid-test, and the condition effect was marginally significant. These results prove that shared

control accompanied by mastery-oriented features can significantly benefit students' domain level learning, especially for students with low prior knowledge. How did the mastery-oriented shared control over problem selection lead to greater learning gains? First, the students with the mastery-oriented shared control selected almost the same problems as the system control. They rarely violated the Mastery Rule, put differently, the students selected mostly unmastered problems as the Cognitive Mastery algorithm does for the system control. Therefore, we can mostly rule out the possibility that the difference in learning gains was due to differences in the problem sequences being practiced. Second, it is likely that the mastery-oriented features (tutorial, feedback, achievements and problem selection recap screens) might have encouraged the students to adopt metacognitive strategies such as reviewing, reflecting or summarizing, as a mastery-approach orientation has been found to be positively associated with use of such strategies [14]. Prior work has generally found that students with a mastery-approach orientation achieve better learning outcomes, compared to their counterparts who focused more on performance relative to others, i.e., with a performance orientation [12].

We also found that the mastery-oriented shared control resulted in significantly better declarative knowledge of the Mastery Rule, as compared to the full system control condition. It could possibly be attributed to the explicit instructions and motivational messages from the four mastery-oriented features. On the other hand, the mastery-oriented shared control did not lead to significantly higher enjoyment of using the tutor as compared to the full system-controlled tutor. It is likely that the badges, as well as the mastery bars implemented in the system-controlled condition also made it enjoyable to students. Prior work on learner control emphasizes its motivational benefits to students [4], but our finding suggests that enabling learner control does not necessarily enhance students' enjoyment of the learning experience.

Although the mastery-oriented shared control enhanced students' learning while it was in effect, no lasting effect on learning was found with only shared control over problem selection. For the Future Learning Phase, no significant condition effects were observed for learning gains on equation solving. In other words, there was apparently no carry into the next unit of a possible motivational effect on student learning. Additionally, the equations in this phase were more difficult than the Learning Phase, and the learning time was reduced to 2 class periods. The students might experience higher cognitive load when learning more difficult equations within a shorter period of time, making it difficult to initiate metacognitive processes such as reviewing or reflecting that relate to a mastery-approach orientation.

Lastly, with respect to problem selection decisions, students with shared control exhibited good application of the Mastery Rule in both phases. The mastery-oriented shared control condition selected only about 1 % of mastered problems during the Learning Phase. Similarly, the two shared control conditions without the mastery-oriented features in the Future Learning Phase selected around 2 % of mastered problems regardless of whether or not they came from the mastery-oriented shared control condition. The results regarding problem selection decisions were slightly surprising, given that in our prior classroom study, students selected 34 % mastered problems when no Open Learner Model was presented [8]. In other prior work, we also found students admitting that they would keep selecting easy problems if

given control over problem selection [7]. There may be two reasons why students made overall good problem selection decisions in both phases: First, our informal classroom observations found that the badges and the mastery bars strongly encouraged the students to complete the levels without repeating already-mastered problems. Although these two features were designed to make the tutor more fun and reward students' equation solving progress, not to influence problem selection, they might have motivated the students to make problem-selection decisions based on the Mastery Rule. A second reason may have been that the environments for this experiment were not entirely self-regulatory. The students were learning in their math classes and the teachers sometimes gave informal instructions such as "now you should work on the newly unlocked levels". The students were practicing with a "goal" and supervision from their teachers, which might have influenced their problem selection decisions.

To sum up, the current experiment shows that shared control over problem selection accompanied by features that foster a mastery-approach orientation in an ITS leads to significantly better domain-level learning outcomes, as compared to full system control over problem selection, which is standard practice in ITS. This is a novel contribution to the literature on the effects of learner control on student learning, which has generally found that pure learner control leads to worse learning than system control [2, 10] and that shared control only resulted in learning outcomes that were comparable to system control [6]. On the other hand, our experiment did not establish lasting effects of the mastery-oriented features on future learning in a new tutor unit, with improvement only on declarative knowledge of applying the rule on an immediate paper test. Future work is warranted to further investigate how to design ITSs that support learning and motivation of Self-Regulated Learning processes that can transfer to new learning topics and environments.

**Acknowledgement.** We thank Gail Kusbit, Jonathan Sewall, Octav Popescu and Mike Stayton for their kind help with the classroom experiment. We also thank the participating teachers and students. This work is funded by an NSF grant to the Pittsburgh Science of Learning Center (NSF Award SBE0354420).

## References

1. Aleven, V., McLaren, B.M., Roll, I., Koedinger, K.R.: Help helps, but only so much: research on help seeking with intelligent tutoring systems. *Int. J. Artif. Intell. Educ.* **26**(1), 1–9 (2016)
2. Atkinson, R.C.: Optimizing the learning of a second-language vocabulary. *J. Exp. Psychol.* **96**(1), 124–129 (1972)
3. Azevedo, R., Witherspoon, A., Chauncey, A., Burkett, C., Fike, A.: MetaTutor: a meta-cognitive tool for enhancing self-regulated learning. In: *Proceedings of the AAAI Fall Symposium on Cognitive and Metacognitive Educational Systems*, pp. 14–19 (2009)
4. Clark, C.R., Mayer, E.R.: *E-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning*. Jossey-Bass, San Francisco (2011)
5. Corbett, A.: Cognitive mastery learning in the ACT programming tutor. AAAI Technical report, SS-00-01 (2000)

6. Corbalan, G., Kester, L., Van Merriënboer, J.J.G.: Selecting learning tasks: effects of adaptation and shared control on efficiency and task involvement. *Contemp. Educ. Psychol.* **33**(4), 733–756 (2008)
7. Long, Y., Aleven, V.: Active learners: redesigning an intelligent tutoring system to support self-regulated learning. In: *Proceedings of the 8th European Conference on Technology Enhanced Learning*, pp. 490–495 (2013)
8. Long, Y., Aman, Z., Aleven, V.: Motivational design in an intelligent tutoring system that helps students make good task selection decisions. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M. (eds.) *AIED 2015. LNCS*, vol. 9112, pp. 226–236. Springer, Heidelberg (2015)
9. Metcalfe, J.: Metacognitive judgments and control of study. *Curr. Dir. Psychol. Sci.* **18**(3), 159–163 (2009)
10. Niemiec, R.P., Sikorski, C., Walberg, H.J.: Learner-control effects: a review of reviews and a meta-analysis. *J. Educ. Comput. Res.* **15**(2), 157–174 (1996)
11. Roll, I., Aleven, V., McLaren, B.M., Koedinger, K.R.: Improving students' help-seeking skills using metacognitive feedback in an intelligent tutoring system. *Learn. Instr.* **21**(2), 267–280 (2011)
12. Schunk, D.H., Pintrich, P.R., Meece, J.L.: *Motivation in Education: Theory, Research, and Applications*. Pearson/Merrill Prentice Hall, Upper Saddle River (2008)
13. VanLehn, K.: The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educ. Psychol.* **46**(4), 197–221 (2011)
14. Wolters, C.A., Yu, S.L., Pintrich, P.R.: The relation between goal orientation and students' motivational beliefs and self-regulated learning. *Learn. Individ. Differ.* **8**(3), 211–238 (1996)
15. Zimmerman, B.J.: Attaining self-regulation: a social cognitive perspective. In: Boekaerts, M., Pintrich, P., Zeidner, M. (eds.) *Handbook of Self-Regulation*, pp. 1–39. Academic Press, San Diego (2000)

# Providing the Option to Skip Feedback in a Worked Example Tutor

Amruth N. Kumar<sup>(✉)</sup>

Ramapo College of New Jersey, 505 Ramapo Valley Road,  
Mahwah, NJ 07430, USA  
amruth@ramapo.edu

**Abstract.** Providing choice is known to intrinsically motivate learners and support self-regulated learning. In order to study the effect of providing the choice to skip feedback in an online tutor traditionally used in-natura, we conducted a controlled study in Fall 2015. Experimental group was given the choice to skip the worked example provided as feedback after the student had solved a problem incorrectly, whereas control group was not. We found that providing the choice did not lead to greater learning. Experimental group students needed marginally more problems to learn each concept, and their pre-post improvement was marginally less. When we analyzed skipping behavior, we found that neither the grade on a problem nor the grade on the prior problem on the same concept affected a student's decision to read or skip feedback. Novelty of the concept on the other hand may prompt students not to skip feedback. Whether or not students skipped feedback on a problem did not affect their grade on the next problem on the same concept. Students were just as likely to skip as not skip feedback on the various problems. Some students tended to skip far more than others.

**Keywords:** Worked example · Help-seeking · Intrinsic motivation · Programming tutor

## 1 Introduction

Choice intrinsically motivates learning [1]. In addition, it has been shown to increase engagement in learning, increase the amount learned in a fixed period of time and improve the learner's perceived competence and levels of aspiration [2]. In one recent study of choice in an intelligent tutor, when offered a choice of text versus video feedback, those who had the choice outperformed those who did not [3]. On the other hand, in another recent study, learners' help-seeking choices were not found to concur with the intention of the provided feedback [4].

We wanted to investigate the effect of providing the choice of reading or skipping feedback in an online problem-solving tutor on programming concepts. The tutor is adaptive: it uses a pretest to prime the student model, and adapts subsequent practice problems based on the student model [8]. Every time the student's answer is incorrect, after presenting the correct answer, it presents step-by-step

explanation of the correct answer, in the style of a worked example [9]. The tutor is typically used by introductory programming students on their own time, *in-natura*, as course assignment.

We had a few reasons to consider this to be a novel study of choice in tutors:

- The tutor associates each problem with one programming concept, e.g., nested selection statements, back-to-back selection statements. However, these concepts are interdependent. Therefore, we expect transfer of learning to occur between problems associated with different concepts, especially when the feedback is in the style of worked example wherein the entire solution is explained step by step. Given the potential for transfer of learning, we studied the effect of providing the student the choice of reading/skipping the step-by-step feedback. We hypothesized that such choice might enable students to skip feedback deemed unnecessary without hampering learning.
- The tutor is used in unsupervised setting by students on their own time, as after-class assignment. In such a setting, students are often more focused on completing the assignment quickly than on maximizing their learning [10]. So, when offered the choice to skip feedback, they may exercise it for varying reasons - some related to learning, while others may be related to the expediency of completing the assignment quickly.

## 2 The Study

### 2.1 Participants

The participants of the study were students in introductory programming courses from 27 institutions in Fall 2015. Since this was a controlled study, institutions were randomly assigned to control or experimental group. There were 254 students in the control group and 341 students in the experimental group who granted IRB permission to be part of the study.

### 2.2 Instrument

The instrument used for this study was a software tutor on selection statements in C++/Java/C#. The tutor presents a program containing one or more selection statements; has the student predict the output of the program one at a time, along with the line in the program that produces that output; and grades the student's answer. If the student's answer is incorrect, it also provides step-by-step explanation of the correct answer in the style of a worked example.

The tutor covers 12 concepts on one-way and two-way selection statements, including: execution of the statement when the condition is true/false; classification/cascading style nesting; multiple statements appearing back-to-back, and special cases of the condition of the statement being a declaration/assignment expression (C++ only). As mentioned earlier, these concepts are interdependent, e.g., nested and multiple selection statements depend on an understanding of the execution of the statement when the condition is true/false.

The tutor is accessible over the web - students can use it on their own time, as often as they please. It is part of a suite of problem-solving tutors for introductory programming topics called problets ([www.problets.org](http://www.problets.org)).

### 2.3 Protocol

The software tutor administered pretest-practice-post-test protocol as follows:

- **Pretest:** During pretest, the tutor presented one problem per concept to prime the student model. If a student solved a problem correctly, no step-by-step explanation was provided to the student, and no more problems were presented to the student on the concept. On the other hand, if the student solved a problem partially correctly, incorrectly, or opted to skip the problem because the student did not know the answer, step-by-step feedback was presented to the student and additional problems on the concept were scheduled to be presented during practice stage.
- **Adaptive Practice:** Once a student had solved all the pretest problems, practice problems were presented to the student on only the concepts on which the student had solved problems incorrectly during pretest. For each such concept, the student was presented multiple problems until the student had mastered the concept, i.e., solved at least 60 % of the problems correctly. After each problem the student solved incorrectly, the student received feedback listing the correct answer, followed by step-by-step explanation of the correct answer. Since this was a controlled study, experimental group had the option to skip the step-by-step feedback whereas control group did not.
- **Adaptive Post-test:** During this stage, which was interleaved with practice, the student was presented a test problem each on the concepts that the student had mastered during practice. No problems were presented on the concepts the student did not master during practice and the concepts on which the student had solved the problem correctly during pretest.

Table 1 illustrates a typical sequence of problems solved by a student. The student solves problem no. 1 on concept no. 1 incorrectly, problem 2 on concept 2 correctly, problem 3 on concept 3 partially, and does not know the answer to problem 4 on concept 4, all during pretest, which consists of 12 problems. During adaptive practice, the tutor presents problems 13, 14 and 18 on concept 1 and problem 15 on concept 3, and the student satisfies mastery criterion on both these concepts. So, the tutor schedules post-test problems 19 and 21 on concepts 3 and 1 respectively, while continuing to present practice problems on the not-yet-mastered concept 4.

The entire protocol was limited to 30 min and was administered back-to-back, entirely over the web. Since practice and post-test were adaptive, students who solved at least one problem incorrectly on the pretest spent a mean of 19.35 min using the tutor whereas experimental group spent a mean of 18.78 min. These figures include students who ran out of time.

A concept was considered to have been *practiced* during this session if the student solved the problem on that concept incorrectly during the pretest, solved



**Table 1.** A typical problem sequence

	Pretest					Adaptive practice/post-test									
Problem no.	1	2	3	4	...	13	14	15	16	17	18	19	20	21	
Concept no.	1	2	3	4	...	1	1	3	4	4	1	3	4	1	
Grade	Incr	Corr	Part	NotK	...	Incr	Corr	Corr	Incr	Incr	Corr	Post	Corr	Post	

enough problems during adaptive practice to master the concept, and solved a problem on that concept during post-test, e.g., concepts 3 and 1 in Table 1. If the student solved the problem correctly during post-test, the practiced concept was also considered to have been **learned**.

## 2.4 Design

We conducted two types of analysis: (1) **aggregate analysis** to see if the treatment, i.e., having the option to skip feedback, affected overall learning; (2) **problem-level analysis** to see what motivates experimental group students to skip feedback, and whether skipping feedback affected subsequent grades. For aggregate analysis, we used:

- Pretest score per problem to verify that the control and experimental groups were comparable;
- The number of concepts learned as a measure of the amount of learning;
- The number of practice problems solved per learned concept, calculated as the number of practice problems solved by a student on all the learned concepts, divided by the number of concepts learned by the student: the more the problems, the slower the pace of learning and vice versa;
- Pre-post change per learned concept as a measure of improvement in learning;
- The number of pretest problems solved and the time spent per pretest problem - to assess the impact of treatment on the pace of solving problems during pretest.

The only independent variable considered was treatment: whether students could or could not skip the step-by-step feedback.

For problem-level analysis, we considered every pair of successive problems solved by a student on a concept, where the solution to one of the problems was partial/incorrect or the student did not know the solution. For example, in Table 1, we considered the following pairs of problems: (1,13) and (13,14) for concept 1, (3,15) for concept 3, and (4,16), (16,17) and (17,20) for concept 4. Note that the two problems could be back to back or removed by several intermediate problems on other concepts. For these pairs, we used the grade (coded as ordinal values 0 for incorrect, 0.5 for partial and 1 for correct), and time spent per problem as dependent variables. When we considered only experimental group data, the independent variable was whether or not students skipped feedback. When both control and experimental groups were considered, the independent variable was treatment, i.e., the option to skip feedback.

## 2.5 Data Collection

Students could use the tutor as often as they pleased. If a student used the tutor multiple times, for aggregate analysis, we considered data from only the first time when the student had solved all the pretest problems. If the student never solved all the pretest problems, we considered data from the attempt with the most number of pretest problems solved.

On the other hand, for problem-level analysis, we considered every session of a student where the student had solved one or more problems incorrectly. In all, control group solved 573 problems incorrectly, which constituted 497 sequences of problem pairs on the same concept. Experimental group solved 1350 problems incorrectly, which constituted 1598 sequences of problem pairs. The two-problem sequence count is higher because some problems could be part of two pairs - one with the previous problem on the concept, and one with the next problem, e.g., problem 13 on concept 1 is part of (1,13) and (13,14) in Table 1. The numbers are greater for experimental group because the group was larger, and students in this group used the tutor more often per capita (2.00 versus 1.82 for control group).

## 2.6 Data Analysis

A typical program produces a sequence of outputs. The tutor awarded grade for each problem as: (the number of outputs correctly identified in proper sequence - number of incorrect outputs identified)/total number of outputs in the problem. Therefore, for aggregate analysis of learning, the score on each problem was normalized to  $0 \rightarrow 1.0$  regardless of the number of outputs in the program. Univariate ANOVA was used to analyze the data.

# 3 The Results

## 3.1 Aggregate Analysis

To begin with, we analyzed the data to see if treatment, i.e., having the option to skip feedback affected learning. Since feedback and therefore, the option to skip feedback was provided only when the student solved a problem incorrectly, we eliminated from analysis, all the students who had solved all the problems correctly (130 in control group and 114 experimental group), because they did not receive any feedback from the tutor on any problem.

**Effect of Treatment on Problem-solving:** We found a significant difference in the time spent per pretest problem between the two groups [ $F(1,462) = 8.435$ ,  $p = 0.004$ ]: control group spent a mean of  $81.05 \pm 6.67$  sec per pretest problem whereas experimental group spent  $69.43 \pm 4.37$  sec. This is explained by the fact that experimental group had the option to skip feedback when their answer was incorrect. *So, students spent less time per pretest problem when they were provided the option to skip feedback.* Naturally, we also found a significant difference in the number of pretest problems attempted [ $F(1,462) = 4.442$ ,  $p$

= 0.036]: control group attempted  $8.01 \pm 0.35$  problems whereas experimental group attempted  $8.45 \pm 0.228$  problems. However, there was no significant difference in the score per pretest problem between the two groups [.8011 versus .8127,  $F(1,462) = 0.566$ ,  $p = 0.452$ ]. So, *the two groups were comparable vis-a-vis prior preparation.*

**Effect of Treatment on Learning:** Furthermore, there was no significant difference in the number of concepts the two groups learned [ $F(1,296) = 0.893$ ,  $p = 0.345$ ]: control group learned a mean of  $1.52 \pm 0.155$  concepts whereas experimental group learned  $1.43 \pm 0.105$  concepts. So, *the option to skip feedback did not lead to greater learning.* The difference in the number of practice problems solved per learned concept was marginally significant [ $F(1,296) = 2.664$ ,  $p = 0.104$ ]: control group solved  $3.21 \pm 0.322$  problems per learned concept whereas experimental group solved  $3.56 \pm 0.244$  problems. So, *when provided with the option to skip feedback, students solved marginally more problems to learn each concept.* The difference in the pre-post improvement in score on the concepts learned was marginally significant [ $F(1,296) = 3.129$ ,  $p = 0.078$ ]: control group score improved by  $0.86 \pm 0.053$  whereas experimental group score improved by  $0.80 \pm 0.037$ . So, *students improved marginally less when provided with the option to skip feedback.*

### 3.2 Problem-Level Analysis

Next, we considered what may have encouraged experimental group students to skip feedback, how skipping feedback affected their grade on the subsequent problem on the same concept, and whether skipping feedback was a behavior specific to problems or learners.

**Effect of Current Grade on Skipping:** We analyzed *all* the problems that experimental group students solved incorrectly. We found that there was no significant effect of the grade (incorrect or partially correct), on whether students skipped or did not skip the subsequent feedback [ $F(1,1116) = 0.484$ ,  $p = 0.487$ ]. *In other words, incorrect versus partially correct grade on a problem seemed to have no bearing on whether students chose to read or skip the feedback provided after solving it incorrectly.*

**Effect of Previous Grade on Skipping:** May be the decision to skip the feedback was influenced by prior exposure to the concept? The grade of the student on the first problem in a two-problem sequence (as described at the end of Sect. 2.4) might reveal the motivation of the student to skip or not skip feedback on the second problem even after solving the second problem incorrectly.

In Table 2, for each type of grade on the first problem, the number of second problems are listed on which the student skipped/did not skip feedback. No definitive pattern emerges in the data to suggest when students may or may not skip feedback based on previous problem's grade. We analyzed the grades (Correct/Partial/Incorrect) on previous and current problems as repeated measures for the experimental group, while treating as between-subjects factor,

**Table 2.** Counts of reading/skipping feedback based on grade on the previous problem

	Did not skip	Skipped feedback
Not know	24	16
Incorrect	81	91
Partial	122	104
Correct	98	96
Total	325	307

whether the student skipped feedback on the current problem. While a significant drop was observed in the grade from prior problem ( $0.517 \pm 0.033$ ) to current problem ( $0.308 \pm 0.021$ ) [ $F(1,552) = 119.974, p < .001$ ], the interaction between grade and whether feedback was skipped was not significant [ $F(1,552) = 0.765, p = 0.382$ ]. *In other words, the grade on the previous problem on the same concept did not seem to affect a student’s decision on whether or not to skip feedback on the current problem.*

**Effect of Novelty of the Concept on Skipping:** In contrast, when we considered cases where experimental group students solved only one problem on a concept, students skipped feedback on 105 problems, and did **not** skip feedback on 168 problems. So, students seemed to prefer to read the feedback more often than not when their answer was incorrect on the first and only problem on a concept. *This suggests that when students see a problem on a concept for the first time, the novelty of the concept may prompt students not to skip feedback.*

**Effect of Skipping on the Next Grade:** How did skipping feedback affect the grade on the next problem on the same concept? In Table 3, rows refer to grade on the first of the two problems. Columns refer to the grade on the second problem.

We analyzed the (incorrect/partial/correct) grade on current and next problem as the repeated measure and whether the experimental group student skipped feedback on the current problem as the within-subjects factor. We found a significant improvement in grade from current ( $0.243 \pm 0.017$ ) to next prob-

**Table 3.** Experimental group’s grade on the next problem on the same concept

	Without skipping					With skipping				
	Total	Don’t	Incor	Partl	Corr	Total	Don’t	Incor	Partl	Corr
Don’t Know	49	14	6	6	23	40	3	6	5	26
Incorrect	210	4	49	22	135	244	10	42	45	147
Partial	221	5	11	112	93	202	2	23	73	104
Total	480	23	66	140	251	486	15	71	123	277

**Table 4.** Improvement in score from current problem to the next problem on the same concept

	Current score	Next score
Control group	0.196 ± 0.029	0.766 ± 0.024
Experimental group	0.243 ± 0.017	0.707 ± 0.025

lem ( $0.707 \pm 0.025$ ) [ $F(1,854) = 912.768$ ,  $p < .001$ ], but found no main effect for whether students skipped reading the feedback on the current problem or not [ $F(1,854) = 0.118$ ,  $p = 0.731$ ], and no significant interaction [ $F(1,854) = 1.763$ ,  $p = 0.185$ ]. *In other words, whether students skipped or did not skip reading the feedback on the current problem did not affect their grade on the next problem.*

**Effect of Treatment on the Improvement in Grade from Current to Next Problem:** When we compared the grades on current and next problems of experimental group and control group, we found significant interaction between scores and treatment [ $F(1,1145) = 12.619$ ,  $p < 0.001$ ] as shown in Table 4. The improvement in score from current to next problem was greater for those who did not have the option to skip versus those who did. *In other words, providing the option to skip feedback led to smaller improvement in learning*, which confirms the earlier result from aggregate analysis.

**Effect of Skipping on the Time Spent per Problem:** We analyzed the time spent per problem by the experimental group with versus without skipping the feedback. As was to be expected, *students spent marginally less time on a problem they solved incorrectly when they skipped feedback ( $73.22 \pm 18.35$  sec) than when they did not ( $97.08 \pm 18.24$  sec)* [ $F(1,1297) = 3.271$ ,  $p = 0.071$ ].

**Effect of Treatment on the Time Spent per Problem:** When we analyzed the time spent per problem with treatment as the between-subjects factor, we found no significant main effect of treatment [ $F(1,1812) = 0.119$ ,  $p = 0.73$ ]: *the time spent per problem that was solved incorrectly was about the same whether feedback was mandatory ( $88.93 \pm 17.82$  sec) or optional ( $85.23 \pm 11.23$  sec).*

Experimental group students spent marginally less time per problem when they skipped feedback than when they did not. But, they spent about the same time per problem as control group. So, students who skipped feedback on some problems spent the time saved by doing so on reading the feedback provided for other problems that were also solved incorrectly.

**Variation of Skipping by Problem:** May be students were more likely to skip feedback on some problems than others? In all, 65 different problems on 12 different concepts were solved by experimental group students. We tabulated the number of times experimental group students skipped versus did not skip feedback on each of the 65 problems. We conducted repeated-measures ANOVA with skipping as the within-subjects factor, to find no significant main effect for skipping [ $F(1,129) = 0.082$ ,  $p = 0.775$ ]. *In other words, students were just as likely to skip as not skip feedback on the various problems.*

**Table 5.** Experimental group quartiles and their behavior skipping feedback

	Total problems	Feedback skipped	Mean percent	Students
Top quartile	404	360	94.81 %	86
Third quartile	325	193	56.58 %	63
Second quartile	184	63	33.34 %	30
Bottom quartile	196	29	15.17 %	17
Zero Percent	241			83

**Variation of Skipping by Student:** May be some students were more likely to skip feedback than others? Experimental group students were grouped into four quartiles based on the percentage of problems on which they skipped feedback. Data for these four groups and the group that never skipped feedback is shown in Table 5. This includes the total number of problems solved incorrectly, the number of problems on which feedback was skipped, the mean of the percentage of problems on which feedback was skipped and the number of students in each group. Clearly, *skipping feedback varies by person, i.e., some students skip more than others*: each quartile skipped feedback on more problems than all the lower quartiles combined. The same cannot be said about the total number of problems solved incorrectly or the number of students.

### 3.3 Discussion

This was a study of providing students the option to skip feedback, the feedback being step-by-step explanation of the correct answer in the vein of worked examples; and the study was conducted *in-natura*. Given the potential for transfer of learning between concepts in programming domain, and the expository nature of step-by-step feedback, we hypothesized that such choice would enable students to skip feedback that they deemed unnecessary without hampering their learning.

However, the results of the study did not support this hypothesis. Students who were provided the choice of skipping feedback did not learn more. Rather, their pre-post improvement was marginally less and they needed to solve marginally more problems for each concept they learned. Given that 30.82 % of the students skipped feedback on *every* problem they solved incorrectly, the provision of choice may have turned into an instrument for gaming the system [7] when the tutor was used *in-natura*: learning from the tutor may have taken a back seat to quickly completing the tutoring session.

Students do not know when they need help [5]. They must be explicitly taught how to seek help [6]. Considering that fully 29.74 % of the students in the current study *never* skipped feedback, we may have to also teach students when they should *decline* feedback. A happy middle-ground may be to adaptively fade out worked example feedback. Feedback may also have to be tailored to the characteristics of the learner (e.g., learning-orientation versus performance-orientation)

[11]. Future work includes investigating the effectiveness of fading-out and tailoring in *in-natura* use of the tutor.

**Acknowledgments.** Partial support for this work was provided by the National Science Foundation under grant DUE 1432190.

## References

1. Patall, E.A., Cooper, H., Robinson, J.C.: The effects of choice on intrinsic motivation and related outcomes: a meta-analysis of research findings. *Psychol. Bull.* **134**(2), 270–300 (2008)
2. Cordova, D.I., Lepper, M.R.: Intrinsic motivation and the process of learning: beneficial effects of contextualization, personalization, and choice. *J. Educ. Psychol.* **88**(4), 715–730 (1996)
3. Ostrow, K.S., Heffernan, N.T.: The role of student choice within adaptive tutoring. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M.F. (eds.) AIED 2015. LNCS, vol. 9112, pp. 752–755. Springer, Heidelberg (2015)
4. Gross, S., Pinkwart, N.: How do learners behave in help-seeking when given a choice? In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M.F. (eds.) AIED 2015. LNCS, vol. 9112, pp. 600–603. Springer, Heidelberg (2015)
5. Alevan, V., Koedinger, K.: Limitations of student control: do students know when they need help? *Proc. ITS* **2000**, 292–303 (2000)
6. Alevan, V., McLaren, B., Roll, I., Koedinger, K.: Toward meta-cognitive tutoring: a model of help seeking with a cognitive tutor. *IJAIED* **16**(2), 101–128 (2006)
7. Baker, R.S., Corbett, A.T., Koedinger, K.R.: Detecting student misuse of intelligent tutoring systems. In: Lester, J.C., Vicari, R.M., Paraguaçu, F. (eds.) ITS 2004. LNCS, vol. 3220, pp. 531–540. Springer, Heidelberg (2004)
8. Kumar, A.: A scalable solution for adaptive problem sequencing and its evaluation. In: Wade, V.P., Ashman, H., Smyth, B. (eds.) AH 2006. LNCS, vol. 4018, pp. 161–171. Springer, Heidelberg (2006)
9. Kumar, A.N.: Explanation of step-by-step execution as feedback for problems on program analysis, and its generation in model-based problem-solving tutors. *Technol. Instruct. Cognit. Learn. (TICL) J. Special Issue on Problem Solving Support in Intelligent Tutoring Systems*, 4(1) (2006)
10. Kumar, A.N.: All that glitters (in the lab) may not be gold (in the field). In: AIED 2015 Workshop on “Les Contes du Mariage: Should AI Stay Married to Ed?”, Madrid, Spain, pp. 21–27, June 2015
11. Shute, V.J.: Focus on formative feedback. *Rev. Educ. Res.* **78**(1), 153–189 (2008)

# Tell Me How to Teach, I'll Learn How to Solve Problems

Noboru Matsuda<sup>1</sup>(✉), Nikolaos Barbalios<sup>1</sup>, Zhengzheng Zhao<sup>1</sup>,  
Anya Ramamurthy<sup>2</sup>, Gabriel J. Stylianides<sup>3</sup>,  
and Kenneth R. Koedinger<sup>2</sup>

<sup>1</sup> College of Education and Human Development,  
Texas A&M University, College Station, USA  
{Noboru.Matsuda, zhengzhaoap}@tamu.edu,  
nbarmpalios@gmail.com

<sup>2</sup> Human-Computer Interaction Institute,  
Carnegie Mellon University, Pittsburgh, USA  
anya.ramamurthy@gmail.com, koedinger@cs.cmu.edu

<sup>3</sup> Department of Education, University of Oxford, Oxford, UK  
gabriel.stylianides@education.ox.ac.uk

**Abstract.** In this paper we study the effect of adaptive scaffolding to learning by teaching. We hypothesize that learning by teaching is facilitated if (1) students receive adaptive scaffolding on how to teach and how to prepare for teaching (the metacognitive hypothesis), (2) students receive adaptive scaffolding on how to solve problems (the cognitive hypothesis), or (3) both (the hybrid hypothesis). We conducted a classroom study to test these hypotheses in the context of learning to solve equations by teaching a synthetic peer, SimStudent. The results show that the metacognitive scaffolding facilitated tutor learning (regardless of the presence of the cognitive scaffolding), whereas cognitive scaffolding had virtually no effect. The same pattern was confirmed by two additional datasets collected from two previous school studies we conducted.

**Keywords:** Teachable agent · Learning by teaching · Algebra · Adaptive scaffolding · SimStudent

## 1 Introduction

Learning by teaching [1] is known to be effective with empirical evidence of students learning by teaching their peers in various domains [2], across different student populations [3], with different types of interactions and formats of tutoring [4]. In this paper, we use the term *tutor learning* to refer to the effect of learning by teaching on the tutor (i.e., a student who teaches his/her peer) [5]. For the currently study, our focus is on tutor learning in the mathematical domain of solving linear equations.

There has been growing interest in the application of teachable agents to study the effect of tutor learning, in particular in the field of artificial intelligence in education and human-computer interaction [6–8]. Teachable agents are synthetic peers that students can interactively teach.



Using the teachable agent technology, researchers try to understand the effect of tutor learning by, for example, mining stereotypic patterns of effective tutoring interactions [9], analyzing cognitive factors that contribute to tutor learning [8], and studying student's perceptions and motivations while interacting with synthetic peers [10]. Yet, the underlying cognitive mechanism of tutor learning is not fully understood. Without clear understanding of what makes learning by teaching effective, it is impractical to build a technology to facilitate tutor learning despite its promising potential for efficacy and large-scale dissemination.

Learning by teaching is a complicated phenomenon with many factors to be explored. As part of our on-going effort to contribute to advancing cognitive and social theory of tutor learning, the goal of the current paper is to study the effect of *adaptive scaffolding* to facilitate tutor learning, which is motivated by our past study findings as described in the next section.

## 2 Learning by Teaching: Lessons Learned

To understand how and why students learn by teaching others, we have developed an online learning environment for learning to solve equations by teaching, called APLUS (described in Sect. 4). In APLUS, students learn by teaching a synthetic peer, called SimStudent [8]. Prior to the current study, APLUS has been used in five Algebra classroom studies with more than 1,000 middle school students.

Throughout these classroom studies, we have addressed a number of research questions such as questions about the effect of answering tutee's questions [11] and the effect of extrinsic motivation for tutor learning [12].

One of the most important findings thus far is that learning by teaching may not be effective when students do not have sufficient prior knowledge on the task (how to solve equations in our case) and do not know how to teach properly [8]. In previous studies, we often observed that students taught their synthetic peers incorrectly without realizing they were making mistakes. Students also often taught their peers inappropriately—e.g., only teaching “easy” problems, causing the synthetic peer to fail to develop sufficient skills to solve a wide range of equations.

## 3 Research Question and Hypothesis

Our previous studies strongly suggest that students need an assistance for successful teaching in order to facilitate tutor learning. We then hypothesize that providing adaptive scaffolding will resolve this issue. What kinds of scaffolding should be provided? From the past studies, we have two working hypotheses.

First, students need to correctly teach their peers how to solve problems. However, due to the lack of sufficient prior knowledge, students often make mistakes and get stuck. Adaptive scaffolding on how to solve problems is therefore necessary—we call this the *cognitive scaffolding*. We hypothesize that adaptive cognitive scaffolding will be particularly important for students with low prior knowledge since some level of

knowledge is necessary for students to have in order to be able to teach their synthetic peers—the *cognitive scaffolding hypothesis*.

Second, students need to know how to teach their peers appropriately. Students need to know, for example, what problem might be useful to teach next and when to quiz their peers. Adaptive scaffolding on how to teach is therefore necessary—we call this the *metacognitive scaffolding*. Even with a low prior knowledge, students might recognize their mistakes and acquire correct skills on how to solve problems while teaching *if* appropriate feedback is given from the tutoring interaction—e.g., the summary of a formative assessment reveals an inconsistency between the student's belief and actual correctness. However, to receive effective feedback, students must teach their peers properly. Therefore, adaptive metacognitive scaffolding is essential—the *metacognitive scaffolding hypothesis*.

Third, it might be the case that students need both cognitive and metacognitive scaffolding for successful learning by teaching—the *hybrid scaffolding hypothesis*.

Our research question centers on which of these types of adaptive scaffolding facilitate tutor learning. To test our hypotheses, we implemented the cognitive and metacognitive scaffolding on APLUS (Sect. 4) and conducted a classroom study with the extended APLUS (Sect. 5).

## 4 Technology Innovation for Learning by Teaching

We have developed an online environment called **APLUS** (Artificial Peer Learning environment Using SimStudent) where students learn to solve algebra equations by teaching a synthetic peer, SimStudent.

SimStudent is a machine learning agent that interactively learns cognitive skills in the form of production rules through guided-problem solving [13]. SimStudent is an implementation of programming by demonstration in the form of inductive logic programming. This is made possible by generalizing examples that show when to apply particular skills. In the context of learning by teaching, feedback and hints on a step provided by the student to SimStudent become examples.

Figure 1 shows an example screenshot of APLUS. Details of APLUS have been published elsewhere (for example [8]); hence we only provide a brief explanation here. SimStudent is shown as an avatar on the bottom left corner. To teach SimStudent how to solve equations, a student must enter a problem into the *tutoring interface* ('a' in Fig. 1). SimStudent will then attempt to solve the problem one step at a time by applying the skills learned so far. On each step, if SimStudent can make a suggestion, the student is prompted to provide *yes/no* feedback about the correctness of the suggested step ('b' in Fig. 1). Positive feedback ("yes") indicates that the student agrees that the step SimStudent suggested is correct, in which case SimStudent proceeds to the next step. Negative feedback ("no") indicates the student's disagreement. When given negative feedback, SimStudent attempts to apply another skill and make another suggestion. If SimStudent cannot find a skill to apply, SimStudent asks the student to demonstrate the next step, and the student then performs the actual step on the tutoring interface.

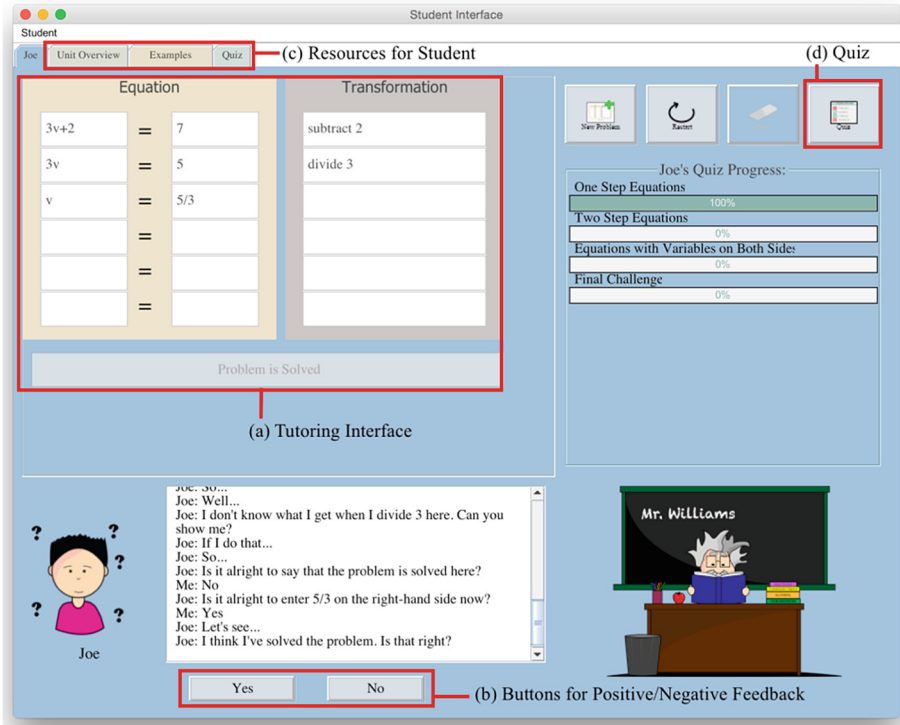
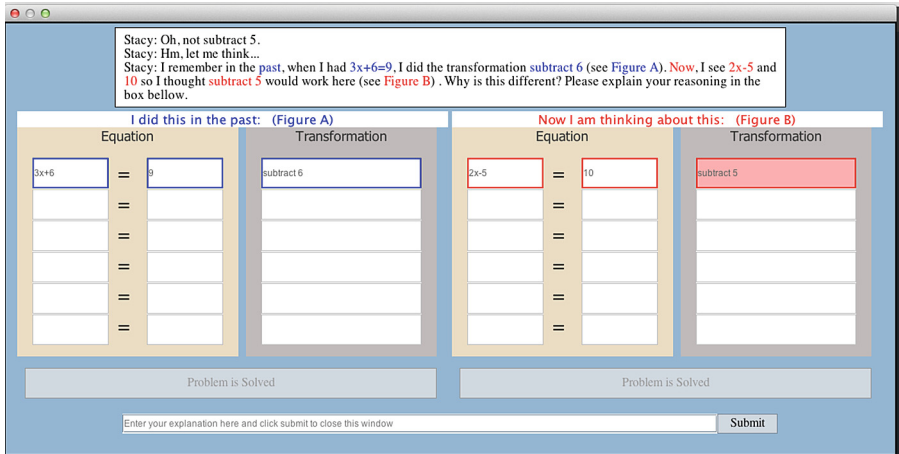


Fig. 1 An example screenshot of APLUS.

When a student provides negative feedback to a step SimStudent suggested, SimStudent occasionally asks the student to explain why he/she thinks that the step is wrong [11]. Figure 2 shows an example screenshot of SimStudent asking a “why” question. SimStudent can also compare (1) a previous step in which the same skill was applied and received positive feedback and (2) the current step which just received negative feedback. SimStudent then asks the student why the current step is incorrect. To proceed with the tutoring process, the student must answer SimStudent’s question in a free-form text.

Students’ goal is to have their SimStudent pass the quiz (‘c’ in Fig. 1). When SimStudent is quizzed by a student, it attempts to solve quiz problems in a target section by applying already learned skills. A teacher agent called Mr. Williams, as shown on the lower right corner in Fig. 1, then provides a summary of the quiz results, and the student can review the exact solutions made by SimStudent one by one. APLUS also includes resources for students to review (‘d’ in Fig. 1) and prepare for tutoring. The resources include worked-out examples with brief explanations on solutions and a unit overview that provides a quick introduction to equation solving.

We have recently modified Mr. Williams to provide adaptive scaffolding in two ways: (1) *Cognitive scaffolding* provides adaptive assistance on how to solve equations. When a student is not sure about the correctness of a step SimStudent performed, or



**Fig. 2** SimStudent is asking student to explain why the step is wrong.

when they do not know what a correct next step is, he/she can click on Mr. Williams to ask for help. Mr. Williams then provide a just-in-time, contextualized assistance to overcome the student's impasse. (2) *Metacognitive scaffolding* provides adaptive assistance on how to teach SimStudent. When a student is not sure about how to proceed tutoring, he/she can click on Mr. Williams to ask for help. Four types of assistance are provided: (a) the quiz assistance suggests when students should take the quiz and explains why, (b) the problem selection assistance suggests what problem students should pose next and explains why, (c) the resource assistance suggests when students should review a particular resource and why, and (d) the impasse recovery assistance suggests students should restart a problem or give a new problem when they are stuck for a long-enough time.

## 5 Evaluation Study

To test the hypotheses on the effectiveness of adaptive scaffolding as mentioned in the previous section, we conducted a classroom in-vivo study in Algebra classes at three urban public middle schools in the greater Pittsburgh area in Pennsylvania.

### 5.1 Method

The study was a randomized controlled trial with three conditions. (1) The *metacognitive condition* (MC for short) used a version of APLUS with metacognitive scaffolding only. (2) The *cognitive condition* (C for short) used APLUS with cognitive scaffolding only. (3) The *hybrid condition* (MC + C for short) used APLUS with both metacognitive and cognitive scaffolding.

In total, 364 students (7th and 8th grade) participated in the study from 22 algebra classes. Students were randomly assigned to one of the three conditions.

The study lasted for six consecutive days with one classroom period (42 min each) per day. On the first day, all students took an online pre-test. On the second day, students first watched a 6-min introduction video on how to use APLUS, and then started tutoring SimStudent. Students used APLUS for four days. On the sixth day, students took an online post-test.

## 5.2 Measures

Students' learning outcome was measured with the online test scores. The online test consisted of two parts: a Procedural Skill Test and a Conceptual Knowledge Test.

The *Procedural Skill Test* has three sections: (a) an equation section that contains 10 problems; 2 one-step equations, 2 two-step equations and 6 equations with variables on both sides; (b) an effective next step section that has 2 equation problems, each showing an intermediate solution step with four candidates for a next step and asking students to indicate if each candidate is correct or not; and (c) an error detection section with 3 equation problems, each showing an incorrect solution for which students are asked to identify the incorrect step and explain their reasoning.

The *Conceptual Knowledge Test* consists of 24 true/false multiple choice questions, with 7 items asking about variable terms, 6 asking about constant terms, 6 asking about like terms, and 5 asking about equivalent terms.

In addition to the learning outcome data, we also used the process data that are detailed interactions between student and system that APLUS automatically logs (e.g., problems used for tutoring, tutored steps, quiz frequency, etc.).

## 5.3 Results

For the following analysis, we included only those students who took both the pre- and post-tests, and “completed” teaching, which we define as either the student participated in all four days of teaching SimStudent or had SimStudent pass all quiz sections. As a result, 257 students are included in the analysis below—89 in the C condition, 88 in the MC condition, and 80 in the MC + C condition.

### 5.3.1 Test Scores

Table 1 shows scores for Conceptual Knowledge Test (CKT) and Procedural Skill Test (PST). There is no condition difference (C vs. MC vs. MC + C) in pre-test score both for CKT ( $F(2, 254) = 0.17, p = 0.85$ ) and PST ( $F(2, 254) = 1.21, p = 0.30$ ). We then ran a repeated-measures ANOVA, for CKT and PST separately, with test scores as a dependent variable, and test-time (pre vs. post) and condition (C vs. MC vs. MC + C) as independent variables.

For the conceptual test (CKT), there is a main effect of test-time ( $M_{Pre} = 0.41 \pm 0.24$  vs.  $M_{Post} = 0.48 \pm 0.20$ ;  $F(1, 254) = 22.52, p < 0.001, d = 0.28$ ), but there is no main effect of condition ( $M_C = 0.44 \pm 0.22$  vs.  $M_{MC} = 0.45 \pm 0.23$  vs.  $M_{MC+C} = 0.46 \pm 0.22$ ;  $F(2, 254) = 0.26, p = 0.77$ ). *The current version of APLUS enhanced students' understanding of algebra concepts measured in CKT. The type of adaptive*

**Table 1.** Test scores both for CKT and PST.

	CKT		PST	
	Pre-test	Post-test	Pre-test	Post-test
<b>MC + C</b>	.42 (.24)	.48 (.20)	.54 (.22)	.63 (.24)
<b>MC</b>	.40 (.24)	.50 (.21)	.53 (.23)	.62 (.23)
<b>C</b>	.41 (.24)	.46 (.20)	.49 (.21)	.54 (.23)

*scaffolding does not have any impact on students' learning on conceptual knowledge measures in CKT.*

For the procedural test (PST), the repeated-measures ANOVA suggested the existence of an interaction between test-time and condition;  $F(2, 254) = 2.85, p = 0.06$ . A simple main effect on condition (paired t-test with test-time as the independent variable) revealed that students in all conditions showed a reliable increase in PST test scores, but the effect size is notably smaller in C condition; C: *paired-t*(88) =  $-2.55, p = 0.01, d = \mathbf{0.20}$ ; MC: *paired-t*(87) =  $-5.07, p < 0.001, d = \mathbf{0.41}$ ; MC + C: *paired-t*(79) =  $-5.54, p < 0.001, d = \mathbf{0.40}$ . A simple main effect analysis on PST post-test (an ANOVA with condition as the independent variable) revealed condition as a main effect;  $F(2, 253) = 3.95, p < 0.05$ . The post-hoc tests confirmed that both MC and MC + C students scored reliably higher on the PST post-test than C students ( $t(175.0) = 2.41, p < 0.05$  for MC and  $t(163.5) = 2.64, p < 0.01$  for MC + C), but there is no reliable difference between MC and MC + C students;  $t(161.9) = -0.37, p = 0.71$ .

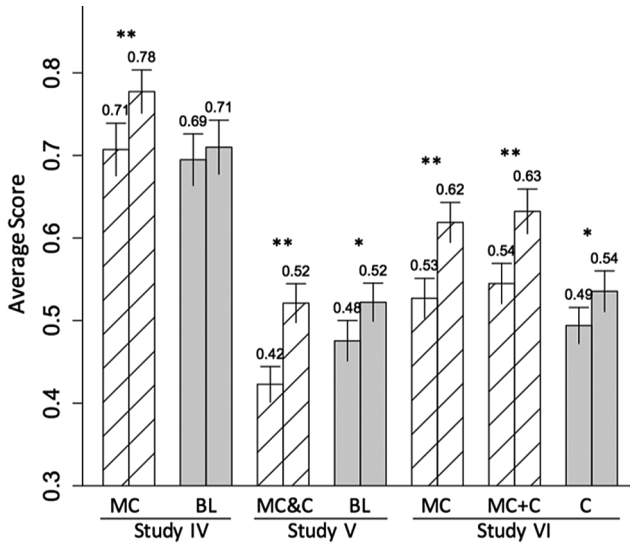
In sum, *metacognitive scaffolding is helpful but cognitive scaffolding does not appear to amplify the effect of tutor learning.* In other words, *adding cognitive scaffolding to metacognitive scaffolding does not yield better effects than metacognitive scaffolding alone.*

Since the current study does not include a baseline condition where no scaffolding is available, we compared the current study with two previous studies we conducted: Study IV [14] where metacognitive scaffolding (MC) was compared with no scaffolding (BL), and Study V where metacognitive plus cognitive scaffolding (MC + C) was compared with no scaffolding (BL). In Study IV there were 173 students (7th through 9th grade) in nine Algebra-I classes, whereas in Study V there were 318 students (7th and 8th grades) in 14 Pre-Algebra and 3 Algebra I classes from two schools.

Figure 3 shows a bar graph of PST pre- and post-test scores. In the graph, the current study is denoted as Study VI. The relative high test scores in Study IV is arguably due to the population difference (the only study with 9th graders).

We tested two hypotheses: (C1) all four conditions with metacognitive scaffolding, regardless of the availability of cognitive scaffolding, show the same gain from pre- to post-tests; and (C2) the C condition in Study VI shows the same gain from pre- to post-test as two BL conditions in Study IV and V.

To test these two hypotheses, we ran two repeated-measures ANOVAs, one for each hypothesis. For (C1), there is a main effect of test-time;  $F(1, 319) = 100.42,$



**Fig. 3** PST scores for 7 conditions in Study IV, V, and the current study (Study VI). \*\*  $p < 0.01$ , \*  $p < 0.05$

$p < 0.001$ . There is also a main effect of condition;  $F(3, 319) = 19.29, p < 0.001$ . There is no statistically reliable interaction between test-time and condition. The same pattern is found for (C2): a main effect of test-time ( $F(1, 245) = 11.77, p < 0.001$ ); condition ( $F(2, 245) = 18.42, p < 0.001$ ); and no interactions between them.

The data collected from three independent classroom studies all suggest that *metacognitive scaffolding facilitates tutor learning, regardless of the availability of cognitive scaffolding. However, cognitive scaffolding does not necessarily facilitate tutor learning (or is equally “effective” as the baseline), and adding cognitive scaffolding to metacognitive scaffolding is as effective as the metacognitive scaffolding alone.*

### 5.3.2 Effect of Cognitive Scaffolding and Students’ Prior Knowledge

To see if cognitive scaffolding helped certain students, we categorized students in the cognitive scaffolding condition into four groups based on their prior knowledge as measured in the pre-test score. Table 2 shows procedural pre- and post-test (PST) scores. The quartile Q1 represents the students who scored lowest on the pre-test.

**Table 2.** Procedural pre- and post-test (PST) scores for students in the cognitive scaffolding condition. Students are grouped based on their pre-test quartile.

	Q1	Q2	Q3	Q4
<b>Pre</b>	0.37(0.16)	0.40(0.13)	0.54(0.17)	0.66(0.21)
<b>Post</b>	0.44(0.16)	0.47(0.24)	0.53(0.25)	0.70(0.18)

A repeated-measures ANOVA with test score as a dependent variable and test-time (pre vs. post) and quartile ( $Q1 \sim Q4$ ) as independent variables revealed a main effect for quartile;  $F(3, 85) = 11.77, p < 0.001$ . Test-time is also a main effect;  $F(1, 85) = 6.57, p = 0.01$ . There is no statistically reliable interaction between test-time and quartile. This result suggests that *the “effect” of cognitive scaffolding does not change by students’ prior knowledge.*

### 5.3.3 Why Is the Metacognitive Scaffolding Effective?

We have yet to fully understand what makes metacognitive scaffolding effective. So far, we found the following. First, the effectiveness of metacognitive scaffolding does not change based on the student’s prior knowledge (measured as pre-test score). This is confirmed by dividing students into quartiles based on their PST pre-test scores.

Second, in a previous study, the data suggested that metacognitive scaffolding on problem selection (i.e., what problem should be taught next) actually influenced students to pose more appropriate problems to SimStudent, which in turn facilitated tutor learning [14]. However, the effect of metacognitive scaffolding on problem selection is not confirmed in the current study.

Third, there is no notable correlation between the number of metacognitive hints received and PST post-test scores when pre-test score is controlled. A stepwise regression revealed that the number of quiz hints received is a statistically reliable predictor of post-test score ( $F(1,160) = 6.19, p = 0.01$ ). However, students quizzed SimStudent an average of nine times in all three conditions.

## 6 Discussion

The metacognitive hypothesis has been supported. The current and past two classroom studies all show that metacognitive scaffolding (helping students to teach and prepare for teaching) is an essential component for successful learning by teaching, whereas cognitive scaffolding (helping students to solve equations) has no effect relative to no scaffolding. In the current implementation, the metacognitive scaffolding is operationalized to support students’ understanding of how to select appropriate problems to teach, when to quiz their peers, and when to review study materials to prepare themselves for teaching (e.g., reviewing worked-out examples and unit overview).

We have yet to fully understand why metacognitive scaffolding, as we defined it in this paper, helps. One hypothesis is that understanding (and actually applying) proper teaching strategies increases the likelihood for students to be exposed to opportunities to learn correct skills (e.g., from worked-out examples) and also to face the knowledge gap (e.g., a step that a student believes to be correct is marked as incorrect on the quiz summary). If these ideas are actually the key events that drive tutor learning, then guiding students to these key events should facilitate students’ learning.

Further research is necessary to understand the underlying mechanism of tutor learning. We are currently analyzing the process data showing detailed interaction between students and SimStudent. Sequence mining is one potential technique to address the question of why metacognitive scaffolding helps.



We were surprised that the current data do not provide evidence that cognitive scaffolding, as we defined it in this paper, helps tutor learning. It might be the case, however, that the current implementation of cognitive scaffolding needs to be improved. Students might have used the cognitive scaffolding as a mere mechanism to provide correct feedback and hint—similar to those students excessively asking for hints when using cognitive tutors just to perform a step correctly—that must be discouraged. A future study will be designed to explore this new hypothesis.

## 7 Conclusion

We found that the adaptive scaffolding on how to tutor and how to prepare for tutoring (the metacognitive scaffolding) facilitates tutor learning, while the adaptive scaffolding on how to solve problems (the cognitive scaffolding) has virtually no impact on tutor learning. In the present study, the metacognitive scaffolding provided just-in-time assistance on what problem should be taught next, when to quiz (i.e., a formative assessment), when to review resources to prepare for tutoring, and when to recover from an impasse. The cognitive scaffolding provided assistance on the correctness of the steps performed by the peer (to provide feedback to the peer), and the next step to be performed (to provide hints to the peer on what to do next).

In our classroom studies, we often see students get excited about interactively teaching on a computer with actual dialogue with a synthetic agent. Our data from recent classroom studies consistently show evidence of the effect of learning by teaching. Understanding how and why metacognitive scaffolding helps but not cognitive scaffolding is therefore an important research agenda to further advance the theory of learning by teaching and to build an effective technology for learning by teaching.

**Acknowledgement.** The research reported here was supported by National Science Foundation Award No. DRL-1252440

## References

1. Gartner, A., Kohler, M., Riessman, F.: *Children Teach Children: Learning by Teaching*. Harper & Row, New York (1971)
2. Cohen, P.A., Kulik, J.A., Kulik, C.L.C.: Education outcomes of tutoring: a meta-analysis of findings. *Am. Educ. Res. J.* **19**(2), 237–248 (1982)
3. Robinson, D., Schofield, J., Steers-Wentzell, K.: Peer and cross-age tutoring in math: outcomes and their design implications. *Educ. Psychol. Rev.* **17**(4), 327–362 (2005)
4. Cohen, E.G.: Restructuring the classroom: conditions for productive small groups. *Rev. Educ. Res.* **64**(1), 1–35 (1994)
5. Roscoe, R.D., Chi, M.T.H.: Understanding tutor learning: knowledge-building and knowledge-telling in peer tutors' explanations and questions. *Rev. Educ. Res.* **77**(4), 534–574 (2007)

6. Biswas, G., et al.: Learning by teaching: a new agent paradigm for educational software. *J. Appl. Artif. Intell.* **19**(3&4), 363–392 (2005)
7. Bredeweg, B., et al.: DynaLearn - engaging and informed tools for learning conceptual system knowledge, in cognitive and metacognitive educational systems (MCES 2009). In: Pirrone, R., Azevedo, R., Biswas, G. (eds.) *AAAI Fall Symposium*. AAAI Press, Arlington, pp. 46–51 (2010)
8. Matsuda, N., et al.: Cognitive anatomy of tutor learning: lessons learned with SimStudent. *J. Educ. Psychol.* **105**(4), 1152–1163 (2013)
9. Biswas, G., et al.: Measuring self-regulated learning skills through social interactions in a teachable agent environment. *Research and Practice in Technology Enhanced Learning*, pp. 123–152 (2010)
10. Okita, S.Y., Bailenson, J., Schwartz, D.L.: The mere belief of social interaction improves learning. In: McNamara, D.S., Trafton, J.G. (eds.) *The Proceedings of the 29th Meeting of the Cognitive Science Society*, Nashville, pp. 1355–1360 (2007)
11. Matsuda, N., et al.: Studying the effect of tutor learning using a teachable agent that asks the student tutor for explanations. In: Sugimoto, M., et al. (eds.) *Proceedings of the International Conference on Digital Game and Intelligent Toy Enhanced Learning (DIGITEL 2012)*. IEEE Computer Society, Los Alamitos, pp. 25–32 (2012)
12. Matsuda, N., Yarzebinski, E., Keiser, V., Raizada, R., Stylianides, G., Koedinger, K.R.: Motivational factors for learning by teaching. In: Cerri, S.A., Clancey, W.J., Papadourakis, G., Panourgia, K. (eds.) *ITS 2012. LNCS*, vol. 7315, pp. 101–111. Springer, Heidelberg (2012)
13. Matsuda, N., Cohen, W.W., Koedinger, K.R.: Teaching the teacher: tutoring simstudent leads to more effective cognitive tutor authoring. *Int. J. Artif. Intell. Educ.* **25**, 1–34 (2015)
14. Matsuda, N., Griger, C.L., Barbalios, N., Stylianides, G.J., Cohen, W.W., Koedinger, K.R.: Investigating the effect of meta-cognitive scaffolding for learning by teaching. In: Trausan-Matu, S., Boyer, K.E., Crosby, M., Panourgia, K. (eds.) *ITS 2014. LNCS*, vol. 8474, pp. 104–113. Springer, Heidelberg (2014)

# Scale-Driven Automatic Hint Generation for Coding Style

Rohan Roy Choudhury<sup>(✉)</sup>, Hezheng Yin, and Armando Fox

University of California, Berkeley, USA  
{rrc,hezheng.yin,fox}@berkeley.edu

**Abstract.** While the use of autograders for code correctness is widespread, less effort has focused on automating feedback for good programming style: the tasteful use of language features and idioms to produce code that is not only correct, but also concise, elegant, and revealing of design intent. We present a system that can provide real-time actionable code style feedback to students in large introductory computer science classes. We demonstrate that in a randomized controlled trial, 70% of students using our system achieved the best style solution to a coding problem in less than an hour, while only 13% of students in the control group achieved the same. Students using our system also showed a statistically-significant greater improvement in code style than students in the control group.

**Keywords:** Coding style · Autograding · Automatic hint generation · MOOCs

## 1 Motivation and Overview

Rapid feedback is integral to mastery learning. Prior work has shown that students learn best through the process of repeatedly submitting, receiving immediate actionable feedback and resubmitting [1, 9, 13]. Automatic graders (autograders) provide this capability and are thus used extensively in programming courses, especially Massive Open Online Courses (MOOCs). However, while the use and development of autograders for code correctness is widespread, less effort has focused on automating feedback for good programming style [17].

Software with poor code quality has been shown to require significantly higher maintenance, a sobering fact considering that maintenance dominates software cost [5]; good coding style therefore has significant implications for the software industry. By providing students with rapid and actionable style feedback, intelligent tutoring systems can help future software developers develop good coding style habits early.

Most existing code style tools check code against a fixed set of style rules that do not depend on the specific code being analyzed. Checkers such as `lint(1)` and `pylint` and existing autograders such as `rag` [6] are unable to account for subtleties such as whether using a different data structure, language construct or

library call might be stylistically better, and therefore cannot provide actionable feedback on how to improve style [6, 11]. As a result, providing actionable style feedback usually requires instructors to manually read student code, which can be resource-prohibitive in large courses. Our university’s rigorous introductory computer science course relies on over 40 teaching assistants to manually grade over a thousand code submissions per assignment. Given scarce TA resources, style is lightly graded on a coarse-grained scale based on a “style guide” given to students. Automating style grading would save significant instructor time and could provide more tailored feedback to support mastery learning.

Our approach to providing such guidance automatically is to (1) identify similarities among student code submissions for a short assignment (a few lines to tens of lines of code), (2) analyze these similarities using clustering techniques and Abstract Syntax Tree (AST) comparison, and (3) use them to deliver a combination of instructor-authored guidance and auto-generated syntactic hints, such that the guidance provided on a given submission is based on properties of another student’s structurally similar but stylistically superior submission.

Specifically, we make the following contributions:

1. Two techniques for analyzing similarities in student code for short assignments: one based on unsupervised classification and the other based on differencing of the ASTs of student submissions.
2. A workflow based on the above techniques that enables instructors to efficiently provide style feedback for a large body of submissions to the same assignment, with effort proportional to the number of distinct approaches to solving the problem, not the number of students.
3. An unsupervised, automated, student-facing workflow that provides students with a combination of instructor-authored guidance and automatically-generated guidance based on similar submissions by other students.
4. A randomized controlled trial experiment demonstrating the efficacy of our system. Students in the treatment group showed a statistically-significantly greater improvement in style than students in the control group.

## 2 Related Work

Most work on hint generation has focused on code correctness. Lazar and Bratko [14] construct hints for Prolog programs in a generative manner based on specific editing operations that transform the program code. Rivers and Koedinger [19] propose a method for automatic code correctness feedback by using AST differencing to identify a student’s state in a solution space and showing the student another student’s slightly-better program as feedback, developing various techniques to reduce the vast solution space and make the hint-generation problem tractable. In contrast, we assume students start with a correct but possibly ugly solution, which they may have produced on their own or with the help of such a system and/or verified against a test-based autograder [6].

Whereas early work on providing automated feedback was based on (often manually-constructed) “bug libraries,” as large corpora of code have become

available (due the increasing class sizes and the availability of cloud services such as GitHub), guidance systems have begun generating feedback by comparing student code to an existing corpus. Codex [2] discovers common language idioms (integral to good style) and detects patterns in the student’s code that might benefit from applying them. Codewebs [18] tries to identify semantically-equivalent code blocks in different students’ submissions, to which the same instructor feedback can be applied. Both approaches use abstract syntax tree (AST) differencing to compare code exemplars. We use similar techniques to identify correct student submissions that are similar but have salient stylistic differences, and use these submissions to generate style feedback.

We also draw upon recent work on using machine learning techniques to increase instructor leverage. Huang et al. [10] found that clustering ASTs of student submissions produces clusters that embody similar strategies to solving the problem and could potentially receive the same feedback. Glassman et al. [8] hierarchically cluster student submissions, based first on student strategy and then on implementation. They identify the features required for effective clustering. We draw upon their work to cluster existing student submissions to allow instructors to provide predetermined style feedback for students solving the problem using a particular strategy.

### 3 Approach

We and others have observed that given a large enough corpus of submissions to a given programming problem, there exists a range of stylistic mastery, from naïve to expert [17]. Figure 1 shows three correct submissions from students with pseudonyms Alice, Bob, and Charlie, who provide three correct solutions to the same simple problem: given a list of words, return a list of groups such that all words in each group are anagrams of each other. As the figure shows, correct solutions vary in length (and therefore complexity) by nearly a factor of ten. While we could simply show Alice’s solution to Charlie, many conceptual gaps separate her concise solution from his 30-line solution. In contrast, guiding students to incrementally improve and discover the best solution has been shown to be more conducive to mastery learning by reducing cognitive load, especially for struggling students [20]. Thus, we seek a sequence of hints that will guide Charlie to incrementally transform his solution to one like Alice’s.

In order to provide style-improvement feedback based on differences between student submissions, we need a way to measure both style goodness and differences. The software engineering literature suggests a variety of metrics of stylistic quality [12]. We have found empirically that the ABC score, which tallies a weighted count of assignments, branches, and conditional statements in a block of code [3], is a good proxy for stylistic quality when used on short (a few lines to a few tens of lines) code fragments. It relies on static analysis only, and is easy to implement and fast to compute. In general, a lower score is better, but it is an ordinal metric, i.e. cutting the ABC score by half does not necessarily imply that the code has doubled in stylistic quality. That said, the choice of

<pre>def combine_anagrams(words) #Alice   words.group_by{ w  w.chars.downcase.sort}.values end</pre>	<pre>def combine_anagrams(words) #Charlie   rtn = Array.new   words.each do  word      p(word)     wordDowncase = word.downcase     letters = wordDowncase.split("")     exist = false     rtn.each do  rtnAry        rl = rtnAry[0].downcase.split("")       if (rl.length==letters.length) then         p(rl)         rl.sort!         letters.sort!         match = true         i = 0         rl.each do  rli            p(((rli + "_") + letters[i]))           match=false if (rli!=letters[i])           i = (i + 1)         end         if (match == true) then           (rtnAry &lt;&lt; word)           exist = true         end       end     end     rtn &lt;&lt; [word] if (not exist)   end   return rtn end</pre>
<pre>def combine_anagrams(words) #Bob   dict = {}   words.each do  word      letters = word.downcase.each_char.sort     if dict.has_key?(letters) then       dict[letters] += [word]     else       dict[letters] = [word]     end   end   return dict.values end</pre>	

**Fig. 1.** A 3-line correct solution by Alice, 12-line correct solution by Bob, and 30 line correct solution by Charlie to the same problem, illustrating the range of stylistic mastery commonly found in the type of assignments used in introductory classes.

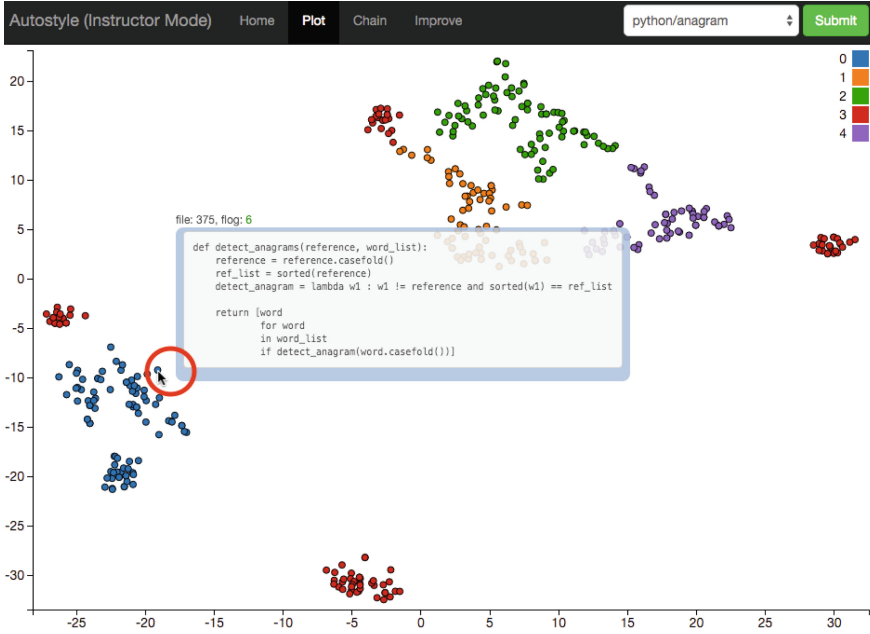
algorithm used to compute the quality score is an input to our workflow, and any metric that obeys the triangle inequality can be used.

The edit distance between the abstract syntax trees (ASTs) is a common measure of similarity between two code fragments [22]. To emphasize the importance of higher-level structure (the “problem solving strategy”), we use the *normalized* tree edit distance (n-TED) of the AST, which weights nodes closer to the root of the AST more heavily, thus preventing minor syntactic differences at the leaves from affecting the similarity score of programs that are structurally similar, but differ in low-level details [21].

## 4 Instructor and Student Workflow

Our workflow starts with a corpus of existing submissions to a programming problem, which may include an instructor-authored canonical solution. This corpus may consist of submissions from a previous offering of the course, or it can be bootstrapped using submissions from a subset of the students in a large-enrollment course. We perform an offline computation to generate the AST and quality score for every submission, and the pairwise similarity between all pairs of submissions. The submission(s) with the best style score(s) are judged to be the best possible style exemplars for this problem. The result of this step is an undirected weighted complete graph in which each student submission is a vertex and the tree edit distance between submissions are the weights on the edges.

We then cluster the student submissions to aggregate groups of submissions that use the same problem-solving strategy. We observed that stylistically-better solutions tend to be densely clustered, whereas stylistically weak solutions tend to form sparse clusters (informally, there are many more varied distinct ways



**Fig. 2.** t-SNE [15] 2D visualization of clustering 425 submissions. Each dot represents a submission, colors represent clusters, and hovering over a dot shows the actual code associated with that submission.

to be stylistically “wrong” but only a few ways to be stylistically “right” for a short assignment). We therefore use the OPTICS density-based clustering algorithm [21].

The instructor then annotates each cluster with three items. The first is a label: **good**, **average**, or **weak**. A **good** cluster has solutions close to or identical to the best solution. **Average** clusters contain solutions that solve the problem using a mundane approach and can thus still improve on both approach and language idioms. **Weak** clusters contain solutions that generally exhibit lack of knowledge of one or more important language concepts or constructs that are essential to solving the problem with excellent style. There is clearly instructor subjectivity in applying these labels; to aid the instructor, we display an interactive 2D visualization, as Fig. 2 shows.

The second item is an *approach hint* for the cluster. Approach hints aim to correct a misunderstanding or lack of awareness of the best way to approach the problem; they illustrate the high-level reasoning of how to approach the problem from a new direction while still leaving the work of developing and implementing a more elegant solution to student. That is, this is the hint that the instructor would give a student whose submission was similar to the cluster members.

The third item is an *exemplar* the instructor chooses from another cluster that she believes to represent a better approach. In keeping with our philosophy of incremental improvement, we ask the instructor not to simply select an exemplar from the “best” cluster as part of the approach hints.



**Fig. 3.** Example of a chain and the hints generated for such a chain.

In addition to the instructor-authored annotations on each cluster, our system automatically produces two other types of guidance. *Code Skeletons* are redacted versions of other students’ solutions that demonstrate the key control flows and structure of a possible solution, while obfuscating variable names and function call names. *Syntactic hints* guide the student to add (remove) specific structures (loops, conditionals, special language constructs, calls to common built-in or library functions) in order to improve style, based on the presence (absence) of those features in submissions with better style. Syntactic hints are derived by *chain-building* [17], a process that traverses the complete graph generated in the preparation step to find a path from a given submission to one of the “best possible” submissions. The path is subject to the constraints that for each edge  $A \rightarrow B$ , the n-TED structural difference between  $A$  and  $B$  does not exceed a set threshold, and  $B$ ’s style score is better than  $A$ ’s by a set threshold. The path is analyzed to determine the most important syntactic hints corresponding to structural features present (absent) in later links in the chain, as shown in Fig. 3. The feature vectors used in this analysis check for specific language features such as built-in functions, language idioms, and basic control flow constructs in each language; we have constructed feature vectors for Ruby, Java, and Python.

## 5 Experiment Design and Setup

We performed an intervention experiment using  $n = 80$  compensated student participants and compensated teaching assistant participants to evaluate the efficacy of our system under realistic conditions.<sup>1</sup> All recruited participants were associated with our university’s large-enrollment introductory computer science course, which introduces a range of programming concepts using the Python language. Participants were recruited by advertising in the course discussion forum and were paid US\$15 for one hour of their time.

<sup>1</sup> IRB Protocol number: 2015-10-8003.



The primary hypothesis is as follows: Compared with students who are given only a set of “good style” guidelines, *students receiving hints via our automated workflow will improve their code quality more in a given period of time.*

We had a corpus of 265 student submissions of this assignment from a previous offering of the course. Prior to working with the study participants, we ran our clustering algorithm on this corpus and labeled each generated cluster as **good**, **average**, or **weak**; we annotated **average** and **weak** clusters with *approach hints*, and picked *exemplars* for the **weak** clusters. To help validate that the clusters do indeed capture common approaches, we recruited two TAs from the same course and asked each to write down in their own words a description of the overall approach represented by each cluster’s members, and two additional TAs to judge whether the descriptions provided by the first two TAs were similar on a five-point scale. We report a square weighted Cohen’s kappa of 0.71 and an average similarity rating of 3.85 ( $\sigma = 0.91$ ). These statistics indicate that different instructors are able to recognize the approaches captured by the clusters.

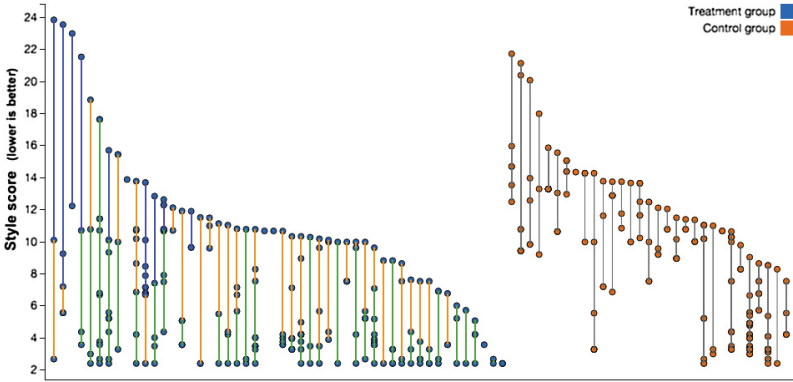
The recruited students were randomly placed into either the treatment group (50 students) or control group (30 students). Both groups were given the same Python programming assignment, based on a previous offering of the course but absent from the current offering. All participants were provided with the “style guide” authored by the course staff and were allowed access to the Internet to look up documentation. All participants were shown the same problem and instructed to submit a solution; participants were allowed as much time as they wanted (within the one-hour time limit of the experiment) to do so. Upon submission, participant solutions were automatically evaluated against a set of test cases for correctness. Upon submitting a correct solution, the participant was immediately shown the computed “style score” for their solution as well as the best possible style score for this problem (2.41 based on the corpus of previous submissions—recall that lower ABC scores are better), and asked to revise their submission to work towards the best score. The control group was given only the style guide (reflecting current practice in the course), whereas the treatment group received specific automatically-generated feedback from our system.

In particular, each submission from a treatment-group student was first analyzed using  $k$ -nearest neighbors to determine which cluster it would belong to. If it belonged to a **good** cluster, the participant was shown only a syntactic hint based on building a chain from his submission to the best submission. If it belonged to an **average** cluster, the participant was shown the instructor’s approach hint for that cluster, *and* a syntactic hint. If it belonged to a **weak** cluster, the participant was shown the instructor’s approach hint for that cluster, *and* the **code skeleton** of the instructor-chosen exemplar for that cluster. Code skeletons are automatically constructed using a regular expression that redacts variables and function call names while retaining control flow structures.

All participants were asked to repeatedly revise their solution based on feedback until they achieved the best possible quality score or exceeded one hour.

## 6 Results

We collected every correct submission made during the experiment for both groups. Figure 4 shows each student’s submission history and the type of feedback they received. There was no significant difference in the style score of the initial submission between the two groups ( $p = 0.21$ , Pearson’s  $\chi^2$  test). However, students in the treatment group ended with significantly better style scores ( $p = 0.007$ , Kruskal-Wallis  $H$  test), indicated in the graph by the treatment group vertical lines ending much lower than the control group ones (lower style scores are better with the ABC metric we used).



**Fig. 4.** Each vertical line represents a student and each dot along the line is a submission. The color of line segments between dots for the treatment group codifies the combination of hints the student received—blue: **approach + code skeleton**, yellow: **approach + syntactic**, green: **syntactic only**. (Color figure online)

Figures 4 and 5 show that the percentage of students that achieved the best style solution (style score of 2.41) is considerably greater in the treatment group than in the control group. Moreover, as shown in Fig. 5, students in the treatment group improved significantly more than those in the control group over the one hour experiment period. They also showed significantly more improvement per submission attempt than control group.

Metric	Treatment	Control	Statistically significant?
% of students achieving best solution	<b>70%</b>	13%	Yes ( $p < 0.001$ ) <sup>†</sup>
Mean improvement in style score	<b>7.1 ± 4.9</b>	4.1 ± 3.1	Yes ( $p = 0.007$ ) <sup>‡</sup>
Mean improvement per attempt	<b>1.8 ± 3.12</b>	0.62 ± 1.9	Yes ( $p < 0.001$ ) <sup>‡</sup>

**Fig. 5.** Key results. <sup>†</sup>Fisher’s exact test <sup>‡</sup>Kruskal-Wallis  $H$  test

To evaluate the effectiveness of the different types of guidance, we asked students to rate the helpfulness of different types of hints on a scale of 1 (not at all helpful) to 4 (very helpful) immediately after completing the study. We find that when students were given different types of hints, neither type of hint was perceived to be significantly more helpful than the others. Specifically, students reported a mean perceived helpfulness of  $3.13 \pm 0.79$  for syntactic hints ( $S$ ),  $2.77 \pm 0.89$  for approach hints ( $A$ ), and  $2.85 \pm 0.82$  for code skeletons ( $C$ ). We also studied the ratings distribution for the subset of students who received some combination of hints ( $A + S$  or  $A + C$ ); at a 5% significance level ( $t$ -test), we found no evidence of significant difference between the perceived helpfulness of different types of hints in either group ( $p = 0.092$  for  $A + S$ ,  $p = 0.760$  for  $A + C$ ).

## 7 Discussion, Limitations, Assumptions

While we are encouraged by the positive results, we note some caveats and assumptions. First, our chosen metric of style (ABC score) favors a particular definition of style consistent with our own opinions as instructors; different metrics may better suit the needs of other pedagogy. Second, we rely on the instructor to write a good approach hint for a cluster. Third, we assume that the best style solution is represented somewhere in the initial corpus, though this is easily ensured by including the instructor’s reference solution. Fourth, although we have tested the clustering and chain-building on other languages and assignments with good results, the current experiments were conducted on a single assignment in one language. Finally, while student feedback on the types of hints suggests that no hint type’s usefulness dominates the others, we plan to try to isolate the effects of each in future experiments.

A clear limitation of the current system is its ability to examine only a single function at a time. A standard style guideline is to improve a function by refactoring it to use “helper” functions, but our system cannot currently handle such assignments. We would need to enhance our n-TED similarity metric to account for such submissions.

Our system deliberately provides guidance consistent with two observations about how professional programmers learn. The first is the importance of *concrete rather than abstract advice* for improving coding style. The “style guide” provided to students in the course we worked with can be seen as a microcosm of the well-developed paradigms in software engineering for improving code readability and maintainability, including refactoring and applying design patterns. Yet the canonical reference books on those topics [4, 7] feature an abundance of concrete examples to illustrate the abstract points. We speculate that like the professional programmers who are the target audience of such books, students learn better when a hint or technique is situated in a concrete example, as our hints and code skeletons try to do, rather than stated as an abstract principle.

Second, programming requires *active independent learning*. Following good design principles requires knowledge of language features or library functions of which students may be unaware. Both syntactic auto-generated hints and

Class size (number of students)	265	425	448	686	951	986	1607
Number of Clusters	8	3	5	5	3	6	4

**Fig. 6.** Class size vs. number of clusters for seven comparable assignments.

instructor-authored approach hints can point students in the right direction by suggesting, for example, “Consider using a call to `set()`”. Even if a code skeleton is provided with the hint, the skeleton is sufficiently redacted that the student cannot simply copy and paste the code without modification. To improve their code, the student has no choice but to go off and learn about the language feature or library function suggested by the hint or code skeleton, possibly seeking the help of peers or instructors in doing so.

Our system allows instructors and students to enjoy these benefits with a level of instructor effort proportional to the number of clusters, not the number of students. Our system currently focuses on giving feedback for one function or method at a time; since good functions should be short [16], there are only a finite number of strategies that might be used for a function, so we expect the number of clusters to grow very slowly with the number of students. Figure 6 shows that this is indeed the case for seven such assignments we studied.

## 8 Future Work

We plan to field-test this system in one or more large-enrollment campus courses as well as free Massive Open Online Courses (MOOCs) that teach programming skills. A key question is whether we can observe transfer of improved code style skills after students interact with our system; MOOCs would be an excellent testbed for a randomized controlled experiment to measure transfer.

We have not focused on the relatively well-explored area of generating hints for program correctness, in part because we have observed as instructors that students will first work toward a correct program “by any means necessary” (including with the support of automated hints from an intelligent tutoring system), and only later think about refactoring and improving its style (if they think about these things at all). Indeed, this process is reflected in the “red-green-refactor” cycle [5] espoused by the Test-First Development approach within the Agile methodology: programmers are advised to start with nonworking code that fails a correctness test (red), debug it until it passes the correctness test (green), then refactor the code and design to improve readability and maintainability.

## References

1. Ericsson, K., Krampe, R., Tesch-Römer, C.: The role of deliberate practice in the acquisition of expert performance. *Psychol. Rev.* **100**(3), 363–406 (1993)
2. Fast, E., Steffee, D., Wang, L., Brandt, J., Bernstein, M.: Emergent, crowd-scale programming practice in the IDE. In: *SIGCHI Conference on Human Factors in Computing Systems*. Toronto (2014)

3. Fitzpatrick, J.: Applying the ABC metric to C, C++, and Java. In: *More C++ Gems*, pp. 245–264. Cambridge University Press, New York (2000)
4. Fowler, M., Beck, K., Brant, J., Opdyke, W., Roberts, D.: *Refactoring: Improving the Design of Existing Code*. Addison-Wesley, Boston (1999)
5. Fox, A., Patterson, D.: *Engineering Software as a Service*. Strawberry Canyon LLC, San Francisco (2014)
6. Fox, A., Patterson, D., Joseph, S., McCulloch, P.: MAGIC: Massive automated grading in the cloud. In: *CHANGEE (Facing the challenges of assessing 21st century skills in the newly emerging educational ecosystem) workshop at EC-TEL 2015*, Toledo, Spain (2015)
7. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Boston (1994)
8. Glassman, E., Singh, R., Miller, R.: Feature engineering for clustering student solutions. In: *1st ACM Conference on Learning at Scale*, Atlanta (2014)
9. Guskey, T.R.: Closing achievement gaps: revisiting Benjamin S. Blooms, “Learning for Mastery”. *J. Adv. Acad.* **19**(1), 8–31 (2007)
10. Huang, J., Piech, C., Nguyen, A., Guibas, L.: Syntactic and functional variability of a million code submissions in a machine learning MOOC. In: *International Conference on Artificial Intelligence in Education (AIED)*, Memphis (2013)
11. Johnson, S.: Lint, a C program checker. Technical report 65, Bell Labs (1977)
12. Kan, S.H.: *Metrics and Models in Software Quality Engineering*, 2nd edn. Addison-Wesley, Boston (2002)
13. Kulkarni, C.E., Bernstein, M.S., Klemmer, S.R.: PeerStudio: rapid peer feedback emphasizes revision and improves performance. In: *2nd ACM Conference on Learning at Scale*. Vancouver (2015)
14. Lazar, T., Bratko, I.: Data-driven program synthesis for hint generation in programming tutors. In: Trausan-Matu, S., Boyer, K.E., Crosby, M., Panourgia, K. (eds.) *ITS 2014. LNCS*, vol. 8474, pp. 306–311. Springer, Heidelberg (2014)
15. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**(2579–2605), 85 (2008)
16. Martin, R.C.: *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall, Upper Saddle River (2008)
17. Moghadam, J., Roy Choudhury, R., Yin, H., Fox, A.: AutoStyle: toward coding style feedback at scale. In: *2nd ACM Conference on Learning at Scale*, Vancouver (2015)
18. Nguyen, A., Piech, C., Huang, J., Guibas, L.: Codewebs: scalable code search for MOOCs. In: *23rd International Conference on world wide web*, Seoul (2014)
19. Rivers, K., Koedinger, K.R.: Data-driven hint generation in vast solution spaces: a self-improving python programming tutor. *Int. J. Artif. Intell. Educ.*, 1–28 (2015). <http://dx.doi.org/10.1007/s40593-015-0070-z>
20. Shute, V.J.: Focus on formative feedback. *Rev. Educ. Res.* **78**(1), 153–189 (2008)
21. Yin, H., Moghadam, J., Fox, A.: Clustering student programming assignments to multiply instructor leverage. In: *2nd ACM Conference on Learning at Scale*, Vancouver (2015)
22. Zhang, K., Shasha, D.: Simple fast algorithms for the editing distance between trees and related problems. *SIAM J. Comput.* **18**(6), 1245–1262 (1989)

# Estimating Individual Differences for Student Modeling in Intelligent Tutors from Reading and Pretest Data

Michael Eagle<sup>1</sup>✉, Albert Corbett<sup>1</sup>, John Stamper<sup>1</sup>, Bruce M. McLaren<sup>1</sup>,  
Angela Wagner<sup>1</sup>, Benjamin MacLaren<sup>1</sup>, and Aaron Mitchell<sup>2</sup>

<sup>1</sup> Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, USA  
{meagle, corbett, jstamper, bmclaren, awagner,  
maclaren}@andrew.cmu.edu

<sup>2</sup> Department of Biological Sciences, Carnegie Mellon University, Pittsburgh, USA  
apm1@andrew.cmu.edu

**Abstract.** Past studies have shown that Bayesian Knowledge Tracing (BKT) can predict student performance and implement Cognitive Mastery successfully. Standard BKT individualizes parameter estimates for skills, also referred to as knowledge components (KCs), but not for students. Studies deriving individual student parameters from the data logs of student tutor performance have shown improvements to the standard BKT model fits, and result in different practice recommendations for students. This study investigates whether individual student parameters, specifically individual difference weights (IDWs) [1], can be derived from student activities prior to tutor use. We find that student performance measures in reading instructional text and in a conceptual knowledge pretest can be employed to predict IDWs. Further, we find that a model incorporating these predicted IDWs performs well, in terms of model fit and learning efficiency, when compared to a standard BKT model and a model with best-fitting IDWs derived from tutor performance.

**Keywords:** BKT · Genetics · Machine learning · Student modeling

## 1 Introduction

Models of student learning have been successfully employed by intelligent tutoring systems to improve learning outcomes for more than two decades. Student modeling has been used both to individualize curriculum sequencing [1–3] and/or to individualize hint messages [4, 5]. Each of the modeling frameworks cited here employs a Bayesian method to infer student knowledge from student performance accuracy, and Bayesian modeling systems have been shown to accurately predict students' tutor and/or posttest performance [1, 3, 6, 7].

These models generally individualize modeling parameters for individual knowledge components (KCs, also referred to as skills) [8], but not for individual students. Several studies have shown that individualizing parameters for students, as well as for KCs, improves the quality of the models [1, 9–12].

These approaches to modeling individual differences among students have monitored student performance after the fact, in tutor logs that have been previously collected to derive individualized student parameters for the tutor module(s). While these efforts have proven successful, they complicate the actual use of student modeling within an ITS module, since the concurrent estimation and use of individualized parameters in a tutor lesson is generally quite challenging, at best. In this paper we examine whether parameter estimates can be individualized for students prior to embarking on a tutor module, based on student performance in earlier activities. In particular, we examine whether parameter estimates can be individualized based on performance in two activities that naturally precede tutor modules: reading on-line instructional text and taking a conceptual knowledge pretest.

We explore this issue in the Bayesian Knowledge Tracing modeling framework [1] and in a unit of the Genetics Cognitive Tutor [6]. In the following sections we describe Knowledge Tracing, the on-line student activities, the predictors derived from students' reading and pretest activities, and our success in using these predictors to model individual differences in student learning and performance in the tutor.

### 1.1 Bayesian Knowledge Tracing

Bayesian Knowledge Tracing (BKT) [1] employs a two-state Bayesian learning model for each knowledge component (KC) in a tutor curriculum: at any time a student either has learned or not learned a given KC. BKT employs four parameters to estimate the probability that a student has learned each KC:

- $pL_0$  the probability a student has already learned how to apply a KC
- $pT$  the probability a student learns a KC at each opportunity to apply it
- $pG$  the probability a student will guess correctly if the KC is not learned
- $pS$  the probability a student will make an error when the KC has been learned

Cognitive Tutors employ BKT to implement Cognitive Mastery, in which the curriculum is individualized to afford each student just the number of practice opportunities needed to enable the student to "master" each KC.

**Individual Differences.** Knowledge Tracing and Cognitive Mastery generally employ best-fitting estimates of each of the four parameters for each individual KC but *not* for individual students. In this work, we incorporate individual differences among students into the model in the form of individual difference weights. Following Corbett and Anderson [1], four best-fitting weights are estimated for each student, one weight for each of the four parameter types,  $wL_0$ ,  $wT$ ,  $wG$ ,  $wS$ . In estimating and employing these *individual difference weights* (IDWs), we convert each of the four probability estimates to odds form ( $p/(1-p)$ ), multiply the odds by the corresponding student-specific weight and convert the resulting odds back to a probability (Eq. 1). Let  $i$  represent the parameter type, ( $pL_0$ ,  $pT$ ,  $pG$ ,  $pS$ ),  $k$  represent the KC and  $s$  the student. The individually weighted parameter for each KC and student,  $p_{iks}$ , is thus:

$$p_{iks} = p_{ik} * w_{is} / (p_{ik} * w_{is} + (1 - p_{ik})) \quad (1)$$

where  $p_{ik}$  is a best fitting parameter estimate for the KC across all students and  $w_{is}$  is the corresponding individual difference weight for the student.

## 2 Student Activities in This Study

The genetics topic in this study is gene interaction, which examines how two genes can interact in controlling a single phenotypic trait. When two genes, each with a dominant and recessive allele, control a single trait, e.g., coat color in cattle, there can be up to four different resulting phenotypes (four colors). But, there are many ways the two genes can interact that result in only two or three different phenotypes. The goal of the Genetics Cognitive Tutor “process-modeling” lesson in this study is to help students understand and model the different ways two genes can interact to yield two, or three, or four phenotype values. This study focuses on three activities on this topic that students completed in succession: reading gene interaction instructional text online, taking a gene interaction pretest, and finally using the Genetics Cognitive Tutor module on Gene Interaction Process Modeling.

**On-Line Instructional Text.** The online instructional text consisted of 23 screens, structured like pages in a book. Students could move forward and backward through the screens, one screen at a time. After a student touched each page once a “done” button appeared and the student could then continue reading, or exit at any time.

**Conceptual Knowledge Pretest.** Students completed a pretest with nine conceptual questions divided into three topics. The first three questions focused on general knowledge of basic Mendelian transmission with 2 genes, the second three questions focused on process modeling — reasoning about phenotypes that could or could not result from given gene interactions, and the last three questions focused on abductive (backward) reasoning, reasoning about gene interactions that could or could not have given rise to observed phenotypes. This is not a problem-solving pretest; the last six questions are not similar to the Cognitive Tutor problems. Instead, they required students to reason about genetics processes and abductive reasoning more abstractly.

**Genetics Cognitive Tutor: Gene Interaction Process Modeling.** The Genetics Cognitive Tutor (GCT) lesson consisted of 5 process-modeling problems. In each problem, students are given a description of how two genes interact to determine a phenotype, e.g., coat color in cattle. Students (a) map the description onto one of seven gene interaction templates with 3 menus, (b) identify the phenotypes of the four true-breeding genotypes. (c) model the offspring genotypes and phenotypes resulting from two different parental crosses, and finally (d) summarize the phenotypes associated with all possible individual genotypes and how the phenotypes arise.

**The Cognitive Model for GCT Process Modeling.** There are an average of 45 steps in each of these tutor problems. Some of the KCs governing these steps are unique to a problem, while others are applicable in multiple problems. In this analysis we excluded KCs that occurred only one or two times across the 5 problems. Of the remaining 31 KCs, 28 appeared 5 times across the curriculum and 3 appeared 4 times.



### 3 Predictors

Our goal is to examine the feasibility of setting individual difference weights for students before students begin work in a tutor lesson. Several studies have focused on incorporating individual differences into BKT models from the tutor data itself. Corbett and Anderson [1] showed that a BKT model with the four IDWs described in Sect. 2 was a better predictor of individual differences in posttest performance than a standard, non-individualized BKT model. Pardos and Heffernan [10] individualized just the  $pL_0$  parameter, the initial probability the student knows a KC before its first application, based on either the student's first attempt at each KC within the lesson or on all attempts at each KC — and found that either individualized method yielded reliably better fits to multiple data sets than the non-individualized BKT model. Yudelson et al. [11] individualized both learning parameters,  $pL_0$  and  $pT$ , based on student accuracy in a tutor lesson, and found that individualizing  $pT$  yielded reliably better fits than the non-individualized BKT model, while individualizing  $pL_0$  did not reliably improve the goodness of fit. Lee and Brunskill [9] derived 4 individual difference parameters based on performance in a tutor module and found that the resulting model recommended substantially more practice for some students and substantially less practice for others than the non-individualized model.

Finally, in an alternative approach to BKT, a variety of student modeling frameworks grounded in Item-Response Theory employ a single individual difference parameter as a basic component of the model [12–14].

To date, these approaches estimate individual student differences after the fact on tutor data that has already been collected. We examine whether individual differences can be modeled based on prior activities that are natural components of an on-line learning system so that they can be used when a student first begins an ITS module.

#### 3.1 Predictors Derived from Instructional Text Reading Performance

We examine two measures of student reading performance: reading time, and revisiting pages in the text.

**Reading Time.** No prior ITS research employs reading rates to individualize parameters in a learning environment, but there is substantial evidence that reading time varies measurably with comprehension difficulty, and it follows that reading time may prove sensitive to individual differences in comprehension difficulty. Harvey and Anderson [15] showed that reading times for on-line declarative instruction in the ACT Programming Tutor are sensitive to differences in processing time necessary to encode familiar vs. novel material. More generally, an extensive research literature demonstrates that reading time is sensitive to relative comprehension difficulty [16].

**Text Pages Revisited.** Students can read through the instructional text as they would pages in a book. Some students may choose to strictly read forward through the text, while others may choose to revisit earlier pages in the text. Students who re-read text may be demonstrating a meta-cognitive self-monitoring skill, which, if it transfers to problem-solving in the tutor may be correlated with  $p(T)$ , learning rate in the tutor.

### 3.2 Predictors Derived from a Conceptual Knowledge Pretest

Some prior projects have employed pretest accuracy to initialize ITS student models [3, 17]. We examine several measures of students' pretest performance.

**Pretest Accuracy.** We examine whether students' pretest accuracy on each of the three types of pretest questions, general knowledge, process modeling and abductive reasoning, predicts individual difference in learning or performance in the GCT gene interaction process modeling lesson.

**Pretest Answer Changes.** We examine whether changing answers in the pretest from a correct initial answer to an incorrect final answer, or vice versa, is a predictor of individual differences in learning or performance in the tutor module. Checking and changing answers may be evidence of a meta-cognitive self-monitoring skill that may translate into higher learning rates in the tutor module. Alternatively, it may be correlated with the slip rate in the tutor,  $p(S)$ , if the students slipped in making the initial error they are correcting.

**Time on Task.** Finally we examine whether time to complete the pretest is a predictor of individual differences in the tutor module.

## 4 Methods

The data analyzed in this study come from 83 undergraduates enrolled in either a genetics or introductory biology course. All students were recruited to participate in the study for pay. Students participated in two 2.5-hour sessions on consecutive days in a campus computer lab. In this study, the first session focused on gene interaction and students read the on-line gene interaction instructional text, took the on-line pretest, and used the gene interaction process modeling tutor module as the first three activities in this session. The study focuses on modeling the 83 students' first actions on 12,287 problem steps in the tutor module.

### 4.1 Fitting Procedures

First, we found best-fitting group parameter estimates for each of the 4 parameters ( $pL_0$ ,  $pT$ ,  $pG$ ,  $pS$ ) in the standard BKT ("SBKT") model for each of the 31 different knowledge components in the tutor lesson, with nonlinear optimization. The objective function takes the observed opportunities for a single skill and a set of group parameters as input and returns the negative log-likelihood (-LogLik). Optimization ultimately returns the set of group parameters that best fit the skill. Both  $pG$  and  $pS$  were bounded to be less than 0.5, as in [18] to avoid paradoxical results that arise when these performance parameters exceed 0.5 (e.g., a student with a higher probability of knowing a KC is less likely to apply it correctly.)

Second, we re-fit the tutor data with an individualized BKT model: We obtained four best-Fitting Individual Difference Weights (IDWs) for each of the 83 students, one weight for each of the four parameter types,  $wL_0$ ,  $wT$ ,  $wG$ ,  $wS$  to construct this "FIDW"

model. As described in Sect. 1 Eq. 1, each student's four weights are mapped across the best-fitting group learning and performance parameter estimates for each of the 31 KCs to individualize these parameter estimates. The objective function takes the fixed group parameters, the observed opportunities for a student, and a set of IDWs ( $wL_0$ ,  $wT$ ,  $wG$ ,  $wS$ ) and returns the  $-\text{LogLik}$ . Optimization ultimately returns the set of IDWs that maximize the fit for the student.

Third, we derived 12 predictive features, 6 from the on-line reading data and 6 from the pretest data to predict these four individual difference weights for the 83 students, as displayed in Table 1. We performed a factor analysis on log reading times for the 23 individual pages to reduce the number of predictors. The factor analysis yielded a total of four factors (RTF1, RTF2, RTF3, RTF4), which each account for at least 10 % of the variance and align with subtopics in the text, as summarized in the table. These four factors accounted for 54 % of the total variance and additional factors each accounted for less than 5 % of the variance.

**Table 1.** 12 Predictor variables extracted from the on-line reading and pretest data

<b>RTF1</b>	Reading: Time for a 5-page intro with familiar content on basic Mendelian genetics
<b>RTF2</b>	Reading: Time for 6 pages with charts of various ways 2 genes can interact
<b>RTF3</b>	Reading: Time for 3 pages on parental crosses with offspring genotypes & traits
<b>RTF4</b>	Reading: Time for 2 pages with full-page diagrams of dominant & recessive alleles
<b>RRNP</b>	Reading: Total number of previous pages re-read
<b>RRTD</b>	Reading: Total distance traversed (intervening pages) in re-reading text pages
<b>PACC1</b>	Pretest: % Correct for 3 general knowledge questions
<b>PACC2</b>	Pretest: % Correct for 3 process modeling questions
<b>PACC3</b>	Pretest: % Correct for 3 abductive reasoning questions
<b>PCIC</b>	Pretest: Number of answers initially incorrect changed to correct
<b>PCCI</b>	Pretest: Number of answers initially correct changed to incorrect
<b>PTime</b>	Pretest: Total time to complete the pretest

Fourth, we employed each of these 12 variables to independently predict the four sets of IDWs:  $wL_0$ ,  $wT$ ,  $wG$ ,  $wS$ . Since these are multiplicative weights, we fit a transformation of the weights  $w/(1+w)$ . This transformation has the property that the neutral weight 1.0 (which does not modify the corresponding best-fitting group parameter) is the midpoint of the transformed scale. We built a robust regression model with the 12 predictors for each of the IDWs. Robust regression is less sensitive to outliers, variable normality, and other violations of standard linear regression assumptions.

Finally, after deriving the 4 predicted IDWs for each of the 83 students, we recalculated the earlier FIDW BKT model with the predicted IDWs, in place of the best-fitting IDWs to construct the "PIDW" model. In summary, we have three BKT model variants:

1. SBKT: Standard BKT model with best-fitting group parameter estimates,
2. FIDW: Standard BKT model with Fitted Individualized Difference Weights,
3. PIDW: Standard BKT model with Predicted Individualized Difference Weights.

## 5 Results and Discussion

Table 2 summarizes our results. Columns 2 and 3 summarize the overall fit of the standard BKT and the two IDW models to the tutor data. Column 2 displays root mean squared error (RMSE) for the fits and column 3 displays Accuracy (the probability a model correctly predicts students’ correct or incorrect responses, with a 0.5 threshold on predicted accuracy). As can be seen, the FIDW model with best-fitting IDWs fits the tutor data best; it reduces RMSE by 8.7 % compared to the standard, non-individualized SBKT model (0.2794 vs. 0.3059). The new PIDW model with predicted IDWs is about 40 % as successful as the best-fitting FIDW model: The new model reduces RMSE by 3.6 % compared to the standard SBKT model (0.2950 vs. 0.3509). The FIDW model is also about 2.4 % more accurate than the SBKT model (0.8948 vs. 0.8742) while the PIDW model is about 0.8 % more accurate than the SBKT model (0.8812 vs. 0.8742).

**Table 2.** Goodness of fit of the 3 models and differences in practice needed to reach mastery.

Model	RMSE	Accuracy	# Students needing less	# Fewer opportunities needed	# Students needing more	# More opportunities needed
SBKT	0.3059	0.8742	–	–	–	–
FIDW	0.2794	0.8948	56 (46)	17.27 (17.24)	27 (19)	27.04 (27.37)
PIDW	0.2950	0.8812	54 (46)	10.48 (10.96)	27 (19)	11.59 (13.58)

Even small differences in model fits, such as what we found in this study, can have large effects on the amount of recommended work assigned to the student [19]. In order to explore the practical impact of the individualized models, we examined the number of practice opportunities that were necessary for students to reach mastery under each of the three models — that is, the number of opportunities required for  $pL$  (the probability the student has learned a rule) to reach 0.95. This analysis is possible because students completed a fixed curriculum in this study with 4 or 5 opportunities per KC, and most students reached mastery for all of the KCS in the available number of opportunities under all three models.

On average students mastered 94 % of the skills under the SBKT model, 90 % under the FIDW model, and 93 % under the PIDW model. If a student failed to reach mastery on a KC under one model, we conservatively estimated that the student would reach mastery on the next opportunity. On average students needed 57.22 total opportunities to reach mastery of the 31 KCs under the SBKT model, 53.65 total opportunities under the FIDW model, and 53.71 under the PIDW model.

The bottom two rows in the last four columns of Table 2 show how many students need less practice to reach mastery under each of the individualized BKT models than under the standard BKT model, and how many students need more practice. The numbers in parentheses show how many students are common to the two sets under the two models. These columns also show how much more or less practice the students need before the model would consider them to have mastered the KCs.

**Table 3.** Coefficient summary table (<0.10, \*<0.05, \*\*<0.01)

	wL0	wT	wG	wS
<b>(Intercept)</b>	0.0528	0.0785	0.0623	0.8752**
<b>RRTD</b>	-0.0314	0.0317	-0.0061	0.0215
<b>RRNP</b>	0.0221	-0.0055	-0.0056	-0.007
<b>RTF1</b>	-0.0046	-0.0834*	-0.0193	-0.0089
<b>RTF2</b>	0.0338	0.0335	-0.0627*	0.0053
<b>RTF3</b>	0.0131	0.017	0.0192	-0.0215
<b>RTF4</b>	-0.004	0.0204	-0.052*	0.0029
<b>PACC1</b>	0.3504**	0.1469	0.1109	-0.3038**
<b>PACC2</b>	0.2154	-0.0398	0.563**	-0.3021**
<b>PACC3</b>	0.0841	0.4373	0.144	-0.1699
<b>PCIC</b>	0.0096	-0.0248	-0.0327	0.005
<b>PCCI</b>	0.0143	-0.0092	0.0352	0.0189
<b>Ptime</b>	0	0.0004	0	0.0001
<b>RMSE</b>	0.1809	0.2245	0.2055	0.1443

Both individualized models, FIDW and PIDW, substantially modify the amount of practice needed to reach mastery compared to the standard SBKT model. Under the best-fitting FIDW model, 56 students needed less practice to master all the KCs than under the standard SBKT model and on average these students required 17.3 fewer practice opportunities to reach mastery under FIDW than under SBKT. Under the predicted PIDW model, 54 students needed an average of 10.5 fewer opportunities to master all the KCS than under the SBKT model. The two individualized model agree on a set of 46 students who need fewer practice opportunities to reach mastery, but again the FIDW model requires less practice (17.2 opportunities) of these students than the PIDW model (11.0 opportunities).

Under both the FIDW and PIDW models, 27 students need more practice opportunities to reach mastery than under the SBKT model, but students need 27 more practice opportunities under the FIDW model and only 11.6 more opportunities under PIDW model. The two models agree on a set of 19 students who need more practice, but again the FIDW model requires more practice than the PIDW model.

Overall, the FIDW and PIDW models were in 78 % agreement on which students needed fewer or more opportunities to master all the KCs than under the standard SBKT model. The new predicted PIDW model reaches roughly 60 % of the potential learning efficiency gains identified by the best-fitting FIDW model, and does so without the use of the student tutor performance data.

### 5.1 The Predictive Models for the Four Individual Difference Weights

Table 3 displays the coefficients for each of the 12 predictors in the regression model for each of the four IDWs. The predictors that entered reliably into the robust regression model are highlighted with asterisks.

The most interesting result is that student behaviors in reading the text are, in fact, reliable predictors of some individual difference weights. Three of the reading time factors, RTF1, RTF2, RTF4 each reliably predicted one of the four individual differences weights ( $wT$ ,  $wG$ , and  $wG$  respectively). The pages that load on RTF1 specifically are introductory pages on basic Mendelian transmission that should be familiar to all the students and this factor is inversely related to  $wT$  — the longer students take reading what should be familiar text, the lower their learning rate in the tutor. However, student behaviors in re-visiting pages did not reliably predict any IDWs.

Not surprisingly, more pretest variables reliably entered into the four IDW models. Differences in student accuracy on general knowledge (PACC1) and on process-modeling (PACC2) — the same type of reasoning as in this tutor unit — each reliably predict two of the four IDWs. Three other pretest measures, including student accuracy on abductive reasoning questions (PACC3) — a type of reasoning not employed in this tutor unit, total time (PTime) and number of changes from an initially incorrect answer to a correct answer (PCIC) each marginally predicted one IDW.

## 6 Conclusion

We have developed and discussed a method of inserting individual student differences into a traditional Bayesian Knowledge Tracing model that employs pre-tutor reading and test data to predict individual difference weights. This is important because integrating IDWs into an intelligent tutor is much easier if the IDWs can be assigned before the student starts working with the tutor. An advantage of our method is that it can be implemented easily; only a single adjustment needs to be made to each of the group parameters before the student starts the lesson. This initial attempt to pre-set individual difference weights is already quite successful.

The goodness of fit of this new predictive PIDW BKT model falls almost midway between the standard non-individualized SBKT model and the fitted FIDW BKT model. Further, the individualized practice recommendations for the predictive PIDW BKT model are similar to the practice recommendations for the fitted FIDW BKT model, although the new PIDW model does not identify all the opportunities to decrease the amount of practice for some students, nor the need to increase the amount practice for other students, that are identified in the best-fitting FIDW model. However, if implemented, the PIDW model would result in some students needing an average of 18 % fewer total practice opportunities to reach mastery with other students needing an average of 20 % more total practice opportunities. This is a potentially meaningful difference, as it could lead to students spending just the right amount of time with the tutor to achieve mastery.

An important finding is that student data from the reading the instructional text is a useful predictor of learning and performance in an intelligent tutor. Three reading time factors entered reliably into predictive models for individual difference weights in the study. Several conceptual pretest variables also reliably predicted individual differences in learning and performance in an ITS. These results suggest that it is possible to assign IDWs to students before they begin to use the tutor. We expect, but it remains for future

research to explore, that other individual difference frameworks can also benefit from using data from the prior to tutor activities as predictors for initial IDW assignment.

**Acknowledgements.** This research was supported by the National Science Foundation under the grant “Knowing What Students Know: Using Education Data Mining to Predict Robust STEM Learning”, award number DRL1420609.

## References

1. Corbett, A.T., Anderson, J.R.: Knowledge tracing: modeling the acquisition of procedural knowledge. *User Model. User-Adap. Inter.* **4**, 253–278 (1995)
2. Mayo, M., Mitrovic, A.: Optimising ITS behaviour with Bayesian networks and decision theory. *Int. J. Artif. Intell. Educ.* **12**, 124–153 (2001)
3. Shute, V.: Smart: student modeling approach for responsive tutoring. *User Model. User-Adap. Inter.* **5**(1), 1–44 (1995)
4. Ganeshan, R., Johnson, W.L., Shaw, E., Wood, B.P.: Tutoring diagnostic problem solving. In: Gauthier, G., Frasson, C., VanLehn, K. (eds.) *ITS 2000*. LNCS, vol. 1839, pp. 33–42. Springer, Heidelberg (2000)
5. Conati, C., Gertner, A., VanLehn, K.: Using Bayesian networks to manage uncertainty in student modeling. *User Model. User-Adap. Inter.* **12**, 371–417 (2002)
6. Corbett, A.T., MacLaren, B., Kauffman, L., Wagner, A., Jones, E.A.: Cognitive tutor for genetics problem solving: learning gains and student modeling. *J. Educ. Comput. Res.* **42**(2), 219–239 (2010)
7. Gong, Y., Beck, J., Heffernan, N.: Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In: Aleven, V., Kay, J., Mostow, J. (eds.) *ITS 2010, Part I*. LNCS, vol. 6094, pp. 35–44. Springer, Heidelberg (2010)
8. Koedinger, K., Corbett, A., Perfetti, C.: The Knowledge-Learning-Instruction (KLI) framework: bridging the science-practice chasm to enhance robust student learning. *Cogn. Sci.* **36**(5), 757–798 (2012)
9. Lee, J., Brunskill, E.: The impact of individualizing student models on necessary practice opportunities. In: Yacef, K., Zaiane, O., Hershkovitz, A., Yudelson, M., Stamper, J. (eds.) *EDM 2012 Proceedings of the 5<sup>th</sup> International Conference on International Educational Data Mining Society*, pp. 118–125 (2012)
10. Pardos, Z., Heffernan, N.: Modeling individualization in a Bayesian networks implementation of knowledge tracing. In: De Bra, P., Kobsa, A., Chin, D. (eds.) *UMAP 2010*. LNCS, vol. 6075, pp. 255–266. Springer, Heidelberg (2010)
11. Yudelson, M.V., Koedinger, K.R., Gordon, G.J.: Individualized Bayesian knowledge tracing models. In: Lane, H.C., Yacef, K., Mostow, J., Pavlik, P. (eds.) *AIED 2013*. LNCS, vol. 7926, pp. 171–180. Springer, Heidelberg (2013)
12. Pirolli, P., Wilson, M.: A theory of the measurement of knowledge content, access, and learning. *Psychol. Rev.* **105**(1), 58–82 (1998)
13. Cen, H., Koedinger, K.R., Junker, B.: Comparing two IRT models for conjunctive skills. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) *ITS 2008*. LNCS, vol. 5091, pp. 796–798. Springer, Heidelberg (2008)
14. Pavlik, Jr., P.I., Yudelson, M., Koedinger, K.R.: Using contextual factors analysis to explain transfer of least common multiple skills. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) *AIED 2011*. LNCS, vol. 6738, pp. 256–263. Springer, Heidelberg (2011)

15. Harvey, L., Anderson, J.: Transfer of declarative knowledge in complex information processing domains. *Hum.-Comput. Inter.* **11**(1), 69–96 (1996)
16. Zwann, R., Singer, M.: Text comprehension. In: Graesser, A., Gernsbacher, M., Goldman, S. (eds.) *Handbook of Discourse Processes*, pp. 83–121. Erlbaum, Mahwah (2003)
17. Arroyo, I., Beck, J.E., Park Woolf, B., Beal, C.R., Schultz, K.: Macroadapting animalwatch to gender and cognitive differences with respect to hint interactivity and symbolism. In: Gauthier, G., VanLehn, K., Frasson, C. (eds.) *ITS 2000. LNCS*, vol. 1839, pp. 574–583. Springer, Heidelberg (2000)
18. Baker, R.S., Corbett, A.T., Alevan, V.: More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) *ITS 2008. LNCS*, vol. 5091, pp. 406–415. Springer, Heidelberg (2008)
19. Koedinger, K.R., Stamper, J.C., McLaughlin, E.A., Nixon, T.: Using data-driven discovery of better student models to improve student learning. In: Lane, H., Yacef, K., Mostow, J., Pavlik, P. (eds.) *AIED 2013. LNCS*, vol. 7926, pp. 421–430. Springer, Heidelberg (2013)



# Building Pedagogical Models by Formal Concept Analysis

Giuseppe Fenza and Francesco Orciuoli<sup>(✉)</sup>

Department of Management and Innovation Systems,  
University of Salerno, via Giovanni Paolo II, 132 - 84084 Fisciano, SA, Italy  
{gfenza,forciuoli}@unisa.it  
<http://www.unisa.it>

**Abstract.** The Pedagogical Model is one of the main components of an Intelligent Tutoring System. It is exploited to select a suitable action (e.g., feedback, hint) that the intelligent tutor provides to the learner in order to react to her interaction with the system. Such selection depends on the implemented pedagogical strategy and, typically, takes care of several aspects such as correctness and delay of the learner's response, learner's profile, context and so on. The main idea of this paper is to exploit Formal Concept Analysis to automatically learn pedagogical models from data representing human tutoring behaviours. The paper describes the proposed approach by applying it to an early case study.

**Keywords:** Intelligent tutoring systems · Pedagogical model · Formal concept analysis · Conceptual scaling · Association rule mining

## 1 Introduction

An Intelligent Tutoring System (ITS) is a software system providing adaptive educational experiences. Some of the main features of an ITS are: providing learning activities coherently with learners' current knowledge and skills in order to foster meaningful learning; providing individualized feedback able to stimulate next learning activities and avoid frustration, demotivation and disengagement due to unsuccessful performances; providing hints helping learners without replacing them during the execution of their learning tasks. From the architectural viewpoint, an ITS is typically divided into five conceptual components [3]: *Expert Model* representing the domain knowledge; *Pedagogical (Tutor) Model* providing the knowledge infrastructure to tailor the selection and the provisioning of the teaching elements according to the student model; *Domain Model* containing the knowledge about the actual teaching material; *Student Model* storing learners' characteristics like, for instance, details about the learner's current problem-solving state and long-term knowledge progress, which are essential for adapting the experience; *Communication (User Interface) Model* that is responsible of the interaction between learner and system. This paper mainly focuses on the automatic building of the pedagogical model and proposes an approach based on Formal Concept Analysis (FCA) [2, 4] to learn that model

from data gathered by observing: (i) learner's interactions with an ITS, and (ii) the support provided by the tutor according to that ITS experience. In [8] the responsibility of a pedagogical module (that can be considered the software implementation of a pedagogical model) is depicted by considering its interactions with the other modules. In particular, the pedagogical module is defined as the set of instructional techniques and strategies and the processes to select them. Such module receives the information on the learner's states (cognitive, affective, psychomotor, social, etc.) and provides the right generalized domain independent strategy to sustain the learning process. In literature, several works, dealing with the learning of pedagogical rules, exist. In particular, the authors of [7] focus on the automatic hints generation by exploiting an approach based on path finding over the graph of the solution space. In [1], the use of reinforcement learning algorithms to build tutoring rules is proposed. Such algorithms continuously adjust the set of existing rules by considering their effectiveness when applied to the intelligent tutoring systems. Furthermore, the authors of [6] adopt educational data mining techniques to generate pedagogical recommendations. With respect to the existing works, the proposed approach aims at learning a pedagogical model by using FCA. In particular, the lattice, resulting from its application, is a flexible structure that can be automatically exploited by a software agent, or explored by a human tutor, to support decision-making (tutoring) processes.

## 2 Overall Approach

This section provides the explanation of the proposed approach and the background knowledge needed to understand the work details.

### 2.1 Behaviour of an ITS

From the architectural viewpoint, an ITS is a software agent whose behavior can be described in an outer (external) loop and an inner loop [10]. The external loop provides learners with a sequence of *tasks* (typically of problem solving) of different difficulties. The default behavior foresees that the next task, to be presented, has a difficulty greater than the previous presented one (mastery learning). However, if the learners' results are negative the ITS can propose a next task with a lower difficulty or propose alternative learning content. This is called macro-adaptivity and a number of strategies can be applied to implement it. Moreover, there is an inner loop for each task where a sequence of *steps* has to be executed by the learners in order to achieve the task objectives and provide a solution for the associated problem. The ITS can provide: (i) adaptive feedback (positive, negative, etc.) in response to the learners' answers for the current step, and/or (ii) hints to anticipate the next step of the same task. Typically, this is called micro-adaptivity. The set of all possible actions that can be carried out by the tutor is called *tutoring actions*. The selection of the right tutoring action can be accomplished on the basis of pedagogical strategies, learners' profiles,

context, domain, etc. According to the above description, the pedagogical model can be defined as a set of rules (pedagogical rules) that are executed to select one or more suitable tutoring actions (at the end of a step/task) according to some variables like, for instance, learner's performance, emotional/affective states.

## 2.2 Formal Concept Analysis

FCA is a technique widely applied to build *formal lattices* that are an effective data structure exploited in order to address tasks like data mining, ontology learning and merging, etc. FCA works on a set of objects having a set of attributes and considers the notion of *formal context* specifying which objects have what attributes; thus a formal context may be viewed as a binary relation between the object set and the attribute set with the values 0 and 1. FCA starts with a formal context defined as a triple  $K = (G, M, I)$ , where  $G$  is a set of objects,  $M$  is a set of attributes, and  $I$  is a binary relation, s.t.  $I \subseteq G \times M$ . In particular, the existence of the relation  $(g, m) \in I$  means that the object  $g$  has the attribute  $m$ . The formal context is often represented as a "cross table": the rows represent the *formal objects* and the columns are *formal attributes*; the relations between them are represented by the crosses. In the proposed approach, interactions and their features play respectively the role of objects and attributes in the matrix describing the formal context.

**Definition 1 (Formal Concept).** *Given a context  $(G, M, I)$ , for  $A \subseteq G$  applying a derivation operator,  $A' = \{m \in M \mid \forall g \in A : (g, m) \in I\}$ , and for  $B \subseteq M$ ,  $B' = \{g \in G \mid \forall m \in B : (g, m) \in I\}$ . A formal concept is identified with a pair  $(A, B)$ , where  $A \subseteq G$ ,  $B \subseteq M$  such that  $A' = B$  and  $B' = A$ .  $A$  is called the *extent* and  $B$  is called the *intent* of the concept  $(A, B)$ .*

The subsumption relation among concepts is defined as follows:

**Definition 2 (Subconcept).** *Given two concepts  $C_1 = (A_1, B_1)$  and  $C_2 = (A_2, B_2)$ , then  $C_1$  is a subconcept of  $C_2$  (equivalently,  $C_2$  is superconcept of  $C_1$ ) if  $(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 (\Leftrightarrow B_2 \subseteq B_1)$ . The set of all concepts of a particular context, ordered in this way, forms a complete lattice.*

Starting from the above definitions it is possible to build the formal lattice by exploiting algorithms like that provided in [5].

## 2.3 Approach Workflow

The main idea of this work is to build a coherent dataset to provide a good formal context on which applying FCA to generate the lattice that can be further processed to obtain pedagogical rules and, lastly, collect them into the final pedagogical model. The proposed approach can be explained by using the workflow presented in Fig. 1 where it is clear that the process consists of four phases.

The first phase is needed to define the schema to organize data on which applying FCA. The schema is defined by modifying an existing one and applying

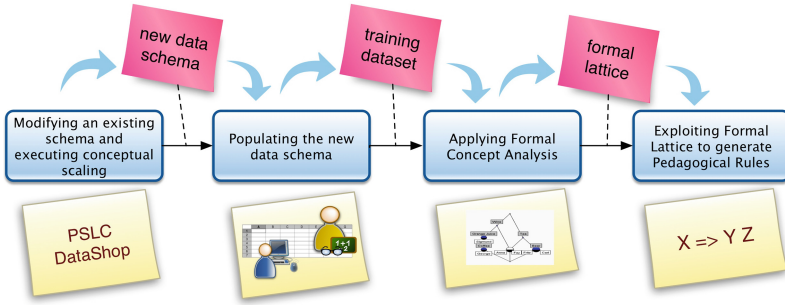


Fig. 1. Workflow of the proposed approach.

some adjustments on it. The second phase consists in collecting information on learners' and tutors' behaviours during a real ITS experience (where the automatic tutor is replaced by a human tutor) and populating the previously defined schema. The third phase is focused on applying FCA over the obtained dataset and generating the lattice that organizes the concepts mined from data. Lastly, in the fourth phase it is possible to explore the lattice in order to obtain the rules that will be collected into the pedagogical model.

### 3 Building the Formal Lattice

This Section describes the steps needed to build the formal lattice exploited to generate the pedagogical rules.

#### 3.1 PSLC Data Schema

In order to provide a generally applicable approach for building pedagogical models by means of FCA, we propose to start from a data schema adopted by the Pittsburgh Science of Learning Center (PSLC) DataShop<sup>1</sup> for organizing data generated by users' interactions (with an ITS). PSLC DataShop is a data repository and a web application for learning science researchers. It provides secure data storage as well as an array of analysis and visualization tools available through a web-based interface. Typically, datasets, adopting such schema, are tables composed by rows (records) automatically generated by the logging module of an ITS. Each row describes a single interaction (transaction) among the learner and the system (intelligent tutor). In particular, the learner is provided with content describing a task (**Problem Name**), its level (**Level-Number**) and a step (**Step Name**). Next, the learner has to provide a response that is modelled by using the fields **Action**, **Input**, **Student Response Type** and **Student Response Subtype** and has a correctness value provided by the field **Outcome**. The learner's response and other data are evaluated by the

<sup>1</sup> <https://pslcdatashop.web.cmu.edu>.

tutor who reacts by selecting a suitable action to provide. The tutoring action is modelled by means of the fields **Tutor Response Type** and **Tutor Response Subtype**. Further details on tutoring actions are provided by fields like **Feedback Text** and **Feedback Classification**. Additional fields like **Is Last Step** (a value informing if this step is the last step of the task), **Attempt At Step** (a value informing on how many attempts have been done for the current step in the current task), **Duration** (how many seconds the interaction takes), **Help Level** and **Total Num Hints** are used to define a context used by the tutor in order to perform more accurate selections of her actions. Moreover, **KC** and **KC Category** (KC stays for Knowledge Component) attributes are used to associate to a specific step, in a given task, a knowledge pre-requisite the learner should have to successfully execute the step.

### 3.2 Adjusting the Schema

Starting from the data schema previously described (for the next we will call it *PSLCschema*) we need to process it and provide a derived schema: *DERISchema* in order to achieve two objectives: (i) making the schema suitable for the construction of a pedagogical model; (ii) adjusting the schema in order to satisfy the FCA requirements. In order to define such schema we need to apply some transformations (of existing fields/attributes) and the *conceptual scaling* operation. In particular, the latter is needed to provide fields having only two possible values (true/false, yes/no, etc.) to effectively support the FCA process.

Conceptual scaling is useful when objects (in the formal context) have many-valued attributes. For instance, if you consider objects like cars it is possible to have the attribute **color** that is typically associated to a set of plausible values (red, green, etc.) and, thus, it is a many-valued attribute. In the case of the *PSLCschema* we have mostly many-valued attributes like, for instance, **Attempt At Step**, whose admissible values are in the set of numbers  $\{1, 2, 3, 4, \dots\}$ . Now, our goal is to derive a single-valued context from a many-valued context and thus we need to map each many-valued attribute (in the *PSLCschema*) into a set of single-valued attributes. In particular, from **Attempt At Step** it is possible to derive four single-valued attributes: **One Attempt**, **Two Attempts**, **Three Attempts** and **More Than Three Attempts**. The number of derived single-valued attributes depends on both the semantic of the multi-valued attribute and the goal of the process. In this case, we need to know if the learner provided a correct response soon (one attempt), if she learns by trying (two or three attempts) or if she has serious problems to provide the correct response (more than three attempts).

Figure 2(a) provides four rows corresponding to four interactions and show the values originally associated to the attribute **Attempt At Step**. Values **int-1**, **int-2**, etc. are interaction identifiers. Once the **Attempt At Step** attribute has been scaled we obtain the result reported in Fig. 2(b).

The derived schema (*DERIdataset*) is presented in Fig. 3. Some additional considerations have to be done: there is no need to separate different session interactions or different learners interactions because the focus is on tutoring

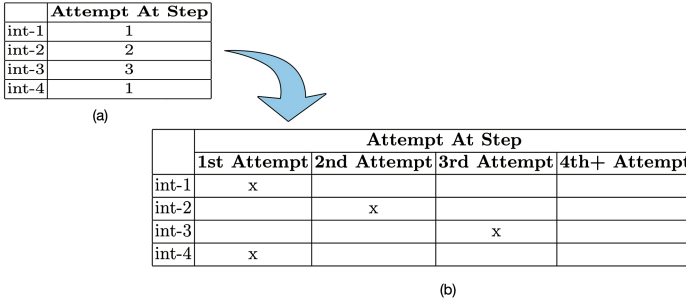


Fig. 2. Applying Conceptual Scaling to Attempt At Step.

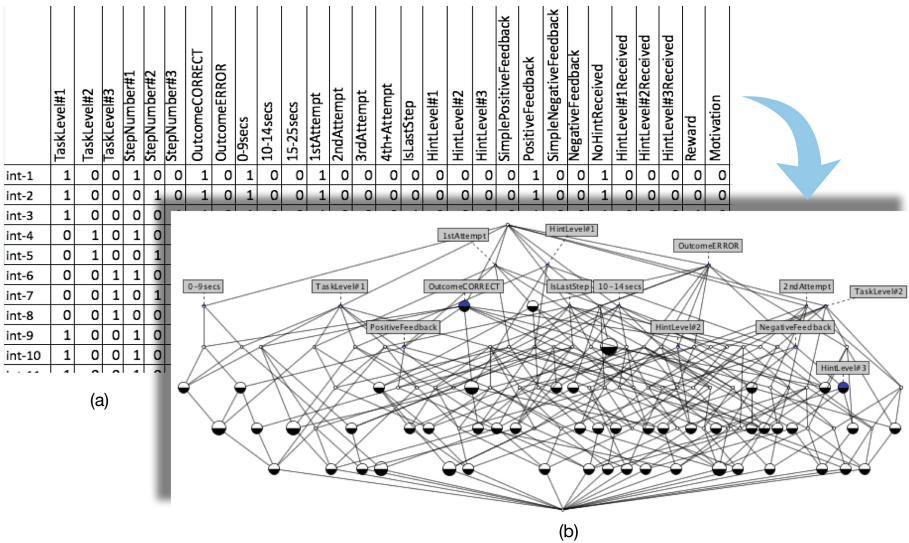


Fig. 3. Fragment of the formal context and reduced version of the formal lattice.

decisions; learners interact with the system by using always the same device and user interface; the obtained pedagogical model should be used in the same context in which the training dataset has been generated; we did not consider attributes concerning learners' profiles because PSLC does not provide it but we think that such attributes could be very useful for the construction of an effective pedagogical model; the pedagogical rules obtained at the end of the process are domain-independent rules.

Among the whole set of attributes of the derived schema (see Fig. 3 for the whole list of attributes) there are few ones belonging to the tutoring actions category, i.e., feedback, hints and additional feedback (e.g. reward and motivation). Such category is important because in the next sections we will define pedagogical rules on the basis of it.

### 3.3 Applying FCA over Prepared Data

In order to apply the FCA over the prepared data we need to perform a three-steps process: (i) defining the formal context; (ii) computing all the formal concepts; (iii) drawing the formal lattice.

The first step is trivial, in fact, the organization of data into rows (interactions) and the scaling of all multi-valued attributes allow us to simply obtain a table representing the formal context. The dataset (satisfying the *DERISchema*) has been constructed by contemporary observing the interactions of three 5–6 years old learners with an Educational Game App<sup>2</sup> running on Android Tablets (and Smartphones) and the tutor actions provided by an expert (human) teacher to those learners. The game is organized in problems (tasks) and each problem can be solved by correctly executing a sequence of steps. At the end of each step, the teacher reacts to the learners' answers by providing feedback, hints or other tutoring actions. A Python script has been used to process the gathered data and transform them into a CSV file representing the formal context. The CSV file has been subsequently loaded in the ConExp tool [11].

The second step and the third step are executed by running the ConExp *Build Lattice* function. At the end of these steps we obtained the whole set of formal concepts that are connected to produce a formal lattice. The result of this computation is reported in Fig. 3. Take care that, the one presented in the figure is only a sample lattice obtained by applying the *Build Lattice* function to only a part of the available attributes in the formal context.

## 4 Exploiting the Formal Lattice

The formal concepts, obtained by applying FCA, can be explored in order to let emerge correlations among attributes. In particular, we find correlations among the attributes in the set of tutoring actions (feedback, additional feedback and hints) and the set of attributes representing (in some way) the learner's interaction with the task/step content, the context in which such interaction comes to life and the information (metadata) about the task/step itself. We will focus on two approaches to extract pedagogical rules from the lattice. The first one must be performed by a human-based exploration. The second one is completely automatic and is based on association rule mining algorithms.

### 4.1 Pedagogical Rules as Association Rules

The pedagogical rules we are looking for are a special case of *association rules*. In general, given a set of items  $I = I_1, I_2, \dots, I_m$  and a database of transactions  $D = t_1, t_2, \dots, t_n$  and each  $t_i = I_{i_1}, I_{i_2}, \dots, I_{i_k}$  and  $I_{i_j} \in I$ , an association rule is an implication of the form  $X \Rightarrow Y$  where  $X, Y \subset I$  are sets of items and  $X \cap$

<sup>2</sup> This App has been developed by one of the authors of this paper and presented in a paper accepted at the 8<sup>th</sup> International Conference on Computer Supported Education (CSEDU 2016).



$Y = \emptyset$ . In order to evaluate the importance of an association rule, two features called *support* and *confidence* are defined. The *support* for an association rule  $X \Rightarrow Y$  is the percentage of transactions in the database that contain  $X \cup Y$ . The *confidence* for an association rule  $X \Rightarrow Y$  is the ratio of the number of transactions that contain  $X \cup Y$  to the number of transactions that contain only  $X$ .

The FCA-based association rules are defined as follows. An implication  $X \Rightarrow Y$  holds, if the largest concept, in the lattice, that is below the concepts that are generated by the attributes of  $X$  is below all concepts that are generated by the attributes in  $Y$ . These rules are called *exact association rules*. Furthermore, valid *approximate association rules* are rules  $X \Rightarrow Y$ , such that concepts  $(A_1, B_1)$  and  $(A_2, B_2)$  exist, with  $(A_1, B_1)$  being a direct upper neighbor of  $(A_2, B_2)$ , such that  $X = B_1$  holds and  $X \cup Y = B_2$  holds [9].

Lastly, in our work, a *pedagogical rule* is a special case of an association rule.

**Definition 3.** *Given  $TA \subset M$  the set of all attributes in  $M$  belonging to the tutoring actions category, a pedagogical rule is an association rule (exact or approximate)  $X \Rightarrow Y$  where  $X$  is a set of attributes not belonging to  $TA$  and  $Y$  is a set of attributes belonging to  $TA$ .*

## 4.2 Exploring Paths in the Lattice

In the lattice representation, each attribute in the formal context is used to label exactly one node. If you select a specific attribute you can find the concept (node) in the lattice to whom such attribute is attached and find exact association rules by ascending the paths in the diagram, starting from this node.

Let us show an example. If you select the `HintLevel#3` attribute it is possible to know which other attributes are present in the data when there is `HintLevel#3`. This information can be easily obtained by ascending the paths in the diagram starting from the node associated to this attribute and collecting all attributes found during exploration. In particular, if we consider the formal lattice produced by FCA over the whole dataset, it is possible to note, as shown in Fig. 4, that when (in data) the attribute `HintLevel#3` appears, also the attributes `Motivation`, `Negative Feedback`, `OutcomeERROR`, `3rdAttempt`, `15-25secs` and `HintLevel#2Received` are present. Moreover, starting from the node labeled with `Motivation` and `NegativeFeedback` it is possible to observe that when these two attributes are present, also the attributes `OutcomeERROR` and `3rdAttempt` appear. The previous two assertions can be easily transformed into pedagogical rules. By using the second assertion we can compute the rule:

$$\text{OutcomeERROR, 3rdAttempt} \Rightarrow \text{NegativeFeedback, Motivation}. \quad (1)$$

The previous rule has the following meaning: if the learner's answer isn't correct and this is the third erroneous attempt for the current step in the current task, then the (intelligent) tutor has to provide a negative feedback and, at the same time, it has to provide some action to motivate the learner. Rule 1 has



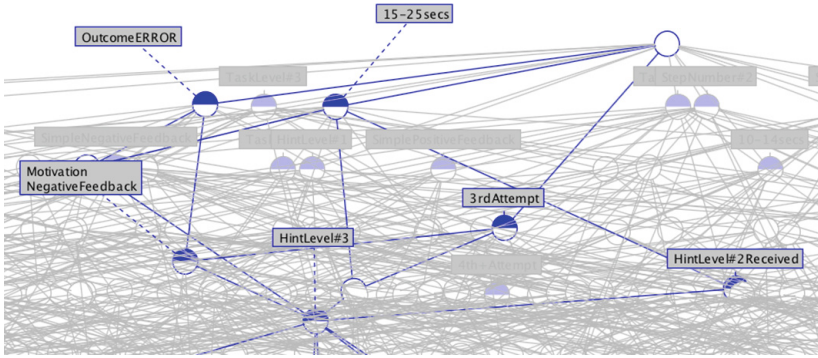


Fig. 4. Analysis of the path ascending from `HintLevel#3` attribute.

support  $\approx 0,13$  and confidence = 1. The ascending exploration produces always confidence = 1, i.e., 100 % of confidence (exact rules).

Moreover, it is possible to observe the descending paths in the diagram starting from the node associated to `HintLevel#2`. More in details, such node has four children nodes representing uncertain associations among attributes. The difference between the ascending and the descending approaches is that in the second case we obtain less than the 100 % of confidence because there exist some transactions in which the considered attributes do not appear all together. Thus, we obtain weaker pedagogical rules. For the attribute `HintLevel#2`, one of the possible descending path produces the rule:

$$\text{OutcomeERROR, TaskLevel\#1, HintLevel\#1Received} \Rightarrow \text{HintLevel\#2.} \quad (2)$$

This rule has support  $\approx 0,05$  and confidence = 0,6 (60%). The descending exploration produces rules with confidence less than 100 % (approximate rules).

### 4.3 Extracting Association Rules Automatically

In addition to the human-based exploration, it is possible to automatically mine association rules by means of the Duquenne-Guigues method [9] for getting exact association rules and the Luxenburger approach [9] for mining approximate association rules. In order to obtain valid pedagogical rules we have to erase those rules that do not respect the definition of pedagogical rule provided in Sect. 4.1.

## 5 Results and Final Remarks

This work proposes an approach to build pedagogical models by using Formal Concept Analysis. The approach has been experimented by using a dataset built with observations gathered from both the execution of an Educational App for Android and the tutoring actions provided by a (human) tutor. The result of

the process consists of a set of association rules forming the pedagogical model. In our experimentation (supported by the ConExp tool) we applied  $minsup = 6$  and  $minconf = 60$ . We obtained 228 rules (49 exact and 179 approximate rules). When we applied the filter for pedagogical rules we obtained 30 rules (8 exact and 22 approximate rules). Although the approach has been experimented by using a small dataset, the experimentation provided promising results. Future works will follow two directions: (i) considering domain-dependent features and semantic technologies to allow context-lifting for pedagogical model; (ii) adding a reinforcement learning algorithm to adjust, at run-time, the learnt rules.

## References

1. Chi, M., VanLehn, K., Litman, D.: Do micro-level tutorial decisions matter: applying reinforcement learning to induce pedagogical tutorial tactics. In: Aleven, V., Kay, J., Mostow, J. (eds.) ITS 2010, Part I. LNCS, vol. 6094, pp. 224–234. Springer, Heidelberg (2010)
2. De Maio, C., Fenza, G., Loia, V., Senatore, S.: Hierarchical web resources retrieval by exploiting fuzzy formal concept analysis. *Inf. Process. Manage.* **48**(3), 399–418 (2012)
3. El-Sheikh, E., Sticklen, J.: A framework for developing intelligent tutoring systems incorporating reusability. In: Moonis, A., Mira, J., de Pobil, A.P. (eds.) IEA/AIE 1998. LNCS, vol. 1415, pp. 558–567. Springer, Heidelberg (1998)
4. Ganter, B., Wille, R.: *Formal Concept Analysis: Mathematical Foundations*. Springer Science & Business Media, New York (2012)
5. Kuznetsov, S.O., Obiedkov, S.A.: Comparing performance of algorithms for generating concept lattices. *J. Exp. Theor. Artif. Intell.* **14**(2–3), 189–216 (2002)
6. Paiva, R.O.A., Bittencourt Santa Pinto, I.I., da Silva, A.P., Isotani, S., Jaques, P.: A systematic approach for providing personalized pedagogical recommendations based on educational data mining. In: Trausan-Matu, S., Boyer, K.E., Crosby, M., Panourgia, K. (eds.) ITS 2014. LNCS, vol. 8474, pp. 362–367. Springer, Heidelberg (2014)
7. Rivers, K., Koedinger, K.R.: Automating hint generation with solution space path construction. In: Trausan-Matu, S., Boyer, K.E., Crosby, M., Panourgia, K. (eds.) ITS 2014. LNCS, vol. 8474, pp. 329–339. Springer, Heidelberg (2014)
8. Sottolare, R.A.: *Fundamentals of adaptive intelligent tutoring systems for self-regulated learning*. Technical report, DTIC Document (2015)
9. Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., Lakhal, L.: Intelligent structuring and reducing of association rules with formal concept analysis. In: Baader, F., Brewka, G., Eiter, T. (eds.) KI 2001. LNCS (LNAI), vol. 2174, pp. 335–350. Springer, Heidelberg (2001)
10. Vanlehn, K.: The behavior of tutoring systems. *Int. J. Artif. Intell. Educ.* **16**(3), 227–265 (2006)
11. Yevtushenko, S.A.: System of data analysis concept explorer. In: *Proceedings of the 7th National Conference on Artificial Intelligence KII*, vol. 2000 (2000)

# Predicting Learning from Student Affective Response to Tutor Questions

Alexandria K. Vail<sup>1</sup>(✉), Joseph F. Grafsgaard<sup>2</sup>, Kristy Elizabeth Boyer<sup>4</sup>,  
Eric N. Wiebe<sup>3</sup>, and James C. Lester<sup>1</sup>

<sup>1</sup> Department of Computer Science, North Carolina State University,  
Raleigh, NC, USA  
{akvail,lester}@ncsu.edu

<sup>2</sup> Department of Psychology, North Carolina State University, Raleigh, NC, USA  
jfgrafsg@ncsu.edu

<sup>3</sup> Department of STEM Education, North Carolina State University,  
Raleigh, NC, USA  
wiebe@ncsu.edu

<sup>4</sup> Department of Computer and Information Science and Engineering,  
University of Florida, Gainesville, FL, USA  
keboyer@ufl.edu

**Abstract.** Modeling student learning during tutorial interaction is a central problem in intelligent tutoring systems. While many modeling techniques have been developed to address this problem, most of them focus on cognitive models in conjunction with often-complex domain models. This paper presents an analysis suggesting that observing students' multimodal behaviors may provide deep insight into student learning at critical moments in a tutorial session. In particular, this work examines student facial expression, electrodermal activity, posture, and gesture immediately following inference questions posed by human tutors. The findings show that for human-human task-oriented tutorial dialogue, facial expression and skin conductance response following tutor inference questions are highly predictive of student learning gains. These findings suggest that with multimodal behavior data, intelligent tutoring systems can make more informed adaptive decisions to support students effectively.

## 1 Introduction

A fundamental goal of the intelligent tutoring systems (ITS) community is modeling student learning during tutoring so that an ITS can effectively adapt its tutorial support [1,2]. Student models often observe the behavior and performance of the student and then use this information to estimate the student's 'hidden' understanding of the material [3,4]. A variety of approaches to student modeling have been investigated and employed successfully, such as cognitive modeling through knowledge tracing [5] and performance factor analysis [6]. Critically, these approaches rely on student task behaviors such as problem-solving traces.

While problem-solving traces have been shown to indicate student progress or lack thereof, other work has found that multimodal data streams can be highly indicative of students' state during learning. For example, multimodal data such as facial expression, posture, and gestures can predict affective outcomes, such as frustration and engagement [7,8]. Additionally, multimodal data can contribute to inferring incoming student characteristics, including self-efficacy [9], personality [10], and domain expertise [11]. These studies of multimodal behavior during learning pose a critical open question: *what is the relationship between learning gain and students' multimodal behavior during tutoring?*

To investigate this research question, this paper presents an analysis of student multimodal trace data immediately after tutor questions. In the domain of introductory computer science and in the specific context of tutor inference questions, we investigate whether multimodal trace data contributes to accurately predicting student learning gains. The results show that a subset of facial expression events, together with skin conductance response, immediately after tutor questions are highly predictive of students' future performance on a posttest. These results reveal the significant potential of leveraging multimodal trace data for student modeling.

## 2 Related Work

The work reported in this paper is grounded in research on multimodal data generated during learning, particularly facial expressions and physiological responses. Multiple studies have explored student facial expression during learning activities. For example, D'Mello and Graesser developed a multimodal classifier of expert-tagged student affect using student dialogue, posture, and facial expression features [12]. A multimodal model built upon all three of these categories yielded higher classification accuracy than using a subset of the data streams, achieving a Cohen's  $\kappa = 0.33$  for fixed emotion judgments and  $\kappa = 0.39$  for spontaneous judgments. A study with Wayang Outpost attempted to predict self-reported affective states using a similar multimodal feature set, with best fit models achieving a correlation coefficient of up to  $r = 0.83$  [13].

There is some evidence that physiological response is predictive of student learning. Stein and Levine proposed a theoretical model in which activation of the autonomic nervous system indicates a mismatch between incoming information and existing knowledge, akin to cognitive disequilibrium [14,15]. Further, they suggest that this state is nearly always an indication of learning. Indeed, some preliminary work on physiological responses to learning interactions has indicated support for this theory. Other work has revealed that skin conductance response after negative feedback and student expressions of uncertainty were highly predictive of student learning [16]. Negative feedback and student expressions of uncertainty are both likely to occur in states of cognitive disequilibrium.

### 3 Study Data

We investigate the relationship between multimodal behavior traces and learning within a tutorial dialogue corpus of computer-mediated human-human tutoring for introductory computer science. The subject matter focus of the tutorial dialogue is Java programming [17, 18]. Each session was conducted within an online remote tutoring system, shown in Fig. 1. The interface consists of four panes: the task description, the student's Java source code, the compilation and execution output of the program, and the textual dialogue messages exchanged between tutor and student. The content of the interface was synchronized in real time between the tutor and the student, with the tutor's interactions constrained to sending textual dialogue messages and progressing to the next task.

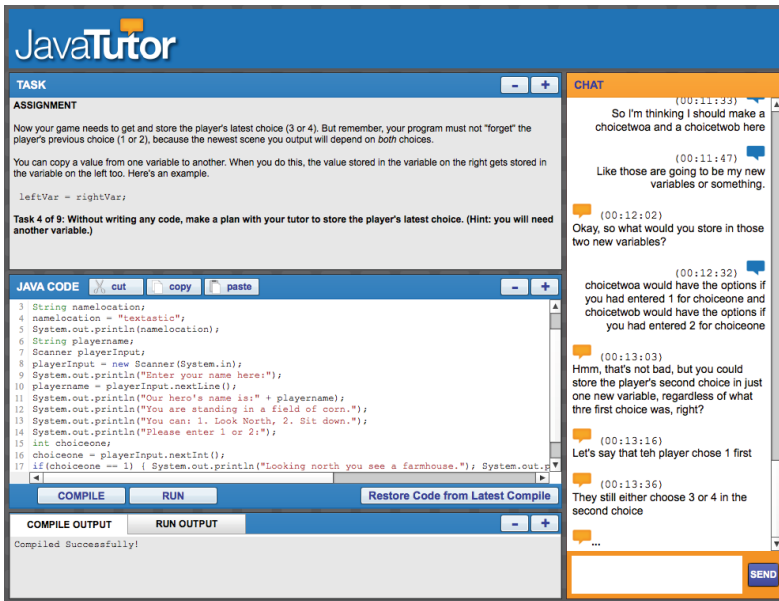


Fig. 1. The web-based tutorial interface for Java programming.

Human tutors ( $N = 5$ ) were primarily graduate students with previous experience in tutoring or teaching introductory programming. Student participants ( $N = 67$ ) were university students in the United States with an average age of 18.5 years ( $s = 1.5$  years). Data were collected using multiple multimodal sensors as seen in Fig. 2, including a Kinect depth camera, an integrated webcam, and a skin conductance bracelet (see following subsections for more detail). The data were collected during the fall 2011 and spring 2012 semesters. Each student's participation was distributed over four weeks across six 40-min sessions. (This analysis examines only data from the first lesson.) Before and after each tutorial session, students completed a content-based pretest and identical posttest.



**Fig. 2.** Multimodal instrumented tutoring session, including a Kinect depth camera to detect posture and gesture, a webcam to detect facial expression changes, and a skin conductance bracelet to detect electrodermal activity.

Normalized learning gain was calculated using the student’s pretest and posttest scores, as shown in Eq. 1.

$$norm\_gain = \begin{cases} \frac{post - pre}{1 - pre} & post > pre \\ \frac{post - pre}{pre} & post \leq pre \end{cases} \quad (1)$$

### 3.1 Task Event and Dialogue Features

As each student progressed through the session, the tutoring system logged dialogue messages, typing in the code window, and task progress. No strict turn-taking was enforced. Students and tutors could type dialogue messages at any time. All tutor and student dialogue messages were tagged automatically (for details please see [19]) with a dialogue act annotation scheme for task-oriented tutorial dialogue [20].

The present analysis focuses on a key tutor dialogue move: inference questions. Inference questions are questions that require reasoning about content knowledge or formulating a plan. For example, ‘*How can you fix this error?*’, and ‘*How do you think this problem can be solved?*’ are inference questions. Questions of this nature are known to stimulate cognitive disequilibrium in students [15], which is considered to be a crucial step in knowledge acquisition [21]. The analysis presented here explores the hypothesis that student multimodal traces following tutor inference questions are significantly predictive of student learning gain.

### 3.2 Facial Expression Features

Facial expression features were automatically identified by a state-of-the-art facial expression recognition and analysis software, FACET (commercial software that was preceded by a research version known as the Computer Emotion Recognition Toolbox, CERT) [22]. FACET provides frame-by-frame tracking of facial action units according to the Facial Action Coding Scheme [23]. These action units include such expressions as AU4 BROW LOWERER, AU15 LIP CORNER DEPRESSOR, and AU23 LIP TIGHTENER (see Fig. 4 for illustration). Facial features were extracted from webcam videos. The FACET software provides an *Evidence* measure for each facial action unit, indicating the chance that the target expression is present.

### 3.3 Electrodermal Activity Features

Skin conductance is a type of electrodermal activity [24]. Skin conductance has two components, *tonic*, which changes gradually over time, and *phasic*, which changes in abrupt peaks [25] in response to a stimulus. These peaks represent *skin conductance response (SCR) events*.

A challenge in analyzing SCRs in the context of a series of task and dialogue events is that SCRs occur in close temporal proximity, even overlapping with each other. In order to address this concern, this analysis utilizes Continuous Decomposition Analysis, which decomposes skin conductance data into its tonic and phasic components and detects overlapping SCRs [25]. This analysis was conducted using the Ledalab MATLAB software, which additionally supports event-related analysis in the context of SCRs. The threshold for detecting SCRs was set to a minimum change in amplitude of  $\delta = 0.02 \mu\text{S}$ , based on the results of prior analysis on this corpus of tutorial dialogue [16].

## 4 Analysis

The primary objective of this analysis is to identify how multimodal signals following tutor questions can predict learning gain. In order to do this, we examine the three seconds (a manually-determined interval) following the delivery of an inference question from a tutor. Student behavior was characterized using the following features, which were all provided to the predictive models reported below. (Only the first two of these were found to be significant predictors, as the results section will describe.)

1. Average Evidence measure for each of the facial expression action units during the interval.
2. Number of skin conductance responses (SCRs) identified during the interval.
3. Percentage of the interval in which a one-hand-to-face or two-hands-to-face gesture was observed.
4. Average student distance from the workstation during the interval.
5. Average difference between the highest and lowest points of the student's body from the workstation during the interval (indicating leaning).

To examine the predictiveness of multimodal traces immediately following tutor inference questions, we averaged the value of each multimodal feature described above across each tutoring session. These features are conditional averages of the form  $Avg(Feature|TutorInferenceQ)$ . If we built predictive models using only these features, we may identify conditional features that are components in a broader, unconditional association between a multimodal feature and learning gain. To control for this we also included a feature  $Avg(Feature)$ , which represents the session-wide average value of that multimodal feature (not conditioned on any preceding event). For each student and for each feature type listed above, one value of  $Avg(Feature|TutorInferenceQ)$  and one value of  $Avg(Feature)$  were generated.

All features were standardized by subtracting the mean and dividing by the standard deviation. This set of features was then used in a stepwise regression modeling procedure that maximizes the leave-one-student-out cross-validated  $R^2$  value (the coefficient of determination), while enforcing a strict  $p$ -value cut-off of  $p < 0.05$  after Bonferroni correction for the significance of each included feature.

## 5 Results

The results show that student facial expression features and skin conductance response are significantly predictive of learning gain. The predictive model for normalized learning gain includes six features, four of which are specific to the multimodal traces from the three-second interval following an inference question from the tutor. The other two predictors are session-wide features (Table 1).

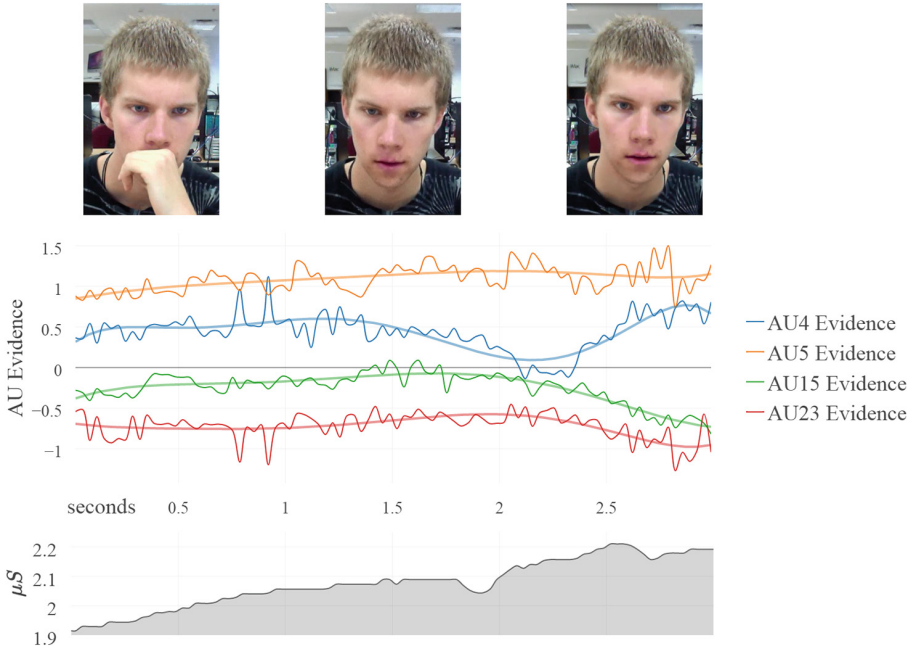
**Table 1.** Predictive model for standardized normalized learning gain after tutor inference questions<sup>a</sup>.

Normalized Learning Gain =	$R^2$	$p$
+1.4012 * AU23 (Session-wide)	0.0445	< 0.001
+0.1523 * SCRs	0.2457	< 0.001
+0.7548 * AU5	0.2669	< 0.001
-0.3502 * AU15 (Session-wide)	0.0024	0.002
+0.2856 * AU4	0.0789	0.005
-0.4503 * AU23	0.1893	0.004
+0.6440 (Intercept)		1.000

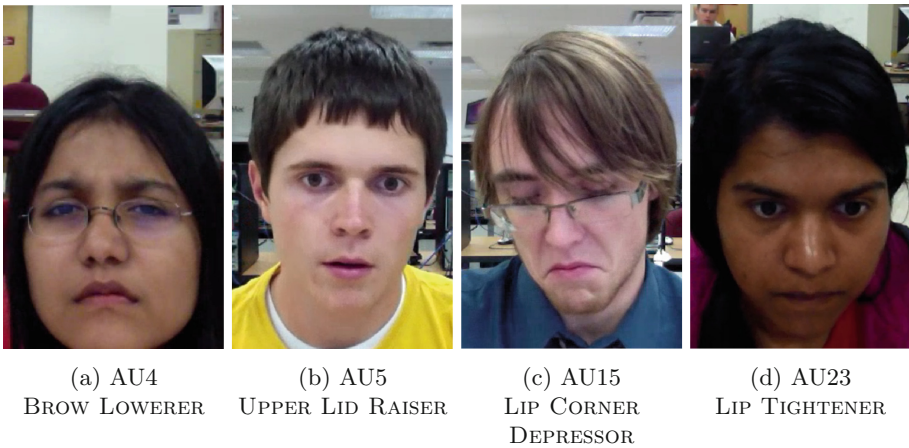
**Leave-One-Out Cross-Validated  $R^2 = 0.8277$**

<sup>a</sup>This model was built as part of a more expansive exploratory analysis. The  $p$ -values reported here have already undergone a Bonferroni correction  $p \leq \alpha/n$ , where  $n = 21$  is the number of statistical tests conducted, in order to reduce the familywise error rate to  $\alpha = 0.05$ .





**Fig. 3.** A segment of the multimodal data collection illustrating a student’s response to an inference question from the tutor (“How can you fix your code?”). Sample webcam frames are displayed, along with standardized FACET readings of the four significant facial action units and the student’s electrodermal activity. Note the overall increase in AU4 and AU5, along with a decrease in AU15, as well as activation of an SCR at approximately two seconds. This student achieved one of the highest learning gains observed in the current study.



**Fig. 4.** Sample frames from the student webcam illustrating the four significant facial action unit features appearing in the predictive model, as identified by FACET.

The results show that session-wide features account for a relatively small portion of the variance in learning gain: only two facial action unit features were selected from this set. One of these is frequency of AU15 LIP CORNER DEPRESSOR (Fig. 4c) which is negatively predictive of student learning gain.

The other session-wide facial action unit feature that is significant in the model is AU23 LIP TIGHTENER (Fig. 4d), the session-wide presence of which is positively correlated with learning. However, this feature is also significant as a stimulus-specific feature but in the opposite direction. Higher frequency of AU23 immediately after tutor inference questions is *negatively* predictive of learning gain. The final two facial action unit features that are significantly predictive of learning gain are AU4 BROW LOWERER and AU5 UPPER LID RAISER (Fig. 4a and b, respectively) both positively correlated with learning gain. Finally, the number of skin conductance responses (SCRs) occurring after tutor inference questions is a significant positive indicator of learning gain (Fig. 3).

## 6 Discussion

Tutor inference questions require students to reason about their knowledge or formulate plans for problem solving. Consequently, student multimodal signals following these pivotal moments offer key insights into the cognitive-affective phenomena that are associated with learning.

Students displaying more frequent AU15 LIP CORNER DEPRESSOR after tutor inference questions learned less. This action unit has been found in prior task-oriented studies to be a strong predictor of lack of focus [26]. In contrast, prior work has indicated that AU23 LIP TIGHTENER is frequently associated with frustration or focused concentration [26, 27]. In the current study, AU23 session-wide was positively associated with learning, but immediately following tutor inference questions it was negatively associated. This finding points to the importance of further study to tease apart frustration from focused concentration, particularly in the context of questions that require reasoning or possibly in the face of cognitive disequilibrium.

AU5 UPPER LID RAISER following tutor inference questions was positively predictive of learning in the current study, and it has previously been found to indicate focused attention in task-oriented domains [26]. Expressing this indicator of engagement directly following tutor questions may suggest that the student is thinking critically about the solution.

AU4 BROW LOWERER following tutor questions was predictive of increased learning in the current study. AU4 has been associated with frustration [7, 8], and in general, frustration has been found to be inversely related to learning gains [28, 29]. However, these results have been discovered mostly in the context of session-wide features; different analyses have found these features indicative of confusion in shorter time periods [30]. Both frustration and confusion are frequently associated with cognitive disequilibrium [15] which, when resolved, is beneficial to learning [21]. AU4 following tutor inference questions may indicate

cognitive disequilibrium at first, the resolution of which fosters learning. In the tutoring sessions investigated here, AU4 session-wide does not have a significant relationship with learning.

Prior work on this tutorial dialogue corpus has suggested the importance of skin conductance responses following events that indicate cognitive disequilibrium, such as student expressions of uncertainty or encountering negative feedback from the system [16]. We might reasonably infer that inference questions from the tutor may induce cognitive disequilibrium [15], and so skin conductance responses following these questions may indicate heightened response that facilitates learning. Further study is needed to elucidate the causal relationships between tutor questions, student cognitive disequilibrium, skin conductance response, and learning.

## 7 Conclusion and Future Work

Modeling student learning during tutoring is central to intelligent tutoring systems. The results presented here demonstrate that student multimodal traces can provide insight into cognitive-affective phenomena while yielding accurate predictions of student learning during tutoring sessions. In particular, facial expression and skin conductance responses during tutoring were highly predictive of learning as indicated by improvement from pretest to posttest. These results complement and expand upon prior work investigating these features by decomposing a tutorial session into salient moments and investigating short-term responses versus long-term session features.

Future work should investigate how student multimodal signals at other critical moments in tutoring sessions are related to student learning. For example, introducing new concepts, or when a student reaches an impasse, are likely key moments in tutoring. Another promising direction for future work is to examine affective outcomes such as frustration or engagement, since multimodal signal analysis holds much promise for providing real-time predictions of these phenomena as well. It is hoped that this line of work will lead to powerful, domain-independent predictive measures of learning and other cognitive-affective phenomena that intelligent tutoring systems can use to adaptively support student learning.

**Acknowledgements.** The authors wish to thank the members of the LearnDialogue and Intellimedia groups at North Carolina State University for their helpful input. This work is supported in part by the Department of Computer Science at North Carolina State University and the National Science Foundation through Grants IIS-1409639, CNS-1453520, and a Graduate Research Fellowship. Any opinions, findings, conclusions, or recommendations expressed in this report are those of the participants, and do not necessarily represent the official views, opinions, or policy of the National Science Foundation.

## References

1. Hartley, D., Mitrović, A.: Supporting learning by opening the student model. In: Cerri, S.A., Gouardères, G., Paraguaçu, F. (eds.) ITS 2002. LNCS, vol. 2363, pp. 453–462. Springer, Heidelberg (2002)
2. Gluga, R.: Long term student learner modeling and curriculum mapping. In: Aleven, V., Kay, J., Mostow, J. (eds.) ITS 2010, Part II. LNCS, vol. 6095, pp. 227–229. Springer, Heidelberg (2010)
3. Corbett, A.T., Anderson, J.R.: Student modeling and mastery learning in a computer-based programming tutor. In: Proceedings of the 2nd International Conference on Intelligent Tutoring Systems, pp. 413–420 (1992)
4. Stevens, R., Soller, A., Cooper, M., Sprang, M.: Modeling the development of problem solving skills in chemistry with a web-based tutor. In: Lester, J.C., Vicari, R.M., Paraguaçu, F. (eds.) ITS 2004. LNCS, vol. 3220, pp. 580–591. Springer, Heidelberg (2004)
5. Corbett, A.T., Anderson, J.R.: Knowledge tracing: modeling the acquisition of procedural knowledge. *User Model. User-Adap. Inter.* **4**, 253–278 (1995)
6. Pavlik Jr., P.I., Cen, H., Koedinger, K.R.: Performance factors analysis - a new alternative to knowledge tracing. In: Proceedings of the 14th International Conference on Artificial Intelligence in Education, pp. 531–538 (2009)
7. Grafsgaard, J.F., Wiggins, J.B., Boyer, K.E., Wiebe, E.N., Lester, J.C.: Automatically recognizing facial expression: predicting engagement and frustration. In: Proceedings of the 6th International Conference on Educational Data Mining, pp. 43–50 (2013)
8. Grafsgaard, J.F., Wiggins, J.B., Boyer, K.E., Wiebe, E.N., Lester, J.C.: Automatically recognizing facial indicators of frustration: a learning-centric analysis. In: Proceedings of the Humane Association Conference on Affective Computing and Intelligent Interaction, pp. 159–165 (2013)
9. Scherer, S., Weibel, N., Morency, L.P., Oviatt, S.: Multimodal prediction of expertise and leadership in learning groups. In: Proceedings of the 1st International Workshop on Multimodal Learning Analytics (2012)
10. Biel, J.I., Teijeiro-Mosquera, L., Gatica-Perez, D.: FaceTube: predicting personality from facial expressions of emotion in online conversational video. In: Proceedings of the 14th International Conference on Multimodal Interaction, pp. 53–56 (2012)
11. Oviatt, S., Cohen, A.: Written and multimodal representations as predictors of expertise and problem-solving success in mathematics. In: Proceedings of the 15th International Conference on Multimodal Interaction, pp. 599–606 (2013)
12. D’Mello, S.K., Graesser, A.C.: Multimodal semi-automated affect detection from conversational cues, gross body language, and facial features. *User Model. User-Adap. Inter.* **20**, 147–187 (2010)
13. Arroyo, I., Cooper, D.G., Burleson, W., Woolf, B.P., Muldner, K., Christopherson, R.: Emotion sensors go to school. In: Proceedings of the 14th International Conference on Artificial Intelligence in Education, pp. 17–24 (2009)
14. Stein, N.L., Levine, L.J.: Making sense out of emotion: the representation and use of goal-structured knowledge. In: *Psychological and Biological Approaches to Emotion*, pp. 45–73 (1990)
15. Piaget, J.: *The Origins of Intelligence*. International University Press, New York (1952)

16. Hardy, M., Wiebe, E.N., Grafsgaard, J.F., Boyer, K.E., Lester, J.C.: Physiological responses to events during training: Use of skin conductance to inform future adaptive learning systems. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, pp. 2101–2105 (2013)
17. Ha, E.Y., Grafsgaard, J.F., Mitchell, C.M., Boyer, K.E., Lester, J.C.: Combining verbal and nonverbal features to overcome the ‘Information Gap’ in task-oriented dialogue. In: Proceedings of the 13th Annual SIGDIAL Meeting on Discourse and Dialogue, pp. 247–256 (2012)
18. Mitchell, C.M., Ha, E.Y., Boyer, K.E., Lester, J.C.: Learner characteristics and dialogue: recognising effective and student-adaptive tutorial strategies. *Int. J. Learn. Technol.* **8**(4), 382–403 (2013)
19. Vail, A.K., Boyer, K.E.: Adapting to personality over time: examining the effectiveness of dialogue policy progressions in task-oriented interaction. In: Proceedings of the 15th Annual SIGDIAL Meeting on Discourse and Dialogue, pp. 41–50 (2014)
20. Vail, A.K., Boyer, K.E.: Identifying effective moves in tutorial dialogue: on the refinement of speech act annotation schemes. In: Proceedings of the 12th International Conference on Intelligent Tutoring Systems, pp. 199–209 (2014)
21. Graesser, A.C., Olde, B.A.: How does one know whether a person understands a device? The quality of the questions the person asks when the device breaks down. *J. Educ. Psychol.* **95**(3), 524–536 (2003)
22. Littlewort, G., Whitehill, J., Wu, T., Fasel, I., Frank, M., Javier, M., Bartlett, M.: The computer expression recognition toolbox (CERT). In: Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition and Workshops, pp. 298–305 (2011)
23. Lepper, M.R., Woolverton, M.: The wisdom of practice: Lessons learned from the study of highly effective tutors. *Impact of Psychological Factors on Education, Improving Academic Achievement*, pp. 135–158 (2002)
24. Boucsein, W.: *Electrodermal Activity*. Springer Science & Business Media, New York (2012)
25. Benedek, M., Kaernbach, C.: A continuous measure of phasic electrodermal activity. *J. Neurosci. Methods* **190**, 80–91 (2010)
26. Vural, E., Cetin, M., Ercil, A., Littlewort, G., Bartlett, M., Movellan, J.: Drowsy driver detection through facial movement analysis. In: Proceedings of the 12th International Conference on Human-Computer Interaction, pp. 6–18 (2007)
27. Mortillaro, M., Mehu, M., Scherer, K.R.: Subtly different positive emotions can be distinguished by their facial expressions. *Soc. Psychol. Pers. Sci.* **2**(3), 262–271 (2011)
28. Goldin, I.M., Carlson, R.: Learner differences and hint content. In: Lane, H.C., Yacef, K., Mostow, J., Pavlik, P. (eds.) *AIED 2013*. LNCS, vol. 7926, pp. 522–531. Springer, Heidelberg (2013)
29. San Pedro, M.O.Z., Baker, R.S.J., Gowda, S.M., Heffernan, N.T.: Towards an understanding of affect and knowledge from student interaction with an intelligent tutoring system. In: Lane, H.C., Yacef, K., Mostow, J., Pavlik, P. (eds.) *AIED 2013*. LNCS, vol. 7926, pp. 41–50. Springer, Heidelberg (2013)
30. D’Mello, S., Lehman, B., Pekrun, R., Graesser, A.: Confusion can be beneficial for learning. *Learn. Instr.* **29**, 153–170 (2004)

# Integrating Real-Time Drawing and Writing Diagnostic Models: An Evidence-Centered Design Framework for Multimodal Science Assessment

Andy Smith<sup>1</sup>(✉), Osman Aksit<sup>2</sup>, Wookhee Min<sup>1</sup>, Eric Wiebe<sup>2</sup>,  
Bradford W. Mott<sup>1</sup>, and James C. Lester<sup>1</sup>

<sup>1</sup> Department of Computer Science, North Carolina State University, Raleigh,  
NC 27695, USA

{pmsmith4, wmin, bwmott, lester}@ncsu.edu

<sup>2</sup> Department of STEM Education, North Carolina State University,  
Raleigh, NC 27695, USA

{oaksit, wiebe}@ncsu.edu

**Abstract.** Interactively modeling science phenomena enables students to develop rich conceptual understanding of science. While this understanding is often assessed through summative, multiple-choice instruments, science notebooks have been used extensively in elementary and secondary grades as a mechanism to promote and reveal reflection through both drawing and writing. Although each modality has been studied individually, obtaining a comprehensive view of a student's conceptual understanding requires analyses of knowledge represented across both modalities. Evidence-centered design (ECD) provides a framework for diagnostic measurement of data collected from student interactions with complex learning environments. This work utilizes ECD to analyze a corpus of elementary student writings and drawings collected with a digital science notebook. First, a competency model representing the core concepts of each exercise, as well as the curricular unit as a whole, was constructed. Then, evidence models were created to map between student written and drawn artifacts and the shared competency model. Finally, the scores obtained using the evidence models were used to train a deep-learning based model for automated writing assessment, as well as to develop an automated drawing assessment model using topological abstraction. The findings reveal that ECD provides an expressive unified framework for multimodal assessment of science learning with accurate predictions of student learning.

**Keywords:** Assessment · Multimodality · Evidence-centered design

## 1 Introduction

Formative assessment can play a central role in enabling intelligent tutoring systems (ITSs) to provide students with personalized, adaptive learning experiences [1]. Effective formative assessment can be used to infer students' underlying mental models as well as their movement through learning progressions [2, 3]. The models inferred

from these assessments can then be used as the basis for real-time feedback and adaptive support [4]. Formative assessment can improve science learning, and because science learning often features both drawing and writing activities, intelligent tutoring systems for science education should support multimodal assessment of both student drawing and student writing [5].

Evidence-centered design (ECD) provides a systematic approach to designing and developing assessments [6]. ECD identifies multiple phases in the design process, each with its own explicit goals. These phases include the creation of a Competency Model, an Evidence Model, and a Task Model that operate in concert to recognize evidence of conceptual understanding from student work. For multimodal assessment, ECD can provide a systematic way of mapping between learning goals and student artifacts from various modalities that show evidence of student learning. Of particular interest is how ECD might provide a unified framework for assessing both written and drawn artifacts of student work for formative purposes.

This paper introduces a new ECD-based framework for multimodal science assessment. First, we use a multimodal approach to ECD to define a competency model and a multimodal evidence model for elementary science to understand how conceptual understanding about magnetism is revealed in both drawing and writing tasks. Specifically we aim to evaluate student writings and drawings using a common competency model that contributes to a deeper understanding of the relative contributions of the two modalities. Second, with the long-term goal of integrating multimodal assessments into an ITS, we present computational models for evaluating student writings and drawings in real-time and compare their predictive accuracy to expert human scorings. The findings reveal that ECD provides a unified framework for multimodal assessment of science learning with accurate predictions of student learning.

## 2 Related Work

Though much less investigated than short-answer writing assessment, there has been some work on assessment of learner-generated drawings. *Mechanix* [7] utilizes free-hand sketch recognition to convert student drawings in the domain of statics into free-body equations that the system can then analyze and provide corrective feedback. Van Joolingen et al.'s *SimSketch* system seeks to merge free-hand sketching with modeling science phenomena. The system first segments the free-hand drawing into distinct objects that can be annotated by the user with a variety of behaviors and attributes [8]. Students can then run a simulation based on those behaviors and attributes. *SimSketch* was used in a planetarium setting by elementary students for modeling and simulation, showing evidence for increasing student learning and engagement. *CogSketch* [9], which aims to support open-domain sketch understanding, has been employed to compare the drawings of expert and novice users to analyze differences in drawings, as well as differences in the ways the drawings are created.

Automatic grading of written short answers has long been the focus of the ITS and natural language processing (NLP) communities, with short answers being defined as natural language responses varying in length from one sentence to one paragraph [10]. Many of these approaches, such as the widely used Latent Semantic Analysis [11], rely

on “bag-of-words” approaches that focus primarily on the occurrence or frequency of words that appear in text. Other approaches, such as the ones embodied in Educational Testing Service’s C-Rater, use a variety of preprocessing techniques to generate syntactic relationships between words in a sentence [12]. The technique employed by Dzikovska et al. uses dependency parses in a facet-based approach to assessment, which provides more fine-grained information about assessments than a monolithic overall score [13]. Other approaches have used word embeddings and convolutional neural networks that incorporate information across sequences of words [14]. Our work proposes an approach combining word conversion techniques and feedforward neural networks to address noisy students’ answers that contain various forms of misspellings to implement a reliable writing assessment solution.

Recent years have also seen a growing interest in evidence-centered design as a method for interpreting the complex data streams generated by virtual learning environments. Gobert et al. used ECD to create predictive models of student inquiry skills from action logs generated in a science microworld [15]. Rupp et al. utilized ECD in both the design of an interactive training application for employees of a networking company, as well as the design of the accompanying assessments [16]. Finally, ECD is used in conjunction with computational methods such as Bayesian networks and stacked autoencoder networks to construct “stealth” assessments for educational games [2, 17]. Our work builds on this line of investigation by introducing a unified framework for *multimodal assessment* of both drawing and writing based on ECD.

### 3 The LEONARDO Digital Science Notebook

Data for the work reported here was collected with LEONARDO, a cloud-based digital science notebook developed for elementary school science education [14]. LEONARDO was designed for use in the classroom and runs on both desktop computers and tablets. LEONARDO supports inquiry learning by providing adaptive support to students as they engage in both virtual and physical lab activities as well as providing them with tools to create their own visual and written representations (Fig. 1). LEONARDO currently

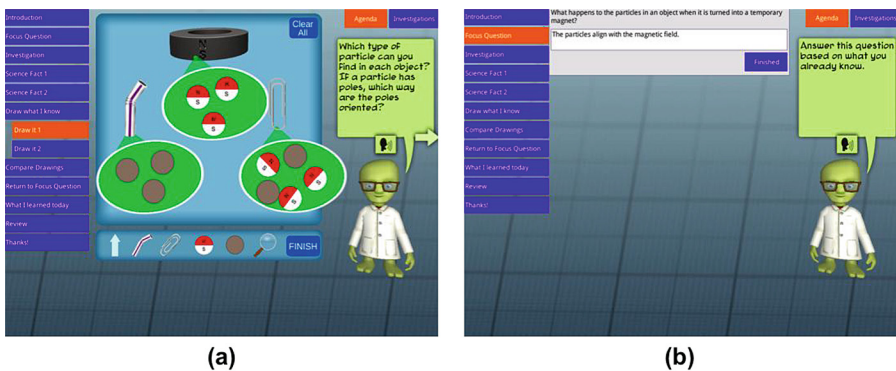


Fig. 1. Examples of LEONARDO drawing (a) and writing (b) prompts.



supports three science units: *Electricity*, *Magnetism*, and *Weather*. Each unit consists of several subunits driven by Focus Questions (FQs), which are organized around an open-ended driving question (e.g., What makes a magnet magnetic?). The activities and tasks employed in each FQ were designed to facilitate student learning of the underlying science concept to answer the driving question.

In most FQs, students are required to construct written and visual explanations by completing a series of drawing and writing tasks to solidify and extend their understanding of the observed scientific phenomena. To facilitate meaningful writing composition, the writing tasks require students to compose short responses, and in some cases a starter prompt is given to help students build an argument sentence, (e.g., A magnet attracts a paperclip because...). In drawing tasks students manipulate built-in pictorial symbols representing key scientific concepts relevant to the current FQ. Pictorial symbol manipulation includes selecting appropriate symbols from a toolbar and organizing them in the drawing field by modifying their direction, alignment and relative placement. At the end of each FQ, students are presented with their original answers to the driving question and offered to construct a new response based on what they have learned so that they can recognize and monitor the change in their own understanding of the subject matter by comparing their old and new response.

#### 4 ECD Coding Framework

Evidence-centered design is a holistic approach to designing, implementing, evaluating, and delivering educational assessments [18]. ECD recognizes assessment as an evidentiary reasoning process that entails making arguments on learning based on the limited evidence provided by the learner [6]. ECD has gained considerable popularity in a broad range of fields in recent years, and it has been used in conjunction with several forms of learning technologies, including game-based learning environments [2], and educational data mining [1, 16]. The ECD framework formalizes the different phases in the assessment design process as “layers” and each layer has its own specific objectives and associated products. In this work, we employed ECD’s Conceptual Assessment Framework layer to analyze our assessment models in LEONARDO’s magnetism unit, and we generated a comprehensive rubric to score students’ drawing and writing artifacts based on this analysis.

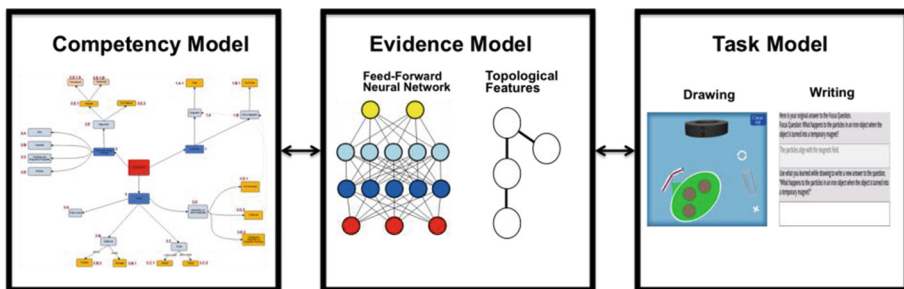


Fig. 2. Application of the ECD process to science notebook task data.

The Conceptual Assessment Framework layer consists of three components: the Competency Model, the Evidence Model, and the Task Model (Fig. 2). The first layer begins by identifying and determining what collection of knowledge, skills or practices on which the learner will be assessed. These concepts are then combined to form the Competency Model, sometimes referred to as the Student Model. Once defined, values in the Competency Model can be inferred across multiple interactions using a variety of techniques including Bayesian knowledge tracing and dynamic belief networks [4]. The second step is determining what types of observations of student work or artifacts will provide measurable evidence for the target competencies, including defining specific evidence for each of the modalities to be evaluated. The Evidence Model is the product of this layer. The final step focuses on designing tasks—the Task Model—that will give the learner relevant opportunities to provide the expected evidence. When the possible evidence that students may exhibit have been identified, the tasks can then be designed that will require students to generate those evidence. Mislevy and Haertel note that ECD is a sequential process but can include iterations and refinements within and across the layers during the design cycle [6].

To develop our models we used a subset of a larger sample of fourth grade students from the 42 schools who implemented LEONARDO's magnetism module during the 2013-2014 and 2014-2015 school years. The participating schools are located across the United States. A total of 98 students from 19 different classrooms were selected based on the requirement that they completed all eight of the drawing and writing tasks in FQ-3 and FQ-5 in the Magnetism unit. Although there are six instructional units (FQs) in the magnetism module, we chose to analyze FQ-3 (What happens to the particles in an iron object when the object is turned into a temporary magnet?) and FQ-5 (Can magnets work through materials like paper, cardboard, and metal foil?) because they provide the richest set of drawing and writing tasks in terms of the number and variety of the scientific concepts that they address.

An initial coding was completed by two human raters individually grading students' drawing and writing artifacts. Initial practice trials were completed using data from students not included in the final sample to train raters, formalize the rubric, and align their interpretations. Cohen's kappa ( $\kappa$ ) was run to determine the inter-rater reliability based on a randomly selected subset of 20% of responses coded by both raters. A high level of agreement was found between the two raters' drawing scores,  $\kappa = .838$  (95% CI, .806 to .869,  $p < .001$ ) and a substantial level of agreement between the two raters' writing scores,  $\kappa = .754$  (95% CI, .669 to .838,  $p < .001$ ).

**Table 1.** Means<sup>a</sup> and standard deviations for total scores (N = 98)

Questions	Min	Max	Mean	SD
FQ-3 drawings	0	100	62.6	32.9
FQ-5 drawings	4	100	60.8	26.2
FQ-3 writings	0	88	28.5	23.6
FQ-5 writings	10	100	63.3	20.9
Post-test	20	95	68.0	20.4

<sup>a</sup>Scores are converted to a 0–100 scale for ease of interpretation.

Table 1 shows the students' drawing and writing scores and post-test performance. Although the mean scores of FQ-3 and FQ-5 drawings and FQ-5 writings are close to each other, the mean score of FQ-3 writings is much lower than the others. This might be explained by the fact that one of the FQ-3 writing tasks asks students to compare their two drawings, and thus has a higher number of potential concepts to be observed than the other writings. However, most students' responses compared only one or two aspects of their drawings resulting in the lower scores. A hierarchical multiple regression test was conducted to analyze how student knowledge revealed by multiple drawing and writing artifacts predict their post-test performance. The first model, which uses only FQ-3 and FQ-5 drawing scores, significantly predicted approximately 36% variance in the post-test scores  $F(2, 95) = 27.17, p < .001, R^2 = .364$ , while the second model containing FQ-3 and FQ-5 both drawing and writing scores significantly predicted about 48% variance in total in the post-test scores  $F(4, 93) = 27.75, p < .001, R^2 = .483$ , producing an  $R^2$  change of .119.

## 5 Automated Assessment Systems

With the goal of integrating these new assessments into LEONARDO, we used the human scorings to devise computational assessment models to assess both student drawing and writing. We next introduce the drawing scoring, using a rule-based system based on topological features, as well as the writing scoring, using word conversion techniques combined with feedforward neural networks.

### 5.1 Automated Assessment of Symbolic Drawings

Building on techniques developed in our previous work [14], drawings are represented as a set of objects and their associated  $x$ ,  $y$  coordinates and rotation. For example, the set of possible objects in the drawing space include a paper clip, a plastic straw, a magnifier bubble to indicate microscopic properties, inert particles, magnetic particles, and an arrow. For this work we decompose the drawing into a set of topological relations between these objects. Topological features allow us to discretize a wide range of continuous features in a way that facilitates symbolic manipulation. In this case we use these topological relations to generate a labeled graph representation of the drawing. The first step in the translation from drawings to topological graphs is encoding the primary elements for the domain. Initially, this consists of defined elements drawn by the student. In the later steps these elements can be combined into new elements, such as converting a group of similarly rotated magnetic particles into a single "aligned particles" element.

After creating the nodes of the graph, edges are generated based on topographical relationships between elements. Many potential 2D relationships are encoded, with the goal of generating a sufficiently large number of relationships to capture the relevant information expressed by the drawing, while excluding irrelevant relationships that will unnecessarily complicate the computation. For example, one solution could be to generate a complete set of all possible relations for every pairwise combination of

elements in the drawing, though, this approach would quickly produce a large amount of features, many of which are unnecessary for the intended analysis. To simplify this task, each object is assigned a type. For each type, we specify a set of related types for which topological relationships will be generated. The set of qualitative 2D relations used in this work are *near*, *far*, *intersects*, and *aligns-with*. Finally, more complex relationships are defined based on combinations of atomic spatial relations. For example, the point of the magnifier object intersecting with a paperclip generates a set of *contains* relationships between the paperclip and the elements that have been drawn within the larger magnification bubble.

Finally, to convert the symbolic representation into a rubric score, we assign a set of rules for each rubric component. For example, for the component associated with the concept of a straw containing only non-magnetic particles, a *contains(straw, inert)* relationship must exist, as well as *contains(straw, aligned)* and *contains(straw, unaligned)* not existing. These rules can also be defined to compare relationships between drawings, as is required by some components of the competency model.

## 5.2 A Feedforward Neural Network for Short Answer Analysis

Building automated writing assessments entails devising computational models that take as input students' text-based responses and predict as output their grades according to the pre-specified rubric discussed in Sect. 4. A key challenge posed by the automated assessment of elementary-grade students' writing is effectively dealing with many forms of misspelled words, including cognitive misconceptions (e.g., *magnetism* misspelled by *magnetizm*) and typographical errors (e.g., *paperclip* misspelled by *paperclip*). Misspellings caused by cognitive misconceptions tend to persistently appear in the student's writing, whereas typographical errors, such as injecting an extra character or mistakenly typing a neighboring character, occur in other places less frequently. To address this challenge, we implement a two-step writing assessment system, in which the system first creates a dictionary to convert similar words to the same representative word using Levenshtein edit distance and then trains classifiers based on a bag-of-words representation based on the induced dictionary.

For computational writing assessment models, we utilize feedforward neural networks. Deep neural networks, often called *deep learning* [19], have demonstrated considerable success for a wide range of computational challenges, such as computer vision, natural language processing, and speech recognition. A model is trained per short-answer question. Since every writing question has multiple labels (i.e., competencies) to predict, this task is cast as multi-label classification. The hyperparameters for neural networks are often empirically determined using grid search [14]. In this work, we explore the number of hidden units using 256 and 512, and the number of hidden layers from 1 to 4. We fix the following parameters: setting all the activation functions to sigmoid, adopting the dropout regularization technique [19] with the dropout rate of 0.5, and using binary cross entropy and stochastic optimization for the loss function and optimizer, respectively.

## 6 Evaluation

To evaluate the assessment models, we conducted validation studies with the corpus of fourth grade writings and drawings collected with the LEONARDO system. The drawing models were assessed using 4 drawings each from the 98 students scored by human coders. For each drawing, rules mapping between topological features and competency scores were authored based on notes from the rubrics used by human scorers and from tuning on a scored set of drawings not used in the evaluation sample. As the drawing models used authored rules and were not machine-learned, cross validation was not used. The baseline accuracy rate is calculated by computing the most common class rate per competency, and then averaging across all the competencies within each question (Table 2).

**Table 2.** Automated drawing assessment results (N = 98)

Question	Concepts	Accuracy	Baseline
FQ3 – drawing 1	11	90.4%	66.8%
FQ3 – drawing 2	13	87.9%	62.8%
FQ5 – drawing 1	12	86.3%	61.8%
FQ5 – drawing 2	13	90.8%	61.6%

As shown in the table, the models performed well compared to the baseline. Analysis of the classification errors showed a small number of cases where the automated model incorporated elements that were occluded from the drawing presented to human coders. The majority of the error cases were the result of the system not giving credit for a concept for which the human coders gave credit. These types of errors could be potentially corrected by creating more scoring rules, though many would be difficult to author without incurring an unacceptable level of false positives.

For the writings, four feedforward networks were trained for each of the four questions (2 for FQ-3 and 2 for FQ-5), adjusting the number of hidden layers from one to four. Each model was evaluated using a 10-fold student-level cross validation. The accuracy levels shown in Table 3 represent the average accuracy rates across all competencies for the question. The baseline accuracy rate was calculated using the same process as for the drawings. The accuracy rate of neural networks that achieve the highest predictive performance in the 10-fold cross validation is reported along with the number of hidden layers the models leverage.

**Table 3.** Automated writing assessment results (N = 98)

Question	Hidden layers	Concepts	Accuracy	Baseline
FQ3 – writing 1	1	13	78%	71.4%
FQ3 – writing 2	1	5	73%	62.6%
FQ5 – writing 1	1	3	90%	82%
FQ5 – writing 2	1	7	76%	65.3%

Overall the writing assessment system performed very well with accuracies ranging from 73% to 90% for the 4 questions. While the shallow networks exhibited the best overall performance for each question, the accuracies of the other models performed very similarly, often less than 1% different. This result is perhaps not surprising given that deep networks are more likely to suffer from overfitting when trained with small datasets [19]. The high baseline accuracy for the first writing sample in FQ-5 suggests that that question in particular may have been over-scaffolded and should be revised in future implementations. Further analysis of the errors reveals the majority are likely due to the high level of misspellings and grammatical errors in the text, indicating that while the steps taken to cope with noisy text were effective, there is room for improvement.

## 7 Conclusions and Future Work

Multimodal assessments that operate on both student drawing and student writing hold great potential for expanding the diagnostic power of ITSs. ECD provides a unifying framework for multimodal assessment by defining targeted learning concepts of a given exercise, and for identifying evidence of those concepts in student work that includes both drawing and writing. We hypothesized that a unified ECD-based multimodal assessment framework would support the design of computational models of assessment that could operate on both drawing and writing.

In this paper, we introduced a framework for applying ECD to multimodal student learning. First, a competency model is defined, identifying scientific concepts of interest. Next, rubrics are created to define which features of student writings and drawings constitute evidence of the previously defined competencies. Using the rubrics we then found that the evidence measured from both drawing and writing were significantly predictive of performance on a multiple-choice summative post-test. We also found that students were generally able to express more concepts through drawing than writing, although this could be related to the inherent scaffolding afforded by the symbolic drawing tasks. Finally, with the long-term goal of incorporating automated multimodal assessments into interactive learning environments such as the LEONARDO digital science notebook, we developed computational methods for the real-time automated assessment of student drawing and writing artifacts. An evaluation of the resulting multimodal assessment framework found that the models outperformed baseline models in accurately assessing student work across multi-faceted rubrics for both modalities.

In future work it will be important to further refine the automated assessment techniques to increase their accuracy. A second promising line of investigation is to use ECD to better understand how student knowledge of low-level concepts relates to higher-order concepts. Finally, it will be important to investigate how to best incorporate multimodal assessment into an ITS and utilize real-time assessment results to drive personalized feedback and scaffolding.

**Acknowledgements.** This work is supported in part by the National Science Foundation through Grant No. DRL-1020229 and the Social Sciences and Humanities Research Council of Canada. Any opinions, findings, conclusions, or recommendations expressed in this report are those of the authors, and do not necessarily represent the official views, opinions, or policy of the National Science Foundation or the Social Sciences and Humanities Research Council of Canada.

## References

1. VanLehn, K.: The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educ. Psychol.* **46**, 197–221 (2011)
2. Shute, V.J., Kim, Y.J.: Formative and stealth assessment. In: Spector, J.M., Merrill, M.D., Elen, J., Bishop, M.J. (eds.) *Handbook of Research on Educational Communications and Technology*, pp. 311–321. Springer, New York (2014)
3. Bennett, R.E.: Formative assessment: a critical review. *Assess. Educ. Principles, Policy Pract.* **18**, 5–25 (2011)
4. Desmarais, M.C., Baker, R.S.J.D.: A review of recent advances in learner and skill modeling in intelligent learning environments. *User Model. User Adap. Interact.* **22**, 9–38 (2011)
5. Minogue, J., Wiebe, E., Bedward, J., Carter, M.: The intersection of science notebooks, graphics, and inquiry. *Sci. Child.* **48**, 52–55 (2010)
6. Mislevy, R.J., Haertel, G.D.: Implications of evidence-centered design for educational testing. *Educ. Measur. Issues Pract.* **25**, 6–20 (2006)
7. Nelligan, T., Helms, M., Polsley, S., Linsey, J., Ray, J., Hammond, T.: *Mechanix*: a sketch-based educational interface. In: *Proceedings of the 20th International Conference on Intelligent User Interfaces*, pp. 53–56. ACM (2015)
8. Bollen, L., van Joolingen, W.: SimSketch: multi-agent simulations based on learner-created sketches for early science education. *IEEE Trans. Learn. Technol.* **6**, 208–216 (2013)
9. Jee, B.D., Gentner, D., Uttal, D.H., Sageman, B., Forbus, K., Manduca, C.A., Ormand, C.J., Shipley, T.F., Tikoff, B.: Drawing on experience: how domain knowledge is reflected in sketches of scientific structures and processes. *Res. Sci. Educ.* **44**, 859–883 (2014)
10. Burrows, S., Gurevych, I., Stein, B.: The eras and trends of automatic short answer grading. *Int. J. Artif. Intell. Educ.* **25**, 60–117 (2014)
11. Graesser, A.: Using latent semantic analysis to evaluate the contributions of students in autotutor. *Interact. Learn. Environ.* **8**, 1–33 (2000)
12. Sukkariéh, J., Blackmore, J.: C-rater: automatic content scoring for short constructed responses. In: *Proceedings of the 22nd International FLAIRS Conference*, pp. 290–295 (2009)
13. Dzikovska, M., Nielsen, R., Brew, C.: Towards effective tutorial feedback for explanation questions: a dataset and baselines. In: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Montreal, Canada, pp. 200–210 (2012)
14. Leeman-Munk, S., Smith, A., Mott, B., Wiebe, E., Lester, J.: Two modes are better than one: a multimodal assessment framework integrating student writing and drawing. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M. (eds.) *AIED 2015. LNCS*, vol. 9112, pp. 205–215. Springer, Heidelberg (2015)
15. Gobert, J.D., Pedro, M.A.S., Baker, R.S.J.D., Toto, E., Montalvo, O.: Leveraging educational data mining for real-time performance assessment of scientific inquiry skills within microworlds. *J. Educ. Data Min.* **4**, 111–143 (2012)

16. Rupp, A., Levy, R., Dicerbo, K.E., Sweet, S.J., Crawford, A.V., Calico, T., Benson, M., Fay, D., Kunze, K.L., Mislevy, R.J., Behrens, J.: Putting ECD into practice: the interplay of theory and data in evidence models within a digital learning environment. *J. Educ. Data Min.* **4**, 49–110 (2012)
17. Min, W., Frankosky, M.H., Mott, B.W., Rowe, J.P., Wiebe, E., Boyer, K.E., Lester, J.C.: DeepStealth: leveraging deep learning models for stealth assessment in game-based learning environments. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M. (eds.) *AIED 2015*. LNCS, vol. 9112, pp. 277–286. Springer, Heidelberg (2015)
18. Mislevy, R.J., Almond, R.G., Lukas, J.F.: A brief introduction to evidence-centered design. *ETS Research Report Series*, 16 (2003)
19. Lecun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**, 436–444 (2015)



# The Bright and Dark Sides of Gamification

Fernando R.H. Andrade<sup>1</sup>, Riichiro Mizoguchi<sup>2</sup>, and Seiji Isotani<sup>1</sup>(✉)

<sup>1</sup> ICMC, University of São Paulo, São Carlos, SP, Brazil  
{Fernando.heb, sisotani}@icmc.usp.br

<sup>2</sup> Japan Advanced Institute of Science and Technology, Ishikawa, Japan  
mizo@jaist.ac.jp

**Abstract.** Everything in life has a bright and a dark side; and gamification is not an exception. Although there is an increasing number of publications discussing the benefits of gamification in learning environments, i.e. looking into the bright side of it, several issues can hinder learning because of gamification. Nevertheless, it seems that only few researchers are discussing the dark side of using gamification in learning environments and how to overcome it. Thus, in this paper, we discuss some of the problems of gamification, namely, addiction, undesired competition, and off-task behavior. Furthermore, to deal with both bright and dark sides of gamification at the same time, we propose a framework for intelligent gamification (FIG) that can offer the necessary infrastructure for ITS to personalize the use of gamification by monitoring risk behavior, exploring how best use game design elements to avoid their overuse and finally supporting “fading” mechanisms that gradually reduces the use of gamification and help students to concentrate on learning and not only on extrinsic motivators.

**Keywords:** Gamification · Intelligent tutoring systems · Addiction · Framework

## 1 Introduction

Everything has a bright side and a dark side like a coin, which has a head and a tail, and Gamification is not an exception. Usually when people find a good thing, they tend to focus only on its bright side. However, they should always be aware of its dark side, to use it appropriately.

In the past few years, Gamification has been drawing attention from different areas, with the promise of increasing users’ engagement, motivation, and promoting changes in behavior [8]. By introducing mechanics and elements from games, several companies and research groups have been trying to increase learners’ performance, communication between different groups of people, and promote better health care and healthy habits [1]. Specifically, in the educational field, several studies have been studying different techniques and benefits of using gamification to raise students’ engagement level and reach the flow state with significant findings [2–4].

Although several positive effects of using gamification has been found to date, particularly to improve student’s performance and increase engagement [8], researchers and educators are ambivalent about using game like materials in education since they

could cause addiction and increase the externalization of behaviors that can hinder learning [5, 6].

This fear should be taken seriously since many recent empirical research reports the benefits of gamification as unexpected side effects, and not as a result of a well-thought-out design [1, 3, 4]. It shows that the gamification implementation techniques are still unconsolidated. Yet, according to two literature reviews on the topic, there are no studies addressing the potential negative effects of gamification in Intelligent Tutoring System (ITS) or any other kind of Virtual Learning Environment (VLE) [1, 7].

Thus, the main goal of this work is to discuss the potential harms of using game elements in an ITS and propose a general framework to use gamification in an intelligent way. Considering positive and negative aspects and suggesting ways to fade the gamification elements to cope with addiction/dependence on gamification.

The remainder of this paper is structured as follows: Sect. 2 describes the related works. Section 3 discusses the dark side of gamification and the proposed framework. Section 4 presents our envisioning application of the framework and how to use it. Section 5 concludes the paper with our final thoughts and the directions towards the validation of our Framework for Intelligent Gamification (FIG).

## 2 Background: Gamification, Flow and Addiction

Kapp [8] defines gamification as “*Using game-based mechanics, aesthetics and game thinking to engage people, motivate action, promote learning, and solve problems*”. The definition of the concept changes slightly according to different authors, but the core idea remains the same, that is, gamification as a tool to “increase engagement in some activity using game features, providing enjoyment and fun” [1–3, 9, 10].

The motivational background of gamification usually relies on the SDT (Self Determination Theory) [11], which considers that a human being has three basic needs: competence, relatedness, and autonomy. Based on the degree of a person’s needs and the kind of activity, he/she can be more or less motivated to perform some activity. According to this theory, the user levels of motivation [11], vary from amotivated (without any motivation to perform the activity) to intrinsically motivated (when the user doesn’t need any external incentive to perform it). Thus, the gamification theory proposes that by introducing game elements in an environment to satisfy some of the user’s needs, it is possible to make the activities more attractive, even if he/she is not intrinsically motivated.

The most common game mechanic applied in educational environments is the reward system based on fast feedback about the students’ performance in the form of points, trophies and badges and the division of the domain content in small units representing game levels [12, 13]. Furthermore, the use of leaderboards is also a common tool to stimulate competition [1, 14].

One of the main goals of using gamification is to keep users in *flow*. The flow is a state of deep concentration in which the user becomes so engaged in the task that he/she loses self-awareness, and track of time [15]. Also known as optimum experience flow;

a highly desired state by game developers, considering that they want to keep the player entertained and engaged as much as possible.

The idea of using gamification in learning environments to put students in a flow state while they are learning is quite attractive to be implemented [2, 16]. On the other hand, a number of studies has been conducted addressing the flow state as a factor associated to game addiction. For example, Sun [17] conducted a research with 234 users, in which they found evidences that associate addiction in mobile games with perceived visibility and flow. Perceived visibility is related to the notion of being noticed by peers and in a position of social presence. Gamification designers also seek to incorporate this characteristic in the systems, by using leaderboards and sharing user achievements, thus fulfilling the relatedness needs of the students according to the SDT. In another study, Jeong and Lee [6] examined whether Big Five personality traits can affect game addiction according to psychological, social, and demographic factors. To do so, the researchers used data from a survey of 789 game users in Korea, seeking associations and the results showed that the neuroticism trait apparently increases game addiction. They also observed that a general self-efficacy affected game addiction in a negative way, whereas game self-efficacy increased the degree of game addiction. Besides that, loneliness enhanced game addiction, while depression showed a negative effect on the addiction. In the context of education, these findings could mean that a student who is confident in his abilities to perform the task is less prone to addiction than a student without confidence, and if the student only has confidence in his game skills, he is more susceptible to addiction.

### 3 The Dark Side of Gamification

The gamification approach originates in the industry with a strong appeal from marketing and service [9]. In the context of learning, to increase students' engagement researchers and professionals have been trying to bring flow experience and immersion to VLE. Even though improving learners' engagement using game elements is a highly attractive idea, contrary to the marketing perspective, the goal is not to make the student loyal to the system, but rather increase his learning.

Therefore, we believe that gamification can be good, as long as it is controlled and monitored. If such measures are not taken, then this could adversely affect the effectiveness of the system and hinder learning. In the following paragraphs, we will present three problems that may appear by adding game elements and mechanics without careful considerations:

**Off-Task Behavior:** If the gamification system is untied to the educational outcomes, the game features can be a distraction to the user. In this case, even if the user likes to use the system, he will not learn more from it. For example, the introduction of resources that provides relatedness to users, such as chats and forums. These resources are not directly related to the learning experience, allowing to the student to spend time in the system without focusing on learning. Another example are the customization features, those are a very important to promote immersion, but also, allows spend time in the system without learning.

**Undesired Competition:** Leaderboards are a common resource to promote competition, and sense of competence. Still, it can be harmful for students with low performance and low self-efficacy, since they can feel forced in a competition with their peers, which can negatively affect their sense of competence and result in the reduction of their interest and engagement.

**Addiction and Dependence:** Based on the literature [6, 17, 18], some game features and sensations like flow can be regarded as addictive factors. Thus, addiction could be a potential problem in gamified environments. Unlike the behavior of alcoholics or gambling addicts, addiction in such environments should not have greater effects such as loss of personal property or family disruption. However, our concern is the kind of dependency created by the game-like experience in education, as the students can resource to “game the system” in order to get rewards or they may not be able to learn without gamification features.

In the first scenario, the student could change the focus from learning the subject to other aspects provided by the system gamification. For instance, earning points to get a higher position in a leaderboard or unlock one exclusive or rare content in the system and gain visibility with his peers. Typically, high positions in ranks or acquisition of virtual goods in a gamified application depends on the progress of the system main objective, but it is not uncommon for students to seek alternative strategies to get their desired results [19]. In the second case, the student creates a dependence of game elements to stay engaged in the system. In other words, the student is only capable to focus on the system and acquire some knowledge if it has game elements or some kind of extrinsic reward for his effort. To identify this condition, the system demands information about the relationship of the student with the game elements.

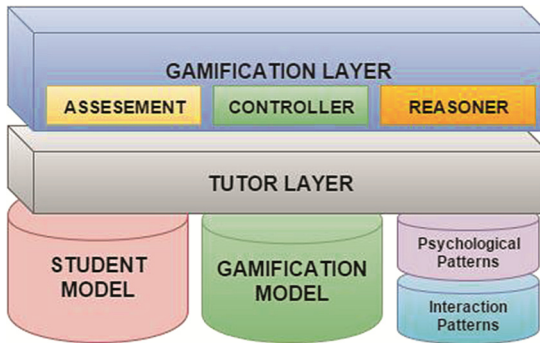
Since the evolution in the gamification in a well-designed system is highly correlated to the success and the learning outcomes, the gamification overuse may go unnoticed; therefore, a constant monitoring of the interactions between the user, the system and the gamification features is required.

## 4 Framework for Intelligent Gamification (FIG)

There are few initiatives towards gamification taken by academics aiming at the improvement and the consolidation of gamification. Previous works on gamification have proposed frameworks with different perspectives, but to our knowledge none of these have discussed how to deal with the negative implications of gamification [20, 21]. However, as discussed before it is crucial to deal with both sides of gamification, not only using its potential to increase the engagement, but also controlling this use of gamification to avoid the creation of new problems.

In order to address this, we propose a framework based on the ITS architecture that considers the information required to implement gamification with personalization and can process its impacts on the students and potential harms. Further, we propose a strategy to reduce the participation of overused elements by fading. Thereby, our framework proposes to increase the engagement aligning the gamification strategies to gamer profiles and also to identify and handle misuses resultant from the gamification in

learning environments, which, for the best of our knowledge, was not addressed by neither the academic community nor the industry. In Fig. 1, we present the proposed framework and its components, which are explained in the following subsections.



**Fig. 1.** Framework for intelligent gamification.

**Gamification Layer.** In this work, we are not approaching the domain content gamification, in this sense the gamification in this framework is a layer independent of the pedagogical objectives proposed by the tutor, allowing dynamical customization. Once it interacts with the student in order to satisfy the motivational needs of competence, relatedness and autonomy, but do not change the pedagogical objectives proposed by the learning designer. Currently, most of the studies only use static elements without or with at least few personalization options, however, the game design literature and also the results of empirical studies provide evidences indicating the need to consider user individual preferences [1, 10].

### Data Modules

- (a) **Gamification Model.** A game element can be considered as a game component, it will behave according to the game mechanic attached to it, and will interact with the user when a game event is triggered due an action taken by him [2]. The gamification model contains all the possible game events that can be triggered in the system and that are controlled and regulated by the Controller Component.
- (b) **Student Model.** The main goal of gamification is to affect the students' motivation and behavior. In order to do so by using an intelligent approach, it is necessary to hold enough information in the Student Model. Thus, we propose a student model divided in five small groups of attributes, as presented in Fig. 2 and explained in the subsequent item.

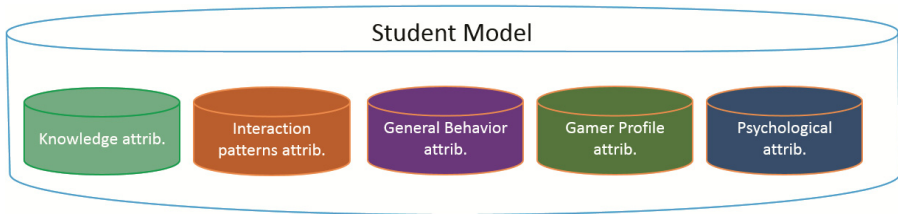


Fig. 2. Student model.

- (b.1) *Knowledge Attributes.* This group contains the traditional information of the Student Model in terms of domain knowledge or skills they learn. There are several ways of representing students data regarding the information used by the ITS Tutor Module to make decisions in order to provide a better quality of content and hints. Thus, it is not in the scope of this study to address the way of representing these data. However, it is important to clarify that there is indeed a need for data on the student's performance, so the knowledge base should be able to provide these data to considerations about improvement or decreasing of student performance.
- (b.2) *Psychological Attributes.* It contains information about the student's personality traits and data on mood. As said in the previous sections, several studies shown that the personality traits influence learning and addiction behaviors, in this sense, the information about the students' personality trait is a useful tool to provide evidences of an undesirable condition.
- (b.3) *General behavior Attributes.* They are responsible for storing information about the student's habits not related to learning. Game addiction shares several symptoms and characteristics with different kinds of addictions, so it is necessary to expand the knowledge about the user in order to obtain evidences of a problem cause-effect relation.
- (b.4) *Interactions Patterns Attributes.* The system logs record the session length, dates, time between tasks, estimated required time to finish that tasks and the information about the interaction with the game elements. Therefore, the interaction patterns attributes contain the analyzes of those information such as mean of interactions during sections, number of tasks performed by section, mean time to solve tasks, frequent subjects, total amount of logins, mean length of the sessions.
- (b.5) *Gamer Profile Attributes.* In this framework, we are considering that students may have different gaming habits and preferences in order to provide a suitable set of game elements and mechanics.
- (c) **Interaction Patterns.** The interaction patterns contain the representation of an expected behavior in the system. This model represents the observable data such as time to finish contents, number of interactions, and frequency of system use. The interaction patterns also contain the model of expected interactions with the game elements. This model will vary according to the gamer profile approach and the

gamification model, since it has to represent the regular interaction pattern for a student in the case of static gamification model, or the standards for a group of students in the case of a gamification model based on different profiles.

- (d) **Psychological Patterns.** The psychological patterns represent the information that, when matched with the situation of one student, provide evidence that this student may be in a risk group. It can be represented by a set of rules, preset by experts or by a series of factors that can be used by the Reasoner to inference about the student situation.

### Operational Modules

- (a) **Assessment Component.** The assessment component is responsible for collecting the student's observable and interactive data and update student model.
- (b) **Behavior Reasoner.** The Behavior Reasoner is the component responsible for analyzing the student's data in order to identify risk behavior. To perform this task, the component compares the information contained in the Student Model with the standards model in the Interaction Patterns and Psychological Patterns. When it identifies anomalies in the student behavior, the Reasoner may inform a human system administrator, such as a teacher, to take an action or, as we propose in this paper, to inform the situation to the Controller, triggering changes in the gamification layer.
- (c) **Controller Component.** The Controller is the component is responsible for the settings of the gamification layer, and in order to do so, the controller needs to cross the information contained in the student model, gamification model, and behavior Reasoner component. In a customizable approach, the student would be able to interact with the controller, changing the suggested gamification components or parameters and, at the same time, giving information to the controller, which will change the student's gamer profile attributes, if needed. When the Reasoner identifies that a user needs to change his interactions with some elements, the controller may act changing the value attributed to that element in order to fade this element for the user interest. Our definition of Fade represents the change in the attributes of the element in order to make it less attractive or difficult to access, like changing its colors or moving it to an area that receives less attention.

## 5 Envisioned Application

### 5.1 Information Gathering

- **Gamer Profile:** To model the gamer profile, there are several player types in game design literature and some new types are proposed considering gamification applications [22, 23]. The game components in the system have to be consistent with the player/user types in the chosen typology. The gamer profile is composed of player type attributes and the values for each of these attributes are updated by the controller according to the interaction patterns to personalize the gamification and fading for that specific player.

- **Psychological Attributes:** Two very common tools for data acquisition about personality traits are the Big Five [24] and the MBTI (Mayers Briggs Types Indicator) [25]. However, several researches criticized the use of MBTI as a psychometric instrument. Our model is composed of the personality traits, and can contain other psychological variables that may be used to identify anomalies in the user behavior. For instance, history of mood changes and history of emotions.
- **General Behavior Attributes:** The function of this model is to store complementary information about user habits. To this effect, the use of intelligent agents or chatbots is highly recommended. Such agents can also be used to acquire information about mood modifications and other behavioral attributes.
- **Gamification Model:** Each gamer profile has a list of adaptation attributes that correspond to the game components that will be available to that specific profile in the interface. Each attribute can receive a value between 0 (inactive) and 1 (fully active). The Gamification Model contains the standards for these values, and changing these attributes affects the standards for the player types.
- **Interaction Patterns:** Normal user behavior can be established by experts, pilot running of the system or by the behavior of the majority of the users in the system.
- **Psychological Pattern:** The psychological pattern represents the risk group in the system. In this sense, this model has to contemplate the traits, and the associations with other variables that provide evidences of a risk scenario. E.g. One student that has the trait of irresponsibility, but solves a number of tasks above the mean of the other students, in a much shorter time than the required, should be considered as a candidate for change.

## 5.2 Operation

Initially the student provides information about his gamer profile and personality traits. After that, the Controller consults the gamification model and adapts the interface to the elements recommended for the student. Then, the assessment component starts to log the user's interactions and the intelligent agent interacts with the student in predetermined intervals to fill the general behavior model. Once the general behavior model is populated, the Reasoner starts to compare the patterns periodically, in order to identify anomalies.

As the Reasoner becomes more knowledgeable about the anomalies in the student interaction patterns, it generates a list of gamification artifacts<sup>1</sup> eligible for fading. To maximize the learners growth capabilities, the fading method has been previously used to minimize user's reliance on the system's help [26]. When an artifact hits the predetermined threshold, the Reasoner marks it for the fading process. Once the process starts, the system agent makes an intervention signaling the excess of interactions with that artifact and tracks the user performance and interactions seeking changes in his behavior. This intervention intends to increase his self-awareness and provides the opportunity for self-regulation. However, if after a certain period the behavior remains same, the

---

<sup>1</sup> A Gamification artifact is defined as a composition of a visual game element, that directly interacts with the user, and the game mechanic, that define how this element will behave.



system starts to fade away the artifact, up to removal, until the number of interactions go back to normal. After that, the artifact is restored to the original state and the agent informs the student to observe his behavior.

To identify the implications of fading on the user performance and how much he depends of gamification to keep motivated, the student is constantly monitored. If during the fading process the student's performance declines, the agent makes an intervention in order to find out whether this is due to fading the artifact. If the reason for the decline is inherent to the process, it provides evidence with respect to the student's dependence on gamification. Nevertheless, in both cases, the element is restored to the original state and the agent informs the user about the importance of keeping focus on learning. The artifact is restored so as not to impair their learning. Furthermore, the intervention will reinforce his self-awareness and provide, once more, the opportunity for self-regulation, which we believe could be more meaningful since the user knows that he can be "punished" somehow for his overuse.

## 6 Concluding Remarks

Most of the time people tend to focus too much on the bright side and overlook the dark side of matters. Similarly, the interest in gamification has been growing; however, no one seems to have shown interest in its dark side (negative effects). In this paper, we identified addiction as the dark side of gamification and addressed the elements used in gamification that related to this phenomenon and how it occurs in gamified environments. Further, we proposed a framework to monitor and fade with the gamification elements to avoid the negative implications of addiction.

Our next steps include providing a detailed addiction model for learning environments and the experimental evaluation of the fading strategy of gamification elements and the impact of this strategy in terms of engagement and performance.

The ITS architecture was chosen because such systems consider student information to make decisions in order to improve learning. However, we believe that the same reasoning can be applied to any VLE with proper dynamics to interact and retain enough information about the student and the environment.

**Acknowledgments.** We thank CNPq and CAPES for supporting this research.

## References

1. Hamari, J., Koivisto, J., Sarsa, H.: Does gamification work? A literature review of empirical studies on gamification. In: Proceedings of Hawaii International Conference on System Science, pp. 3025–3034 (2014)
2. Chalco, G.C., Andrade, F.R.H., Oliveira, T., Isotani, S.: Towards an ontological model to apply gamification as persuasive technology in collaborative learning scenarios. In: Proceedings of the Simpósio Brasileiro de Informática na Educação, pp. 499–508 (2015)

3. Pedro, L.Z., Lopes, A.M.Z., Prates, B.G., Vassileva, J., Isotani, S.: Does gamification work for boys and girls? An exploratory study with a virtual learning environment. In: Proceedings of the ACM Symposium on Applied Computing, pp. 214–219 (2015)
4. De-Marcos, L., Domínguez, A., Saenz-de-Navarrete, J., Pagés, C.: An empirical study comparing gamification and social networking on e-learning. *Comput. Educ.* **75**, 82–91 (2014)
5. Schmitt, Z.L., Livingston, M.G.: Video game addiction and college performance among males: results from a 1 year longitudinal study. *CyberPsychology Behav. Soc. Netw.* **18**, 25–29 (2015)
6. Jeong, E.J., Lee, H.R.: Addictive use due to personality: focused on big five personality traits and game addiction. *Int. J. Soc. Behav. Educ. Econ. Bus. Ind. Eng.* **9**(6), 1995–1999 (2015)
7. Borges, S., Reis, H.M., Durelli, V., Isotani, S.: A systematic mapping on gamification applied to education. In: Proceedings of the ACM Symposium on Applied Computing, pp. 216–222 (2014)
8. Kapp, K.M.: *The Gamification of Learning and Instruction: Game-Based Methods and Strategies for Training and Education*. Pfeiffer, San Francisco (2012)
9. Huotari, K., Hamari, J.: Defining gamification - a service marketing perspective. In: Proceedings of the International Academic MindTrek Conference, pp. 17–22 (2012)
10. Monterrat, B., Lavoué, É., George, S.: A framework to adapt gamification in learning environments. In: Proceedings of the European Conference on Technology Enhanced Learning, pp. 578–579 (2014)
11. Ryan, R., Deci, E.: Intrinsic and extrinsic motivations: classic definitions and new directions. *Contemp. Educ. Psychol.* **25**, 54–67 (2000)
12. Aparicio, A.F., Vela, F.L.G., Sánchez, J.L.G., Montes, J.L.I.: Analysis and application of gamification. In: International Conference on Interacción Persona-Ordenador, pp. 1–2 (2012)
13. Maragos, K., Grigoriadou, M.: Towards the design of intelligent educational gaming systems. In: Proceedings of the AIED Workshops, pp. 35–38 (2005)
14. Nah, F.F.-H., Eschenbrenner, B., DeWester, D., Park, S.R.: Impact of flow and brand equity in 3D virtual worlds. *J. Database Manag.* **21**, 69–89 (2010)
15. Csikszentmihalyi, M.: *Flow: The Psychology of Optimal Experience*. Harper Perennial Modern Classics, New York (2008)
16. Chalco, G.C., Andrade, F.R.H., Borges, S.S., Bittencourt, I.I., Isotani, S.: Toward a unified modeling of learner’s growth process and flow theory. *Educ. Technol. Soc.* **19**(2), 1–14 (2016)
17. Sun, Y.Y., Zhao, Y., Jia, S., Zheng, D.: Understanding the antecedents of mobile game addiction: the roles of perceived visibility, perceived enjoyment and flow. In: Proceedings of the Pacific Asia Conference on Information Systems, p. 141 (2015)
18. Chou, T., Ting, C.: The role of flow experience in cyber-game addiction. *CyberPsychology Behav.* **6**(6), 663–675 (2004)
19. Baker, R.S.J., Walonoski, J., Heffernan, N., et al.: Why students engage in “gaming the system” behavior in interactive learning environments. *J. Interact. Learn. Res.* **19**(2), 185–224 (2008)
20. Wongso, O., Rosmansyah, Y., Bandung, Y.: Gamification framework model, based on social engagement in e-learning 2.0. In: International Conference on Technology, Informatics, Management, Engineering, and Environment, pp. 10–14 (2014)
21. Simões, J., Redondo, R.D., Vilas, A.F.: A social gamification framework for a K-6 learning platform. *Comput. Hum. Behav.* **29**, 345–353 (2013)
22. Yee, N.: Motivations for play in online games. *CyberPsych Behav.* **9**, 772–775 (2006)

23. Nacke, L.E., Bateman, C., Mandryk, R.L.: BrainHex: preliminary results from a neurobiological gamer typology survey. In: Proceedings of International Conference on Entertainment Computing, pp. 288–293 (2011)
24. Barrick, M.R., Mount, M.K.: The big five personality dimensions and job performance: a meta-analysis. *Pers. Psychol.* **44**(1), 1–26 (1991)
25. Myers, I.B., McCaulley, M.H., Quenk, N.L., Hammer, A.L.: *MBTI Manual: A Guide to the Development and Use of the Myers-Briggs Type Indicator*, 3rd edn. Consulting Psychologists Press, Palo Alto (1998)
26. Ueno, M., Miyasawa, Y.: Probability based scaffolding system with fading. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M.F. (eds.) *AIED 2015. LNCS*, vol. 9112, pp. 492–503. Springer, Heidelberg (2015)

# Behavior Changes Across Time and Between Populations in Open-Ended Learning Environments

Brian Gauch<sup>(✉)</sup> and Gautam Biswas

Department of Electrical Engineering and Computer Science,  
Institute of Software Integrated Systems, Vanderbilt University,  
1025 16th Avenue South, Nashville, TN 37212, USA  
{brian.gauch, gautam.biswas}@vanderbilt.edu

**Abstract.** Open-ended computer-based learning environments (OELEs) can be powerful learning tools in that they help students develop effective self-regulated learning (SRL) and problem solving skills. In this study, middle school students used the *SimSelf* OELE to build causal models to learn about climate science. We study their learning and model building approaches by calculating a suite of behavioral metrics derived using *coherence analysis* (CA) that are used as features on which to group students by their type of learning behavior. We also analyze changes in these metrics over time, and compare these results to results from other studies with a different OELE to see determine generalizable their findings are across different OELE systems.

**Keywords:** Open-ended learning environments · Coherence analysis · Self-regulated learning · Temporal analysis

## 1 Introduction

Open-ended computer-based learning environments (OELEs) [1, 2] are learner-centered. Rather than being prescriptive and narrowly focused, they present students with an environment that supports and encourages exploration while solving challenging problems by harnessing resources and tools provided to construct and verify problem solutions. OELEs support a constructivist approach to learning, one that is founded on the tenants that we learn through interactions with our environment and that cognitive conflict or puzzlement is a stimulus for learning [3]. The notion of “cognitive conflict” implies that OELEs place significant cognitive demands on learners. To solve the overall problem, students must simultaneously foster their emerging understanding of a complex topic, improve their skills to support their learning, and employ *self-regulated learning* (SRL) processes to succeed on open-ended tasks [4].

SRL is a theory of learning that describes how learners take control of, evaluate, and reflect on their own learning performance and behaviors [5]. Broadly speaking, it refers to learning that is guided by metacognition (thinking about one’s thinking), strategic action (planning, monitoring, and evaluating personal progress against a standard), and motivation to learn [6]. By logging student activities as they learn,

OELs can gather information that reveals students' understanding of: (i) the problem domain; (ii) the problem-solving task; and (iii) strategies they employ for solving the problem.

In this paper, we use a form of behavior analysis called *coherence analysis* [7] to study students' problem-solving in an open-ended computer-based learning environment (OELE) called *SimSelf*. CA analyzes learner behaviors within an OELE with respect to their demonstrated abilities to seek, interpret, and apply information to building and verifying models of science phenomena. We use the CA features computed over data collected from the period of the study to discover groups of student behaviors using an unsupervised learning method, and interpret these groups using their overall learning and model building characteristics. We then apply correlation measures to link behaviors to pre- to post-test learning gains, and their model building performance. Last, we perform a day-by-day analysis to study how students' learning behaviors change over time as they work in the *SimSelf* environment.

## 2 Background

To analyze and support students SRL, the OELE must assess learners' skill proficiencies, interpret their actions in terms of goals and learning strategies, and evaluate their success in accomplishing their tasks. Many aspects of self-regulation are difficult to capture and even more difficult to quantify. Analysis may be further complicated by the fact that students may modify or switch between approaches as they develop their own problem-solving skills.

Despite this complexity, researchers have developed approaches to measuring aspects of self-regulation in OELEs. In Crystal Island [8] and EcoMUVE [9], students' logged activities were manually coded with higher-level codes, and then used to construct predictive models. In Crystal Island, Sabourin et al. [8] also asked students to enter "status updates" at regular intervals while they worked on the system. These manual updates were easier to interpret than trying to infer progress from the raw recorded user activities. Snow et al. [10] embedded theory-driven models of SRL into iSTART-ME, a game-based learning environment to help students improve their science comprehension. They calculated the Shannon Entropy and interpreted lower entropy as indicative of ordered and self-regulated behaviors. Similarly, EcoLab [11] measured students' metacognitive awareness of their own ability by comparing the system's assessment of students' ability levels with the difficulty of the activities they choose to pursue.

## 3 Method

In our work, we use Coherence analysis (CA), a theory-driven metric that combines information across sequences of actions together to calculate *action coherence* [7]. CA is quite general, and should be relevant to any OELE where a student (1) can take actions to gain information and (2) can take actions which demonstrate learning of said information.

Two ordered actions ( $x \rightarrow y$ ) taken by a student in an OELE are **action coherent** if the second action,  $y$ , is based on information generated by the first action,  $x$ . In this case,  $x$  provides support for  $y$ , and  $y$  is supported by  $x$ . Should a learner execute  $x$  without subsequently executing  $y$ , the learner has created **unused potential** in relation to  $y$ . Note that actions  $x$  and  $y$  need not be consecutive.

CA interprets students' behavior in terms of the information they encounter and whether or not this information is used during subsequent actions. CA assumes that learners with higher levels of action coherence possess stronger metacognitive knowledge and task understanding. Thus, these learners will perform a larger proportion of supported actions and take advantage of a larger proportion of the potential that their actions generate. In this paper, we incorporated the following six CA metrics (identical to those used in [12]):

- *Edit frequency*: The number of causal link edits/annotations per minute made by the student
- *Unsupported edit percentage*: The percentage of unsupported causal link edits/annotations not supported by previous views within five minutes
- *Information viewing percentage*: The percentage of time the student spent viewing resources and quiz results
- *Potential generation percentage*: The percentage of time spent viewing information that could support correct causal map edits
- *Used potential percentage*: The percentage of potential generation time associated with views that lead to correct causal map edits within five minutes
- *Disengaged percentage*: The percentage of five minute time periods during which the student neither viewed information nor edited the map

Equivalent metrics could be applied to any OELE where a student (1) can take actions to gain information and (2) makes discrete edits to an artifact which can be linked to gained information.

## 4 SimSelf

*SimSelf* presents students with a complex array of tasks united in the single context of creating correct models of scientific processes [13]. Students demonstrate their learning and understanding by creating a *causal map*, a set of concepts connected by directed links that capture causal relationships among the concepts, such that a chain of causal relations can be used to derive or explain relevant behaviors of the system [14]. At the middle school level, the causal relations are simplified to the qualitative options of increase (+) and decrease (-), e.g., vegetation *decreases* the amount of carbon dioxide in the air. The goal for students using *SimSelf* is to construct causal maps that match hidden, expert models of the domain.

As an OELE, *SimSelf* includes tools for acquiring information, applying that information to the problem-solving context, and assessing the quality of the constructed solution. Students acquire domain knowledge by reading hypertext resources that include descriptions of scientific processes, e.g., the water cycle, and information pertaining to each concept that appears in these processes, e.g., vegetation. As students

read, they need to identify causal relationships between relevant concepts that will appear on their map. Students can then apply this learned information to their maps. Learners can assess their maps by having *SimSelf* automatically reason with the maps to complete quizzes. The software can then grade the generated answers and show how they were derived from the maps. When a student's map answers a quiz question correctly, this indicates that the links used to answer that question are correct. Similarly, if a question is answered incorrectly, this indicates that at least one of the links used to answer the question is incorrect (Fig. 1).

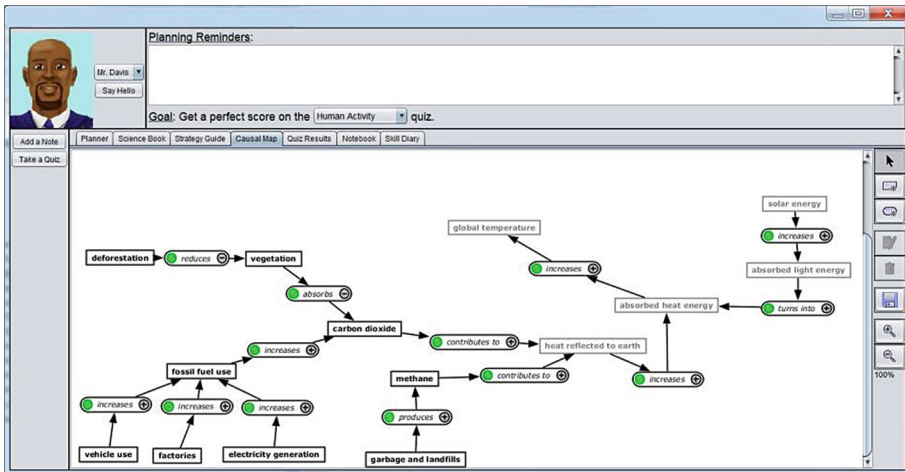


Fig. 1. The *SimSelf* system showing the causal map interface.

## 5 Classroom Study

Data in this paper comes from a recent study of 62 8<sup>th</sup> graders at a Tennessee middle school. Students used the *SimSelf* system for two weeks to learn about climate change. The educational material was presented in two instructional units, with the second building off the first. Two days were spent on training, two days were spent on pre-tests and post-tests (one day each), and two days were spent on each of the two instructional units, for a total of 4 system days. The expert map for the first unit contained 14 concepts and 13 links while the second unit's causal map contained 22 concepts and 25 links. There were 32 science resources pages (4068 words) that, when combined, had a Flesch-Kincaid reading grade level of 7.9.

## 6 Results

Considered as a whole, students performed significantly ( $p = 0.02$ ) better on the science post-test than the pre-test, improving by an average of 2 points, out of a possible 16. In unit 1, they had map scores with an *average* = 8.77 (*standard deviation*

( $sd$ ) = 5.96) out of a possible 13. In unit 2, they continued building on the map from unit 1, after being given the correct 13 links for map 1. They increased their map scores by an *average* of 8.18 ( $sd$  = 4.54) out of a possible 12 (new links).

## 6.1 Behavior Cluster Analysis

A complete-link clustering analysis [15] of the six CA metrics revealed four distinct clusters among the 62 students. Table 1 shows the CA metrics for all clusters. Cluster 1 students ( $n$  = 15) may be characterized as *frequent researchers and careful editors*; these students spent a larger proportion of their time (50.8 %) compared to the other groups viewing resources, and they edited their maps with moderate frequency. However, their edits were more often supported by recent activities than any of the other groups (unsupported edit percentage 54.3 %), and the information they viewed was useful for improving their causal maps (potential generation percentage 56.0 %). Cluster 2 students ( $n$  = 33) may be characterized as ‘*Guess and check*’ experimenters. By far the largest group of learners, these are best characterized by their frequent map edits (0.62 per minute), most of which (76.3 %) were unsupported. They did not view the resources often (38 %), and the potential generated from their actions was low (38.4 %). However, they had the highest map scores of all of the groups.

**Table 1.** Means (and standard deviations) of CA metrics by cluster, from *SimSelf* data

Cluster	Edit Freq.	Unsup. edit %	Info. view %	Potential Gen. %	Used potential %	Disengaged %
s1. Researchers/ Careful editors ( $n$ = 15)	0.55 (0.25)	54.3 % (10.1 %)	50.8 % (7.6 %)	56.0 % (12.5 %)	50.9 % (14.1 %)	18.2 % (4.4 %)
s2. Guess and check experimenters ( $n$ = 33)	0.62 (0.25)	76.3 % (10.9 %)	38.0 % (10.1 %)	38.4 % (16.3 %)	39.9 % (14.8 %)	21.9 % (8.3 %)
s3. Confused guessers ( $n$ = 9)	0.40 (0.21)	86.1 % (9.0 %)	40.9 % (12.2 %)	69.4 % (9.7 %)	9.1 % (7.1 %)	32.9 % (9.5 %)
s4. Disengaged ( $n$ = 5)	0.14 (0.11)	99.5 % (1.1 %)	20.8 % (6.9 %)	29.7 % (25.3 %)	0.26 % (0.58 %)	63.4 % (7.8 %)

*Confused guessers* (Cluster 3;  $n$  = 9) edited their maps fairly infrequently and usually without support (86.1 %). They spent an average of 40.9 % of their time viewing sources of information, and much of their reading did generate potential (potential generation percentage = 69.3 %). Unfortunately, when they did view useful information, they often did not take advantage of it (used potential percentage = 9.1 %), indicating that they may have struggled to understand the relevance of the information they encountered. Students in Cluster 4 ( $n$  = 6) may be characterized as *disengaged from the task*. On average, these students spent more than 63.4 % of their time on the system in a state of disengagement. Unsurprisingly, disengaged students had a very high proportion of unsupported edits (99.5 %), low potential generation percentage (29.7 %), and almost no used potential percentage (0.26 %).



Table 2 shows the pre-test and post-test scores broken down by cluster. These results show that the researchers/careful editors and the guess and check experimenters both did significantly better on the post-test than the pre-test, as determined by  $t$ -tests ( $p$  values are reported in Table 2), but that the confused guessers and disengaged students did not. However, it is worth noting that this could be an artifact of the group sizes (clusters 3 and 4 were smaller, so they will necessarily have higher  $p$  values due to less statistical power). Table 3 shows a similar pattern of map building performance by cluster. It is expected that the careful editors might perform well, but they may have been better supported if they were given feedback on how to use quiz results. Similarly, the guess and check experimenters could have performed better if they had been provided feedback on how to find relevant material in the resources, and then translate them into components of the causal model. Previous work by Segedy [16] has achieved some success through such feedback. Similar feedback early in the intervention would have helped the Confused Guessers, and the Disengaged group ( $n = 5$ ) very likely needed help directly from the teacher outside of the system.

**Table 2.** Means (and standard deviations) of assessment test scores by cluster

Cluster	Pre-test	Post-test	$t$	$p$	Cohen's $d$
s1. Researchers/Careful editors	8.00 (3.39)	9.44 (2.40)	1.93	0.045	0.49
s2. Guess and check experimenters	8.50 (4.11)	11.30 (2.80)	4.92	$3 \times 10^{-5}$	0.29
s3. Confused guessers	5.75 (3.24)	5.13 (11.60)	0.51	0.31	0.07
s4. Disengaged	2.33 (6.33)	4.33 (1.53)	0.96	0.22	0.43

**Table 3.** Means (and standard deviations) of map scores by cluster

Cluster	Unit 1 map score	Unit 2 map score
s1. Researchers/Careful editors	10.47 (3.81)	9.8 (3.65)
s2. Guess and check experimenters	10.82 (3.75)	9.52 (3.73)
s3. Confused guessers	2.44 (2.01)	3.78 (4.27)
s4. Disengaged	1.60 (2.19)	2.4 (3.36)

## 6.2 Temporal Behavior Analysis

To do a finer grained analysis of student activities, we considered an individual student's work to be a sequence of student-days, each with its own coherence metric values based on what the student did on that day, resulting in 213 student-day feature vectors. Student-days were then clustered on these metrics, (note that this means that an individual student appears as multiple data points during this clustering). We then look at the clusters themselves and the transitions of individual students between clusters from day to day.

Table 4 presents the student-day cluster analysis. The clusters of daily activities reveal similar patterns of activity to the overall student clustering, and are accordingly similarly named. On a daily basis, researchers/careful editors view the most information and make the fewest unsupported edits. Confused guessers view a lot of information that

had potential to answer their questions, but nevertheless make a large number of unsupported edits. The clustering of daily data reveals a much larger percentage of disengaged learners than the overall study analysis revealed. This indicates that although many students spent part of a day (average 38 %) disengaged ( $n = 70$ ), only very few were disengaged for extended periods of the study ( $n = 5$ ).

**Table 4.** Means (and standard deviations) of by-day CA metrics by cluster, from *SimSelf* student-day data

Cluster	Edit Freq.	Unsup. edit %	Info. view %	Potential Gen. %	Used potential %	Disengaged %
d1. Researchers/ Careful editors ( $n = 32$ )	0.40 (0.20)	64.2 % (16.2 %)	43.0 % (12.0 %)	41.8 % (15.7 %)	59.0 % (22.1 %)	16.2 % (9.2 %)
d2. Guess and check experimenters ( $n = 75$ )	0.76 (0.33)	41.7 % (14.5 %)	52.9 % (10.1 %)	66.9 % (16.1 %)	44.1 % (21.6 %)	21.2 % (14.9 %)
d3. Confused guessers ( $n = 39$ )	0.34 (0.35)	96.0 % (8.3 %)	29.0 % (21.3 %)	12.3 % (13.9 %)	5.3 % (12.6 %)	38.5 % (30.8 %)
d4. Disengaged ( $n = 70$ )	0.40 (0.34)	89.9 % (13.1 %)	40.3 % (20.7 %)	72.1 % (18.0 %)	13.0 % (17.9 %)	35.0 % (18.0 %)

Considering the student-days as sequences shows how students’ problem-solving behaviors changed on a day-to-day basis. Table 5 shows the frequency of all day-to-day transitions made by students with the ratio (in brackets) of the observed frequency of that transition to the frequency expected from a baseline random transition model where the number of transitions from cluster A to cluster B is proportional to the size of A times the size of B. The three highest and three lowest transition ratios across different clusters are shown in bold.

**Table 5.** Count of day-by-day *SimSelf* cluster transitions [and ratio with respect to random].

Cluster	d1	d2	d3	d4
d1. Researchers/Careful editors	4	9	<b>6</b>	8
	[1.17]	[1.13]	<b>[1.44]</b>	[1.12]
d2. Strategic experimenters	11	25	6	12
	[1.38]	[1.34]	[0.62]	[0.72]
d3. Confused guessers	<b>0</b>	7	<b>10</b>	11
	<b>[0.00]</b>	[0.72]	<b>[1.98]</b>	[1.26]
d4. Disengaged	<b>3</b>	<b>8</b>	<b>13</b>	18
	<b>[0.42]</b>	<b>[0.48]</b>	<b>[1.49]</b>	[1.20]

The results of this analysis show several interesting trends. First, transitions to the same cluster (along the diagonal, i.e. non-transitions) were more common than random for all clusters, indicating that students’ behavior profiles are at least somewhat stable

from one day to another. In particular, the confused guessers seemed stuck to their suboptimal work mode (1.98:1). Many more disengaged students became confused guessers than expected (1.49:1), but rarely they became guess and check experimenters or researchers/careful editors. In fact, only guess and check experimenters showed behavior transitions to researchers/careful editors (1.38:1) more often than expected. Researchers/careful editors had more of all four transitions than expected, indicating that it was an unlikely cluster for the final day. Overall, the experimenters were the most engaged group, and they were the only cluster that became disengaged less frequently than expected (0.72:1).

### 6.3 Comparison to Previous Work

To evaluate the stability of the CA features when identifying similar clusters of students in different student populations, we compared our clusters to those in a previous study [7] of an OELE called *Betty's Brain*, in which students similarly build a causal map. Clustering analysis of our data identified clusters of with similar characteristics, indicating that the CA metrics can be used across different student populations learning different material (Table 6).

**Table 6.** Means (and standard deviations) of CA metrics by cluster, from *Betty's Brain* data

Cluster	Edit Freq.	Unsup. edit %	Info. view %	Potential Gen. %	Used potential %	Disengaged %
b1. Researchers/ Careful editors (n = 24)	0.30 (0.11)	29.4 % (16.1 %)	42.4 % (11.0 %)	71.4 % (10.6 %)	58.9 % (15.4 %)	15.7 % (9.9 %)
b2. Strategic experimenters (n = 39)	0.60 (0.23)	54.4 % (14.8 %)	33.5 % (8.3 %)	58.7 % (18.9 %)	62.6 % (16.2 %)	10.9 % (7.4 %)
b3. Confused guessers (n = 5)	0.21 (0.06)	73.5 % (13.5 %)	58.9 % (7.7 %)	45.8 % (19.4 %)	23.1 % (12.6 %)	4.8 % (5.4 %)
b4. Disengaged (n = 6)	0.33 (0.11)	74.7 % (17.4 %)	27.0 % (9.6 %)	54.9 % (9.3 %)	28.0 % (8.7 %)	33.6 % (8.4 %)
b5. Engaged/Efficient (n = 24)	1.04 (0.32)	29.1 % (15.2 %)	35.4 % (8.6 %)	76.8 % (9.5 %)	82.0 % (9.0 %)	3.1 % (5.0 %)

The *Betty's Brain* data analysis identified an additional cluster (*Engaged and Efficient*) as well as one where students were more *strategic experiments* than guess and check experimenters (see Table 2). Their *engaged and efficient* students were characterized by a high edit frequency most of which (70.9 %) were supported. This cluster of students is distinct the other four clusters the two studies have in common in that they used a large majority of the potential they generated (82.0 %) and were rarely in a state of disengagement (3.1 %). A reason for this may be that the *Betty's Brain* study was run for a longer period of time (4 weeks instead of 2), and students evolved into engaged and efficient learners as the intervention progressed (see [16]). For the same reason, the guess and check experimenters may not have evolved into the strategic experimenters we saw in the *Betty's Brain* study. In addition, the *SimSelf* students were

much less engaged, 25.3 % disengaged versus 11.2 % disengaged. This large difference is most likely due to the fact that our recent study involved students at a regular, urban middle school whereas the other study involved students at a magnet school for high-achievers.

Although it is difficult to quantitatively compare the transition results from [12] to these new results because the cluster centroids are different, the results are fairly qualitatively consistent. In both studies, there were more transitions to the same cluster (i.e., non-transitions) than would be predicted by a random model. In addition, students tended to transition back and forth between “Confused” and “Disengaged”. Finally, and most interestingly, in both studies “Researchers/Careful Editors” were more likely than the “Experimenters” to become confused and eventually disengaged. This indicates that edit frequency is a better predictor of future engagement than the proportion of edits which are supported by information viewing.

## 7 Discussion and Conclusions

In this paper, we presented an analysis of students’ day-to-day problem solving approaches of middle school students using *SimSelf* [13] to learn about climate change. We modeled the students’ behavior with coherence analysis-based metrics [7] and clustered the resulting feature vectors to reveal commonalities in their problem solving approaches. We compared the resulting clusters to those in a different study [12] and we identified four of the five student learning behavior clusters found by that research. The missing cluster, engaged and efficient students, was likely missing due to the different student populations at the magnet school.

We also compared the students’ activities on a daily basis with their activities over the entire study. This revealed that although many students were disengaged for parts of a single day, only a very small number (5) were disengaged for most of the study. Finally, the cluster transition analysis showed that students were more likely than expected to transition to disengaged than expected, with the exception of the guess and check experimenters. As discussed before, with proper scaffolding, these students could have been encouraged to become more strategic experimenters. Similarly, confused guessers are highly likely to remain in that state or transition to disengaged, even though they are reading many of the right resources. This indicates that OELEs need to better support students who are not making progress to help them make the connections between the information that they are viewing and the problem that they are attempting to solve.

We have demonstrated that coherence-based metrics can be applied to model students in different populations doing different tasks. These metrics can be used to identify clusters of learning behaviors that are characteristic of common learning patterns. In future, these metrics could be embedded within OELEs to detect unproductive learning behaviors and to adapt the user interactions to encourage students to avoid becoming disengaged and/or prevent the confusion that leads to guessing.

**Acknowledgements.** This work has been supported by Institute of Educational Sciences CASL Grant #R305A120186 and the National Science Foundation’s IIS Award #0904387.

## References

1. Land, S., Hannafin, M., Oliver, K.: Student-centered learning environments: foundations, assumptions and design. In: Jonassen, D., Land, S. (eds.) *Theoretical Foundations of Learning Environments*, pp. 3–25. Routledge, New York (2012)
2. Segedy, J.R., Biswas, G., Sulcer, B.: A model-based behavior analysis approach for open-ended environments. *J. Educ. Technol. Soc.* **17**(1), 272–282 (2014)
3. Savery, J.R., Duffy, T.M.: Problem based learning: an instructional model and its constructivist framework. *Educ. Technol.* **35**(5), 31–38 (1995)
4. Winters, F., Greene, J., Costich, C.: Self-regulation of learning within computer-based learning environments: a critical synthesis. *Educ. Psychol. Rev.* **20**(4), 429–444 (2008)
5. Zimmerman, B., Schunk, D. (eds.): *Handbook of Self-Regulation of Learning and Performance*. Routledge, New York (2011)
6. Butler, D.L., Winne, P.H.: Feedback and self-regulated learning: a theoretical synthesis. *Rev. Educ. Res.* **65**(3), 245–281 (1995)
7. Segedy, J.R., Kinnebrew, J.S., Biswas, G.: Using coherence analysis to characterize self-regulated learning behaviours in open-ended learning environments. *J. Learn. Anal.* **2**(1), 13–48 (2015)
8. Sabourin, J., Shores, L., Mott, B., Lester, J.: Understanding and predicting student self-regulated learning strategies in game-based environments. *Int. J. Artif. Intell. Educ.* **23**, 94–114 (2013)
9. Baker, R.S., Ocumpaugh, J., Gowda, S.M., Kamarainen, A.M., Metcalf, S.J.: Extending log-based affect detection to a multi-user virtual environment for science. In: Dimitrova, V., Kufflik, T., Chin, D., Ricci, F., Dolog, P., Houben, G.-J. (eds.) *UMAP 2014. LNCS*, vol. 8538, pp. 290–300. Springer, Heidelberg (2014)
10. Snow, E.L., Jackson, G.T., McNamara, D.S.: Emergent behaviors in computer-based learning environments: computational signals of catching up. *Comput. Hum. Behav.* **41**, 62–70 (2014)
11. Luckin, R., Hammerton, L.: Getting to know me: helping learners understand their own learning needs through metacognitive scaffolding. In: Cerri, S.A., Gouardères, G., Paraguaçu, F. (eds.) *ITS 2002. LNCS*, vol. 2363, pp. 759–771. Springer, Heidelberg (2002)
12. Segedy, J.R., Kinnebrew, J.S., Biswas, G.: Coherence over time: understanding day-to-day changes in students’ open-ended problem solving behaviors. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M. (eds.) *AIED 2015. LNCS*, vol. 9112, pp. 449–458. Springer, Heidelberg (2015). doi:[10.1007/978-3-319-19773-9](https://doi.org/10.1007/978-3-319-19773-9)
13. Kinnebrew, J.S., Gauch, B.C., Segedy, J.R., Biswas, G.: Studying student use of self-regulated learning tools in an open-ended learning environment. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M. (eds.) *AIED 2015. LNCS*, vol. 9112, pp. 185–194. Springer, Heidelberg (2015)
14. Leelawong, K., Biswas, G.: Designing learning by teaching agents: the Betty’s brain system. *Int. J. Artif. Intell. Educ.* **18**(3), 181–208 (2008)
15. Jain, A., Dubes, R.: *Algorithms for Clustering Data*. Prentice Hall, Upper Saddle River (1988)
16. Segedy, J.R.: *Adaptive scaffolds in open-ended computer-based learning environments*. Doctoral dissertation, Vanderbilt University (2014)

# Are Pedagogical Agents' External Regulation Effective in Fostering Learning with Intelligent Tutoring Systems?

Roger Azevedo<sup>(✉)</sup>, Seth A. Martin, Michelle Taub,  
Nicholas V. Mudrick, Garrett C. Millar, and Joseph F. Grafsgaard

Department of Psychology, North Carolina State University, Raleigh, NC, USA  
{razeved, samarti7, mtaub, nvmudric,  
gcmillar, jfgrafsg}@ncsu.edu

**Abstract.** In this study we tested whether external regulation provided by artificial pedagogical agents (PAs) was effective in facilitating learners' self-regulated learning (SRL) and can therefore foster complex learning with a hypermedia-based intelligent tutoring system. One hundred twenty ( $N = 120$ ) college students learned about the human circulatory system with MetaTutor during a 2-hour session under one of two conditions: adaptive scaffolding (AS) or a control (C) condition. The AS condition received timely prompts from four PAs to deploy various cognitive and metacognitive SRL processes, and received immediate directive feedback concerning the deployment of the processes. By contrast, the C condition learned without assistance from the PAs. Results indicated that those in the AS condition gained significantly more knowledge about the science topic than those in the C condition. In addition, log-file data provided evidence of the effectiveness of the PAs' scaffolding and feedback in facilitating learners' (in the AS condition) metacognitive monitoring and regulation during learning. We discuss implications for the design of external regulation by PAs necessary to accurately detect, track, model, and foster learners' SRL by providing more accurate and intelligent prompting, scaffolding, and feedback regarding SRL processes.

**Keywords:** Self-regulated learning · Metacognition · Pedagogical agents · Externally regulated learning · ITS · Scaffolding · Learning · Product data · Process data

## 1 Objectives, Theoretical Framework, and Related Work

Self-regulated learning (SRL) is a hallmark of human learning and a key factor in problem solving, reasoning, and understanding complex instructional and training materials with advanced learning technologies (ALTs) such as intelligent tutoring systems (ITSs) [1, 2]. For example, when learning about complex STEM topics, research indicates that individuals can gain deep conceptual understanding through the effective use of cognitive, affective, metacognitive, and motivational (CAMP) self-regulatory processes [1, 3–6]. The successful use of cognitive and metacognitive SRL processes involves setting meaningful goals for one's learning, planning a course of action for

attaining these goals, deploying a diverse set of effective learning strategies in pursuit of the goals, continuously and accurately monitoring one's own understanding of the material and the appropriateness of the current information, and adapting one's goals, strategies, and navigational patterns based on the results of such monitoring processes and resulting judgments [7]. Unfortunately, there is ample interdisciplinary evidence to show that few learners engage in effective SRL [8, 9]. Although motivation and affect [10–12] play a role in determining learners' willingness to self-regulate, we assume a lack of cognitive and metacognitive self-regulatory knowledge and skills is the main obstacle to adequate regulation and, subsequently, deficient learning gains and conceptual understanding [2].

Furthermore, learners attempting to self-regulate often face limitations in their own metacognitive knowledge and skills, which, when compounded with a lack of domain knowledge, can result in cognitive overload, negative affective reactions, and decreased interest and persistence [6, 11, 12]. One method of relieving the cognitive burden placed on learners in this situation is to provide assistance in the form of adaptive scaffolding. Similar to seminal work by Graesser and colleagues and Chi and colleagues, previous experiments conducted by Azevedo and colleagues [7] on human tutors as external regulating agents established that adaptive scaffolding provided by a human tutor leads to greater deployment of sophisticated planning processes, metacognitive monitoring processes, and learning strategies as well as larger shifts in mental models of the domain. The purpose of the current work on externally regulated learning (ERL) is to empirically test whether the adaptive scaffolding provided by multiple artificial PAs (as externally regulating agents) within a hypermedia-based ITS (i.e., MetaTutor) is also capable of producing the same, or better, learning outcomes and increased use of effective SRL processes during STEM learning. As such, this study examines the effectiveness of several PAs in externally regulating and fostering complex learning with ITSs.

## 2 Method

### 2.1 Participants

One hundred twenty college students (52 % female) from a large university in North America participated in this study in 2015. The mean age of the participants was 20.4 years and their mean GPA was 3.29. All participants were paid up to \$40 for completion of the 2-day, 4-hour experiment.

### 2.2 Pretest and Posttest Measures

Several materials were developed for this study including self-report measures of emotions (e.g., EV, revised Agent Persona Inventory) and motivation and two versions of the pretest and posttest about the human circulatory system. For example, the pretest and posttest each included 30 four-foil multiple-choice items.

### 2.3 MetaTutor: Intelligent Hypermedia-Based Tutoring System for Biology

MetaTutor is an intelligent hypermedia-based tutoring system that includes 47 pages of text and static diagrams of the human circulatory system [13, 14]. During learning participants were guided by four PAs that provided timely scaffolding for each participant. Each agent, aside from *Gavin the Guide*, offered support on one specific component of SRL (i.e., planning, metacognition, and cognitive strategies). *Gavin's* objective was to support participants as they navigated the environment. *Pam the Planner* supported participants by emphasizing planning, activating prior knowledge, and creating relevant subgoals. *Mary the Monitor* supported participants by helping them monitor various metacognitive processes and make accurate metacognitive judgments during the session. *Mary* recommends the use of metacognitive processes such as content evaluations (CE), feelings of knowing (FOK), judgments of learning (JOL), and monitoring progress toward goals (MPTG). *Sam the Strategizer* encouraged effective cognitive strategy use (i.e., coordinating informational sources, making inferences, taking notes, summarizing hypermedia science content) as participants progressed toward completing their goals.

The MetaTutor interface (see Fig. 1) was designed to support, model, and foster self-regulated learning. The center of the interface contains the text and diagrams. These are the learning materials that are used to accomplish all subgoals and the overarching goal of learning about the circulatory system. The SRL palette is located on the right pane of the interface and enables participants to engage in SRL strategies. By clicking on the elements of the palette, participants can use eight strategies: creating summaries, making inferences, taking notes, activating prior knowledge, MPTG, CE, JOL, and FOK. Participants are free to use any of these components at any time throughout the session, and the strategies can be either user- or system-initiated. One of

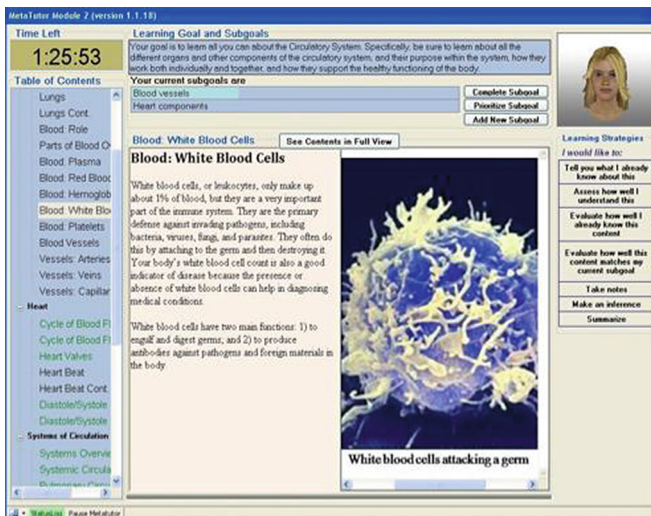


Fig. 1. Screenshot of the MetaTutor interface.



the four PAs is located just above the palette, in the top right corner of the interface. The agent that is displayed is dependent upon the circumstances of the session (i.e., what type of scaffolding the system is deploying, what type of instruction is being offered). Only one agent is displayed at a time in this window. To the left of the agent window and at the top of the interface are the participants' subgoals. The subgoal that the participant is currently working on is located at the top of the list as a reminder. On the left-hand pane of the interface, below the timer, is the table of contents. This allows participants to view the titles of each of the 47 pages, which are organized by section. At the bottom of the interface we find a textbox where participants are able to enter their subgoals and their prior knowledge about those subgoals. The textbox is also used throughout the session when the participant engages in certain SRL processes.

#### 2.4 Adaptive Scaffolding vs. No Scaffolding Conditions

We used two different versions of MetaTutor: one for each of the two conditions, an adaptive scaffolding (AS) condition and a control (C) condition. In the AS condition participants received timely scaffolding from the agents. This scaffolding was designed to reflect the interaction a learner would receive from a human tutor. In this condition there were both user- and system-initiated actions. User-initiated actions were made by using the SRL palette. For example, participants could click on the SRL palette to indicate they wanted to take notes or metacognitively judge the relevancy of text or diagrams to their current subgoals. System-initiated actions were brought on by a complex set of production rules that fire when certain conditions are met. For instance, when participants navigated to a page that was not relevant to their current subgoal and remained on this page for 15 s, a production rule was initiated that would fire Mary the Monitor. Mary would then prompt participants to make a CE about the relevancy of the page and image to their current subgoal. In total, MetaTutor uses 20 production rules (13 cognitive, 7 metacognitive) that are triggered by time and action thresholds. Participants in the AS condition were also afforded feedback from Pam while setting up their subgoals. She informed the participants on whether their proposed subgoal was too broad or too general and then continued to assist the participants in setting an appropriate subgoal.

In the *control condition*, participants were not afforded feedback or scaffolding from the agents. For example, during the subgoal setting phase, Pam only suggested the subgoal that the participant should choose. In this condition, participants were free to navigate the environment without any feedback or scaffolding from the agents. Further, they were not prompted to use any SRL strategies. However, it is important to note that the participants were still able to engage in SRL strategies on their own, if they so chose; they were afforded the same instructions and instructional videos; and they were exposed to the same multimedia learning content. Thus, the conditions were separated by the element of scaffolding and feedback, whereby the prompt and feedback group engaged in interaction with the agents, and the control condition did not. This design makes it possible to investigate the effectiveness of PAs in scaffolding participants as they engaged in conceptual learning. A complete description of the production rules governing the PAs' behaviors is beyond the scope of this paper.

## 2.5 Experimental Procedure

The MetaTutor study took place over two sessions that had to take place within a 3-day span. The first session lasted approximately 30–60 min and the second session lasted up to 180 min. After consenting to the study, participants were instrumented, and the eye-tracker and Attention Tool were calibrated. A baseline was established for electrodermal activity (EDA) as well as for the facial recognition of emotion software. The participant was then presented with an overview of the study. Following the overview, participants filled out a demographic questionnaire and several self-report measures of personality, emotions, and motivation. After completing these measures, participants were administered the 30-item multiple-choice pretest.

During the second session of the study, participants were instrumented, and the eye-tracker and Attention Tool were calibrated. A baseline was established for EDA as well as for the facial recognition of emotion software. The participant was then presented with an overview of what was going to take place during the session and was allowed to begin. The session started with Gavin giving a short introduction, and then an introductory video launched to introduce the agents and give an overview of the user interface and its functionality. After the video, Gavin gave the participants their overarching goal of the session, which was to learn all they could about the circulatory system. Before starting, the participant had to complete the AGQ and EV (i.e., motivation and emotions self-report measures). Next Pam the Planner assisted the participants to set up their subgoals. This was aided by an instructional video that explained how to set subgoals. After the participants successfully set their two subgoals, they were provided with their pretest scores and were offered the opportunity to switch either or both of them with any of the other five possible subgoals. After the participants made a decision on their subgoals, they were asked to recall everything they knew about that particular subgoal. This was used to determine prior knowledge of the learning content. Next the participants were required to take several self-report measures of emotions and motivation, and viewed an informational video that explained how to efficiently use the interface at a higher level (i.e., how to use the SRL palette). At this point, the participants were ready to start learning with the system. Throughout the 90-minute learning session, instrumented participants were presented with several emotions and motivation self-report measures (presented by the system based on time thresholds, learning episodes, assessment results, and SRL activities) while rich trace data were collected for subsequent analyses. At the completion of the session, they were administered the same self-report measures and an equivalent posttest, paid for their time, and debriefed on the study.

## 3 Results

### 3.1 Question 1: Do Different Scaffolding Conditions Lead Students to Gain Significantly More Knowledge About the Human Circulatory System?

An analysis of covariance (ANCOVA) with two levels (scaffolding conditions: AS or C), using posttest as the dependent measure and pretest ( $M_C = 18.73$ ,  $SD = 3.81$ ;  $M_{AS} = 15.90$ ,  $SD = 4.58$ ) as the covariate, was performed to answer this research

question. Before conducting each analysis, we ensured homogeneity of variance and significance of the covariate for each dependent variable. Results indicated that there were significant differences in posttest scores between experimental conditions while controlling for pretest score;  $F(1, 117) = 76.90, p < .001, \eta_p^2 = .40$ . Specifically, learners in the AS condition had significantly higher posttest scores ( $M = 21.12, SD = 4.25$ ) compared to learners in the C condition ( $M = 19.80, SD = 3.83$ ). Thus, results indicate that when learners were provided with prompts and feedback from the PAs, they outperformed learners who did not receive prompts and feedback from the PAs on the posttest. The maximum score on both pretest and posttest measures was 30.

### **3.2 Question 2: Do Different Scaffolding Conditions Impact the Duration, Frequency, and Quality of Learning and Knowledge Construction Activities, and Performance on Embedded Assessments During Learning with MetaTutor?**

Adaptive scaffolding of SRL by PAs involves well-orchestrated learner-system interactions involving learning activities (e.g., learners reading relevant multimedia content on the biology topic while monitoring several aspects of their existing knowledge of the material, emerging understanding, relevancy of content, etc.) and knowledge construction activities (e.g., taking notes on relevant multimedia content and adding newly found biology content to existing notes) followed by periodic embedded assessments at both the page and subgoal levels to assess the quality of the cognitive and metacognitive SRL processes deployed during learning with MetaTutor. As such, we conducted several independent *t*-tests<sup>1</sup> on key variables from learners' log-files to determine whether scaffolding conditions impacted the duration, frequency, and quality of learning and knowledge construction activities, and performance on embedded assessments during learning with MetaTutor.

The results show that those in the AS condition spent a significantly greater amount of time with the system during the second session (on average approx. 129 min for those in the AS condition vs. 106 min for those in the C condition; see Table 1). This result is accounted for by the amount of time learners in the AS condition were externally regulated by the four PAs while attempting to self-regulate their learning about the circulatory system. In contrast, those in the C condition spent a significantly greater proportion of time reading the science content. The acquisition and retention of the science content, based on reading, was periodically assessed by having learners perform page-level quizzes, and the results show that those in the AS condition took significantly more page-level quizzes and scored significantly better on them compared to those in the C condition. Note taking is a key knowledge construction activity, and while our results indicate no significant differences in the frequency and duration of note-taking events (including note checking) by both groups, we did find significant differences that show learners in the AS condition checked their notes more frequently

---

<sup>1</sup> The Bonferroni correction was used to adjust *p* values since several statistical tests were performed simultaneously on the data set.

**Table 1.** Means (and *SDs*) for learning and knowledge construction activities, and embedded assessments by scaffolding condition.

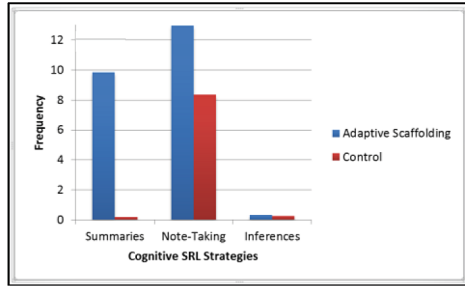
	No scaffolding	Adaptive scaffolding
Variable	<i>M (SD)</i>	<i>M (SD)</i>
Duration of session (s)	6372.12 (303.78)	7776.97 (796.01)*
Time spent reading (s)	5681.75 (266.92)	6268.40 (580.43)*
Proportion spent reading	0.8919 (0.04)*	0.8079 (.02)
Frequency of page quizzes	2.78 (6.70)	9.02 (7.74)*
Page quiz score	1.05 (1.11)	1.84 (.56)*
Frequency of note taking	6.25 (11.20)	7.90 (9.54)
Duration of note taking	503.52 (995.25)	585.10 (778.55)
Duration of note checking	87.63 (319.66)	139.85 (162.20)
Frequency of checking notes	1.88 (3.80)	3.35 (3.59)*
Number of summaries added to notes	0.12 (0.32)	4.63 (5.88)*
Frequency of subgoal quizzes	3.43 (2.03)	3.52 (2.38)
Subgoal quiz score	5.27 (2.24)	5.96 (2.11)

*Note.* \* =  $p < .05$ ; s = seconds.

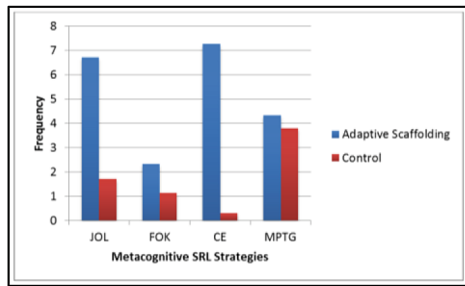
and added more summaries to their existing notes compared to learners in the C condition. Lastly, learners in neither condition differed in the number and scores on subgoal quizzes (see Table 1).

**3.3 Question 3: Do Different Scaffolding Conditions Impact the Frequency of Cognitive Strategies and Metacognitive Processes Deployed During Learning with MetaTutor, After Controlling for Pretest?**

Understanding the impact of PAs' external regulation of learners' SRL requires analyses of the frequency of both learner-initiated SRL moves and system-initiated ERL moves during learning with MetaTutor. As such, for assessing the differences in the use of cognitive learning strategies between scaffolding conditions while controlling for pretest score, we ran a MANCOVA with total summaries, total note taking, and total inferences made as the three dependent variables, scaffolding condition as the independent variable, and pretest score as the covariate. Results indicated a significant MANCOVA; Wilks'  $\lambda = .44$ ,  $F(3, 115) = 49.37$ ,  $p < .001$ ,  $\eta_p^2 = .56$ . Between-subjects effects indicated, while controlling for pretest score, significant differences in total summaries ( $F(1, 117) = 144.50$ ,  $p < .001$ ,  $\eta_p^2 = .55$ ) and total note taking instances ( $F(1, 117) = 4.88$ ,  $p = .03$ ,  $\eta_p^2 = .04$ ) between scaffolding conditions; however, there were no significant differences in total inferences made ( $F(1, 117) = 1.01$ ,  $p = .32$ ,  $\eta_p^2 = .01$ ) between scaffolding conditions. Specifically, while controlling for pretest, learners in the AS condition made significantly more summaries ( $M = 9.83$ ,  $SD = 6.04$ ) and took significantly more notes ( $M = 12.97$ ,  $SD = 12.82$ ), compared to the total summaries ( $M = 0.20$ ,  $SD = 0.44$ ) and note taking instances ( $M = 8.37$ ,  $SD = 14.21$ ) by those in the C condition (see Fig. 2).



**Fig. 2.** Mean frequency of three cognitive strategies by scaffolding condition (Color figure online).



**Fig. 3.** Mean frequency of four metacognitive strategies by scaffolding condition (Color figure online).

Additionally, we assessed the differences in use of metacognitive processes between scaffolding conditions, and ran a second MANCOVA with total JOL, total FOK, total CE, and total MPTG as the four dependent variables, scaffolding condition as the independent variable, and pretest score as the covariate. Results indicated a significant MANCOVA; Wilks'  $\lambda = .48$ ,  $F(4, 114) = 30.74$ ,  $p < .001$ ,  $\eta_p^2 = .52$ . Between-subjects effects revealed, while controlling for pretest score, that there were significant differences in total JOLs ( $F(1, 117) = 16.79$ ,  $p < .001$ ,  $\eta_p^2 = .13$ ) and total CEs ( $F(1, 117) = 113.14$ ,  $p < .001$ ,  $\eta_p^2 = .49$ ) between scaffolding conditions; however, there were no significant differences in total FOKs ( $F(1, 117) = 1.64$ ,  $p = .20$ ,  $\eta_p^2 = .01$ ) or total MPTGs ( $F(1, 117) = .73$ ,  $p = .40$ ,  $\eta_p^2 = .01$ ) between scaffolding conditions. Specifically, learners in the adaptive scaffolding condition made significantly more JOLs ( $M = 6.72$ ,  $SD = 7.68$ ) and CEs ( $M = 7.27$ ,  $SD = 4.65$ ) than learners who made JOLs ( $M = 1.72$ ,  $SD = 4.59$ ) and CEs ( $M = 0.30$ ,  $SD = 0.70$ ) in the control condition (see Fig. 3).

## 4 Conclusions and Future Directions

Our results indicate the adaptive scaffolding provided by PAs is effective in fostering complex learning about challenging STEM topics with ITSs such as MetaTutor in a relatively short amount of time (approx. 2 h). We demonstrated that compared to a control condition (where learners were not provided external regulation by PAs), those in the AS condition significantly improved their learning from pretest to posttest, spent disproportionately less time reading content (compared to other activities), took more page-level quizzes and scored significantly better on them, and also checked their notes more often and added summaries to them throughout the learning session. In addition, PAs' adaptive scaffolding was effective in prompting learners to use more cognitive strategies such as creating summaries and notes about the topic as well as using key metacognitive processes such as making JOLs and CEs to enhance their learning.

Our data also revealed some interesting results that need further examination by analyzing the rich multimodal trace data collected in this study. First, despite spending more time reading content, those in the C condition did not outperform those in the AS condition. This leads us to believe that finer-level analyses of the trace data are necessary to understand the dynamics between learners' SRL and the PAs' ERL throughout the session that facilitated better performance on the posttest. In addition, this finding also raises the questions about quantity versus quality—that is, more reading does not directly translate into more learning because more accurate and efficient reading by using key cognitive and metacognitive processes such as JOLs and CEs in combination with cognitive strategies such as summaries and note taking is key to foster complex learning. The same reasoning applies to the duration of note taking during learning. Second, while those in the AS condition outperformed those in the C condition on page-level quizzes, further investigation is still needed as to why learners in both conditions performed equally poorly on subgoal quizzes. Why is the ERL provided by PAs in the AS condition not leading to significantly better subgoal quiz scores? Third, it is evident that some cognitive strategies, such as making inferences, are too sophisticated for low prior knowledge learners who need to spend time reading to acquire knowledge about the topic and therefore should only be prompted by PAs once they have demonstrated a certain level of content understanding. Fourth, it is also evident that several metacognitive processes such as FOKs and MPTGs are seldom used during a learning task and therefore may not need to be prompted and scaffolded as often as other key metacognitive processes. On the other hand, this may also reveal low SRL prior knowledge.

Lastly, SRL and ERL between human and artificial agents is a core issue in the ITS community [16]. Contemporary research on ITSs with multiple agents has focused on SRL while relatively little effort has been made to use *externally regulated learning* as a guiding theoretical framework [7–9, 15]. This oversight needs to be addressed given the complex nature that self- and other-regulatory processes play when human learners and artificial agents interact to support learners' internalization of SRL processes. For example, learning with MetaTutor involves having a learner interact with four artificial PAs. Each agent plays different roles including modeling, prompting, and scaffolding SRL processes (e.g., planning, monitoring, and strategy use) and providing feedback

regarding the appropriateness and accuracy of learners' use of SRL processes in *real time* and potentially changing the ERL strategies based on its ability to monitor and reflect on the impact on the learners' individual responses to ERL. For example, the external regulating agent may have to modify its cognitive and metacognitive scaffolding at some point during learning and include affect regulation strategies (e.g., cognitive reappraisal) due to its perception, understanding, and reflection that its scaffolding and feedback is resulting in increasingly negative affective reactions (e.g., frustration) from a learner. Lastly, our goal is to build intelligent artificial agents capable of ERL by detecting, tracking, modeling, and fostering learners' cognitive, affective, metacognitive, and motivational (CAMP) SRL. By doing so, we will extend the human and computerized theoretical models typically used in this research area and therefore revolutionize the field of ITS by having interdisciplinary researchers address conceptual, theoretical, methodological, and analytical issues.

**Acknowledgements.** This study was supported by funding from the National Science Foundation (DRL 1431552).

## References

1. Azevedo, R., Alevin, V. (eds.): *International Handbook of Metacognition and Learning Technologies*. Springer, Amsterdam (2013)
2. Winne, P.H., Azevedo, R.: Metacognition. In: Sawyer, K. (ed.) *Cambridge Handbook of the Learning Sciences*, 2nd edn, pp. 63–87. Cambridge University Press, Cambridge (2014)
3. D'Mello, S., Graesser, A.: Confusion and its dynamics during device comprehension with breakdown scenarios. *Acta Psychol.* **151**, 106–116 (2014)
4. Kinnebrew, J., Segedy, J., Biswas, G.: Integrating model-driven and data-driven techniques for analyzing learning behaviors in open-ended learning environments. *IEEE Transactions on Learning Technologies* (in press). doi:[10.1109/TLT.2015.2513387](https://doi.org/10.1109/TLT.2015.2513387)
5. Sabourin, J., Lester, J.: Affect and engagement in game-based learning environments. *IEEE Trans. Affect. Comput.* **5**, 45–56 (2014)
6. Taub, M., Azevedo, R., Bouchet, F., Khosravifar, B.: Can the use of cognitive and metacognitive self-regulated learning strategies be predicted by learners' levels of prior knowledge in hypermedia-learning environments? *Comput. Hum. Behav.* **39**, 356–367 (2014)
7. Azevedo, R.: Issues in dealing with sequential and temporal characteristics of self- and socially-regulated learning. *Metacognition Learn.* **9**, 217–228 (2014)
8. Azevedo, R.: Defining and measuring engagement and learning in science: conceptual, theoretical, methodological, and analytical issues. *Educ. Psychol.* **50**, 84–94 (2015)
9. Azevedo, R., Taub, M., Mudrick, N.V., Martin, S.A., Grafsgaard, J.F.: Understanding and reasoning about real-time cognitive, affective, metacognitive processes to foster self-regulation with advanced learning technologies. In: Schunk, D., Greene, J.A. (eds.) *Handbook of Self-regulation and Performance*, 2nd edn. Routledge, New York (in press)
10. Calvo, R., D'Mello, S.K. (eds.): *New Perspectives on Affect and Learning Technologies*. Springer, New York (2015)

11. Duffy, M., Azevedo, R.: Motivation matters: interactions between achievement goals and agent scaffolding for self-regulated learning within an intelligent tutoring system. *Comput. Hum. Behav.* **52**, 338–348 (2015)
12. Harley, J.M., Bouchet, F., Hussain, S., Azevedo, R., Calvo, R.: A multi-componential analysis of emotions during complex learning with an intelligent multi-agent system. *Comput. Hum. Behav.* **48**, 615–625 (2015)
13. Azevedo, R., Harley, J., Trevors, G., Duffy, M., Feyzi-Behnagh, R., Bouchet, F., Landis, R.: Using trace data to examine the complex roles of cognitive, metacognitive, and emotional self-regulatory processes during learning with multi-agent systems. In: Azevedo, R., Aleven, V. (eds.) *International Handbook of Metacognition and Learning Technologies*, pp. 427–449. Springer, Amsterdam (2013)
14. Azevedo, R., Johnson, A., Chauncey, A., Graesser, A.: Use of hypermedia to convey and assess self-regulated learning. In: Zimmerman, B.J., Schunk, D.H. (eds.) *Handbook of Self-regulation of Learning and Performance*, pp. 102–121. Routledge, New York (2011)
15. Hadwin, A.F., Järvelä, S., Miller, M.: Self-regulated, co-regulated, and socially-shared regulation of learning. In: Zimmerman, B.J., Schunk, D.H. (eds.) *Handbook of Self-regulation of Learning and Performance*, pp. 65–84. Routledge, New York (2011)
16. VanLehn, K.: The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educ. Psychol.* **46**, 197–221 (2011)



# Intervention-BKT: Incorporating Instructional Interventions into Bayesian Knowledge Tracing

Chen Lin<sup>(✉)</sup> and Min Chi<sup>(✉)</sup>

The Department of Computer Science, North Carolina State University, Raleigh, USA  
{clin12,mchi}@ncsu.edu  
<http://www.csc.ncsu.edu>

**Abstract.** Bayesian Knowledge Tracing (BKT) is one of the most widely adopted student modeling methods in Intelligent Tutoring Systems (ITSs). Conventional BKT mainly leverages sequences of observations (e.g. *correct*, *incorrect*) from student-system interaction log files to infer student latent knowledge states (e.g. *unlearned*, *learned*). However, the model does not take into account the instructional interventions that generate those observations. On the other hand, we hypothesized that various types of instructional interventions can impact student's latent states differently. Therefore, we proposed a new student model called Intervention-Bayesian Knowledge Tracing (Intervention-BKT). Our results showed the new model outperforms conventional BKT and two factor analysis based alternatives: Additive Factor Model (AFM) and Instructional Factor Model (IFM); moreover, the learned parameters of Intervention-BKT can recommend adaptive pedagogical policies.

**Keywords:** Knowledge tracing · Hidden Markov Model · Input Output Hidden Markov Model · Student modeling · Instructional intervention

## 1 Introduction

Bayesian Knowledge Tracing (BKT) is one of the most widely adopted student modeling methods. BKT leverages sequences of observations (e.g. *correct*, *incorrect*) from student-system interaction log files to continually update the estimate of student latent knowledge (e.g. *unlearned*, *learned*), regardless of the instructional interventions generate the corresponding observations. Instructional interventions are actions initiated by the system guiding student learning activity. For example, two common instructional interventions are *elicit* and *tell*: *Elicit* represents asking a student what is the next step, while *tell* means delivering educational content via a written statement that reveals the next step.

While conventional BKT does not take into account various types of instructional interventions that generate student observations, they can directly impact student's latent states differently. For example, a correct observation should be treated differently depending on whether it is drawn from an *open-ended question* or a *multiple choice* one. Similarly, a correct observation should be treated differently depending on whether it is generated by the student (e.g., the tutor *elicits*)

or the tutor (e.g., the tutor *tells*). We proposed a new approach Intervention-Bayesian Knowledge Tracing (Intervention-BKT), which can: (1) incorporate different types of instructional interventions into student model, and (2) can tease apart their effects on student’s performance by training a separate set of parameters for each intervention type.

Much of the prior research on evaluating student models did not take different pedagogical strategies that an ITS can employ into account. In our experiment, we trained our models on datasets generated from four training corpus following different pedagogical strategies that vary from ineffective policies to effective ones. We investigated whether Intervention-BKT would outperform conventional BKT regardless of the pedagogical strategies employed. Additionally, we extended our comparison to two other factor analysis based approaches: the widely applied Additive Factor Model (AFM) [4] and the Instructional Factor Model (IFM) [5]. The latter can be seen as an extension of AFM to incorporate different instructional interventions. Finally, we use the parameters learned from our Intervention-BKT models to design adaptive and personalized pedagogical policies.

## 2 Related Work

In recent years a variety of extensions of BKT have been investigated. Pardos and Heffernan [8] proposed KT-IDEM model by adding a problem difficulty node to the conventional BKT model. Their results showed that KT-IDEM model significantly outperformed BKT on ASSISTments dataset but not on Cognitive Tutor dataset [8]. While KT-IDEM assumes that  $S_t$  only depends on  $S_{t-1}$ , but not the input  $I_t$ , our model assumes that  $S_t$  (a student’s knowledge state at a time  $t$ ) depends on both  $S_{t-1}$  (a student’s previous knowledge state at  $t-1$ ) and the current input  $I_t$ . In other words, we assume that the input (i.e., instructional interventions) impact student knowledge state while KT-IDEM does not.

Beck et al. proposed the HELP model [3] to measure the impact of the tutors’ help. The basic structure of the HELP model is very similar to our Intervention-BKT. Note that the input nodes in the Intervention-BKT represents instructional actions (*elicit* vs. *tell*) that are determined by the system. However, in their ITS, help is requested by the student, which may imply the higher knowledge level a student has, the less likely he/she will ask for help. That is, whether the student would ask for help at time  $t$  may depend on the student’s learned state  $S_t$ , which is not reflected in the HELP model. This might be one of the reasons why their results showed that HELP model did not yield a more accurate prediction compared to BKT.

Additionally, a series of research have been done on applying individualized parameters to BKT. For example, Pardos and Heffernan proposed Prior Per Student model [9] which adds a multinomial node representing student’s incoming competence to the BKT model and they showed their model performed better. Yudelson et al. [11] proposed to use student-specific probability of learning and showed that their method is more effective than BKT. Finally, an innovative approach by Baker and Corbett [2] is to contextually estimate whether each student

guesses or slips, thus avoiding the effect of identifiability and model degeneracy caused by uncertainty. Results showed that their model substantially improved accuracy and reliability compared to the conventional BKT model.

In our paper, we will compare our Intervention-BKT model and BKT against two factor analysis based methods. Previously, BKT has been directly compared against factor analysis based method [7] and the results showed that the latter is as good or better than BKT. However, datasets in their comparisons mainly involve single intervention. In this paper, we will compare all four models on a dataset involving two types of instructional interventions *elicit* and *tell*.

### 3 Method

#### 3.1 Bayesian Knowledge Tracing (BKT)

BKT is a user modeling method extensively used in ITS. Figure 1 shows a graphical representation of the model and a possible sequence of student observations. The shaded nodes  $S$  represent hidden knowledge states. The unshaded nodes  $O$  represent observation of students' behaviors. The edges between the nodes represent their conditional dependence.

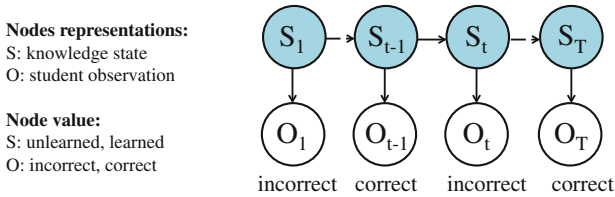


Fig. 1. The Bayesian network topology of the standard Knowledge Tracing model

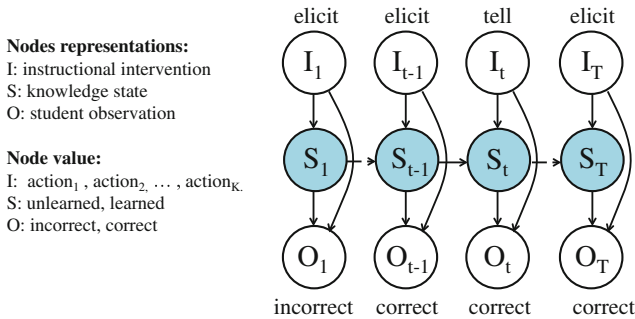
Fundamentally, the BKT model is a two-state Hidden Markov Model (HMM) [13] characterized by five basic elements: (1)  $\mathbf{N}$ , the number of different types of hidden state; (2)  $\mathbf{M}$ , the number of different types of observation; (3)  $\mathbf{\Pi}$ , the initial state distribution  $P(S_0)$ ; (4)  $\mathbf{T}$ , the state transition probability  $P(S_{t+1}|S_t)$  and (5)  $\mathbf{E}$ , the emission probability  $P(O_t|S_t)$ . Note that both  $\mathbf{N}$  and  $\mathbf{M}$  are pre-defined before training occurs, while  $\mathbf{\Pi}$ ,  $\mathbf{T}$  and  $\mathbf{E}$  are learned from the students' observation sequence.

Conventional BKT assumes there are two types of hidden knowledge state ( $\mathbf{N}=2$ ), that is, the student's knowledge states (i.e., *unlearned* and *learned*). It also assumes there are two types of student observation ( $\mathbf{M}=2$ ), that is, the student's performance (i.e., *incorrect* and *correct*). BKT makes two assumptions about its conditional dependence as reflected in the edges in Fig. 1. The first assumption BKT makes is a student's knowledge state at a time  $t$  is only contingent on her knowledge state at time  $t-1$ . The second assumption is a student's performance at time  $t$  is only dependent on her current knowledge state.

These two assumptions are captured by the state transition probability  $\mathbf{T}$  and the emission probability  $\mathbf{E}$ . To fit in the context of student learning, BKT further defines five parameters: (1) **Prior Knowledge** =  $P(S_0=\text{learned})$ ; (2) **Learning rate** =  $P(\text{learned} \mid \text{unlearned})$ ; (3) **Forget** =  $P(\text{unlearned} \mid \text{learned})$ ; (4) **Guess** =  $P(\text{correct} \mid \text{unlearned})$  and (5) **Slip** =  $P(\text{incorrect} \mid \text{learned})$ . Baum-Welch algorithm (or EM method) is used to iteratively update the model's parameters until a maximized probability of observing the training sequence is achieved.

### 3.2 Intervention Bayesian Knowledge Tracing (Intervention-BKT)

Intervention-BKT is build by incorporating different types of instructional interventions into BKT. Its Bayesian network topology is displayed in Fig. 2. Compared with BKT, Intervention-BKT adds a sequence of unshaded input nodes  $I$ . The arrows between input nodes  $I$  and student observation nodes  $O$  represent how instructional interventions affect a student's performance. The arrows between input nodes  $I$  and knowledge state nodes  $S$  represent how instructional interventions affect a student's hidden knowledge state.



**Fig. 2.** The Bayesian network topology of the Intervention-BKT model

Intervention-BKT is a special case of Input Output Hidden Markov Model (IOHMM) [14], which is extended from HMM. This model is characterized by six basic elements: (1)  $\mathbf{K}$ , the number of different types of input; (2)  $\mathbf{N}$ , the number of different types of hidden state; (3)  $\mathbf{M}$ , the number of different types of observation; (4)  $\mathbf{\Pi}$ , the initial state distribution  $P(S_0)$ ; (5)  $\mathbf{T}$ , the state transition probability  $P(S_t|I_t, S_{t-1})$  and (6)  $\mathbf{E}$ , the emission probability  $P(O_t|I_t, S_t)$

Intervention-BKT makes two distinctions compared to BKT. First, it employs a parameter  $K$  representing the number of input types, that is, the instructional intervention types. Second, Intervention-BKT makes two different assumptions about its conditional dependence as represented by the edges in Fig. 2: (1) a student's knowledge state at a time  $t$  is contingent on her previous state at time  $t-1$  **as well as the current intervention  $I_t$** ; (2) a student's performance at time  $t$  is dependent on her current knowledge state  $S_t$  **as well as the**

**Table 1.** Elicit vs. Tell

(a) Elicit Version	(b) Tell Version
1. <b>T:</b> So let’s start with determining the value of v1	1. <b>T:</b> So let’s start with determining the value of v1.
2. <b>T:</b> Which principle will help you calculate the rock’s instantaneous magnitude of velocity at T1? <b>{ELICIT}</b>	2. <b>T:</b> To calculate the rock’s instantaneous magnitude of velocity at T1, we will apply the definition of kinetic energy again. <b>{TELL}</b>
3. <b>S:</b> definition of kinetic energy	

**current intervention  $I_t$ .** Similarly, Our Intervention-BKT employs  $1 + 4 \times K$  parameters (compared with 5 parameters of BKT) to describe its conditional probability. The **Prior Knowledge** share the same definition as conventional BKT: **Prior Knowledge**=  $P(S_0=learned)$ . For each of the K types of interventions  $A_j, j \in [1, K]$ , Intervention-BKT defines four parameters:

$$\begin{aligned}
 \text{Learning Rate}_{A_j} &= P(\text{learned}|\text{unlearned}, I_t = A_j ) \\
 \text{Forget}_{A_j} &= P(\text{unlearned}|\text{learned}, I_t = A_j ) \\
 \text{Guess}_{A_j} &= P(\text{correct}|\text{unlearned}, I_t = A_j ) \\
 \text{Slip}_{A_j} &= P(\text{incorrect}|\text{learned}, I_t = A_j )
 \end{aligned}$$

In this paper, we mainly focus on modeling two types of instructional intervention *elicit* and *tell*. A possible sequence of instructional interventions is suggested above input node in Fig.2. Note that the conventional BKT model is trained from a sequence of output representing the student’s performance, whereas the Intervention-BKT model is trained from a sequence of instructional interventions and the corresponding student’s performance.

## 4 Four Training Corpus

Cordillera [10] is a Natural Language ITS teaching college level introductory physics and all participants in our training corpus experienced identical procedure: (1) completed a survey; (2) read a textbook; (3) took a pretest; (4) solved the same seven training problems on Cordillera, and finally (5) took a post-test. Cordillera provides two types of instructional interventions *elicit* and *tell*. Table 1 demonstrates these two interventions delivering the same domain content.

Four training corpus Random, Hybrid, NormGain and InvNormGain were involved in this study. They follow different pedagogical policies with various effectiveness on deciding when to *elicit* and when to *tell*. The remaining components of them are identical. As reported earlier in [12], students learned greatly in all four training corpus, but NormGain students learned in a significantly deeper way, while no difference was found among the rest three. That is to say, the pedagogical policies are most effective in NormGain corpus.

In total, there are 44923 data points from 170 students. More specifically, Random comprises 19584 data points from 64 students; Hybrid comprises 10113 data points from 37 students; NormGain comprises 7691 data points from 37 students and InvNormGain comprises 7535 data points from 32 students. Each student completed around 300 training problem steps. A data point in our training dataset is either the first attempt by students in response to a tutor *elicits*, or a tutor *tells* the next step. The pretest and post-test have the 33 identical test items. All of the tests were graded in a double-blind manner by a single domain expert (not the author). Each test question was assigned two grades: overall and KC-based grade. The overall grade was a score in the range  $[0, 1]$  describing the correctness of an answer as a whole, while the KC-based grade was a score in the same range describing the correctness regarding a particular KC.

## 5 Experiments

Three experiments were conducted. First, we investigated whether Intervention-BKT would outperform BKT on post-test scores prediction. It is commonly considered that relevant knowledge in domains such as math and science is structured as a set of independent but co-occurring Knowledge Components (KCs). A *Knowledge Component (KC)* is “a generalization of everyday terms like concept, principle, fact, or skill, and cognitive science terms like schema, production rule, misconception, or facet” [10]. It is assumed that the student’s knowledge state at one KC has no impact on her understanding of any other KCs. This is an idealization, but it has served ITS developers well for many decades as a fundamental assumption made by many student models [6]. In Cordillera, two domain experts identified six primary KCs: Kinetic Energy(KE), Gravitational Potential Energy (GPE), Spring Potential Energy (SPE), Total Mechanical Energy (TME), Conservation of Total Mechanical Energy (CTME), and Change of Total Mechanical Energy (ChTME). We investigated both BKT and Intervention-BKT on each of six primary KCs individually and across KCs (Across). Additionally, to investigate whether the pedagogical strategies would play a role on model performance, we compared them across four datasets individually and combined.

Second, we extended our comparison to two widely applied factor analysis based student modeling approaches: Additive Factor Model (AFM) and Instructional Factor Model (IFM). The original work of [5] has shown IFM outperforms AFM in post-test score prediction. Their experiments were performed by training a KC-general model on Random dataset combined with 10-fold cross-validation. In order for our results to be comparable, we followed the same procedure. The same measurements BIC and 10-fold RMSE were reported.

Third, we explored personalized pedagogical policies suggested by the learned parameters from Intervention-BKT. To make pedagogical recommendations when a student is in the unlearned state, we compared the learning rate (defined as the probability that a student will transit from the unlearned state to the learned state) for *elicits* and *tells* (**Learning Rate**<sub>elicits</sub> vs. **Learning Rate**<sub>tells</sub>). The tutor action leading to **higher** learning rate will be preferred. Similarly, to

make pedagogical recommendations when a student is in the learned state, we compared the forget rate (defined as the probability that a student will transit from the learned state to the unlearned state) for elicits and tells ( $\mathbf{Forget}_{\text{elicit}}$  vs.  $\mathbf{Forget}_{\text{tell}}$ ). The tutor action leading to **lower** forget rate is preferred.

## 6 Results

The forget parameter  $\mathbf{Forget}$  in BKT is fixed to be 0 conventionally, yet prior research showed that BKT may perform better when  $\mathbf{Forget}$  is unfixed [1]. We found using the fixed  $\mathbf{Forget}$  or unfixed  $\mathbf{Forget}$  does not make much difference. Besides, using an unfixed  $\mathbf{Forget}$  can provide us with recommended instructional interventions when students are in the “learned” state. Thus, we will report our models with unfixed  $\mathbf{Forget}$  only.

Three statistics Akaike Information Criterion (AIC), Bayesian Information Criteria (BIC) and the cross-validation Root Mean Squared Error (RMSE) are employed to evaluate our models. For all these measurements, the lower the value, the better the model performs.

### 6.1 Intervention-BKT vs. BKT

First we will show the results for *Combined* dataset in Table 2. As can be seen, Intervention-BKT produces better model fit than conventional BKT with lower AIC and BIC for all KCs. Furthermore, Intervention-BKT makes more accurate post-test score prediction with LOOCV RMSE at least 0.05 lower than conventional BKT model. For KC ChTME, Intervention-BKT decreases the RMSE by more than 0.2 as marked by “\*\*\*\*\*”.

Next, we compared Intervention-BKT and BKT on four datasets. The same pattern was found in all datasets. Given the space, we only present results for Random in Table 3(a) and NormGain in Table 3(b). Again, both Table 3 (a) and (b) show Intervention-BKT achieves lower LOOCV RMSE than BKT. For AIC

**Table 2.** Compare Intervention-BKT and conventional BKT on Combined

KC	AIC		BIC		LOOCV RMSE	
	BKT	Intervention-BKT	BKT	Intervention-BKT	BKT	Intervention-BKT
KE	7847	<b>5343</b>	7879	<b>5412</b>	0.356	<b>0.252</b> **
GPE	7712	<b>5376</b>	7743	<b>5445</b>	0.306	<b>0.248</b> *
SPE	3325	<b>2037</b>	3356	<b>2105</b>	0.419	<b>0.288</b> **
TME	7911	<b>5449</b>	7943	<b>5518</b>	0.347	<b>0.229</b> **
CTME	2621	<b>1620</b>	2652	<b>1689</b>	0.326	<b>0.254</b> *
ChTME	2733	<b>1318</b>	2764	<b>1388</b>	0.471	<b>0.233</b> *****
ACROSS	32121	<b>20495</b>	32152	<b>20564</b>	0.369	<b>0.275</b> *

Note: better AIC/BIC in **bold**

\*: difference>0.05; \*\*: difference>0.1; \*\*\*: difference>0.15; \*\*\*\*: difference>0.2

**Table 3.** Compare Intervention-BKT and conventional BKT on Random and NormGain

KC	AIC		BIC		LOOCV RMSE	
	BKT	Intervention-BKT	BKT	Intervention-BKT	BKT	Intervention-BKT
KE	3687	<b>2420</b>	3710	<b>2467</b>	0.353	<b>0.225**</b>
GPE	3026	<b>1879</b>	3047	<b>1926</b>	0.340	<b>0.245*</b>
SPE	1637	<b>953</b>	1659	<b>1000</b>	0.396	<b>0.267**</b>
TME	3565	<b>2184</b>	3586	<b>2231</b>	0.326	<b>0.189**</b>
CTME	1371	<b>836</b>	1392	<b>883</b>	0.275	<b>0.211*</b>
ChTME	1160	<b>607</b>	1182	<b>655</b>	0.453	<b>0.241***</b>
Across	14440	<b>8599</b>	14462	<b>8647</b>	0.373	<b>0.255**</b>

3(a) Random dataset

KC	AIC		BIC		LOOCV RMSE	
	BKT	Intervention-BKT	BKT	Intervention-BKT	BKT	Intervention-BKT
KE	<b>1354</b>	1177	<b>1371</b>	1212	0.295	<b>0.271</b>
GPE	1407	<b>1164</b>	1423	<b>1199</b>	0.229	<b>0.178 *</b>
SPE	491	<b>397</b>	507	<b>432</b>	0.368	<b>0.317 *</b>
TME	1413	<b>1181</b>	<b>1429</b>	1216	0.287	<b>0.218 *</b>
CTME	346	<b>257</b>	362	<b>292</b>	0.307	<b>0.256 *</b>
ChTME	<b>36</b>	51	<b>51</b>	86	0.523	<b>0.436 *</b>
Across	5097	<b>3913</b>	5113	<b>3948</b>	0.342	<b>0.257 *</b>

3(b) NormGain dataset

and BIC, Intervention-BKT yields better results for all KCs consistently for Random. It also beats BKT for all KCs for NormGain except for KE and ChTME.

For both Random and NormGain dataset, Intervention-BKT outperforms BKT. However, Intervention-BKT makes greater improvement on Random than on NormGain dataset. More specifically, Intervention-BKT improves BKT by more than 0.1 in five out of seven cases for Random, whereas none of the case shows an improvement greater than 0.1 for NormGain. One possible explanation is Random makes random tutorial decision, whereas NormGain employs pedagogical policies induced by Reinforce Learning (RL). There might be some dependence between pedagogical policies and student’s performance, which may cause Intervention-BKT behaves less effective for the NormGain dataset.

## 6.2 Intervention-BKT vs. BKT vs. AFM vs. IFM

Table 4 shows the performance of the four models Interventio-BKT, BKT, AFM and IFM with respect to post-test score prediction. As we can see, for both BIC and 10-fold RMSE, we have Intervention-BKT > BKT > IFM > AFM and the difference is significant.



**Table 4.** Intervention-BKT vs. BKT vs. AFM vs. IFM on predicting post-test score

Model	BIC	10-fold RMSE
Intervention-BKT	866	0.268
BKT	1170	0.309
IFM	2252	0.453
AFM	2443	0.470

### 6.3 Intervention-BKT Parameter Analysis

Table 5 shows the pedagogical suggestion made by Intervention-BKT for all 7 KCs across 5 datasets. Columns 1 to 5 show the suggestions for students who are in the unlearned state. Columns 6 to 10 show the suggestions for students who are in the learned state. As we can see, the recommendations for the system to *tell* or *elicit* vary greatly depending on the KC as well as the training corpus used.

**Table 5.** Pedagogical suggestion made for KCs when they are unlearned or learned

	Unlearned					Learned				
	R	H	N	I	C	R	H	N	I	C
KE	0.03	0.04	0.083	-0.010	0.021	-0.001	-0.002	-0.043	0.004	-0.010
GPE	-0.002	0.001	0.019	-0.005	0.010	0.001	0.001	0.009	-0.005	-0.004
SPE	0.008	0.003	0.013	0.004	0.006	0.002	0.006	-0.014	-0.005	0.001
TME	-0.009	-0.475	-0.009	-0.031	0.004	0.005	0.000	0.000	0.011	-0.002
CTME	0.005	-0.020	-0.024	-0.002	-0.001	-0.001	0.011	0.011	-0.010	0.001
ChTME	0.022	0.097	0.000	-0.005	0.025	-0.014	0.052	0.000	0.003	-0.013
ACROSS	-0.005	-0.006	-0.007	-0.041	-0.005	0.001	0.000	0.001	0.013	0.001

Tell
  Elicit

## 7 Discussion

In this work, we proposed the Intervention-BKT model which incorporates multiple types of instructional interventions into conventional BKT’s framework. Our results demonstrated that Intervention-BKT leads to a substantial improvement compared to the BKT model. We also showed that when using RMSE, the former is consistently better than the latter regardless of the effectiveness of the pedagogical policies employed and the difference can be large. Furthermore, we extended our comparison to two other models AMF and IFM and showed Intervention-BKT > BKT > IFM > AFM. Finally, our model showed great potential in closing the loop of instruction design. The learned parameters provide adaptive pedagogical suggestions for students in different learning states.

Based on our results, we are confident to say incorporating instructional interventions into BKT enhances model performance significantly, thus it merits further investigation.

For future work, we will investigate the effectiveness of Intervention-BKT using other larger datasets from other tutoring systems that may involve multiple instructional interventions, such as *skip* (*elicit* a question without asking students for explanation) and *justify* (ask students to explain after they give an answer). Additionally, we will experimentally compare the pedagogical policies suggested by Intervention-BKT with our RL-induced policies.

**Acknowledgments.** This research was supported by the NSF Grant 1432156 “Educational Data Mining for Individualized Instruction in STEM Learning Environments”.

## References

1. Beck, J.: Difficulties in inferring student knowledge from observations (and why you should care). In: Educational Data Mining: Supplementary Proceedings of the 13th International Conference of Artificial Intelligence in Education, pp. 21–30 (2007)
2. Baker, R.S.J., Corbett, A.T., Aleven, V.: More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 406–415. Springer, Heidelberg (2008)
3. Beck, J.E., Chang, K., Mostow, J., Corbett, A.T.: Does help help? introducing the bayesian evaluation and assessment methodology. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 383–394. Springer, Heidelberg (2008)
4. Cen, H., Koedinger, K.R., Junker, B.: Learning factors analysis – a general method for cognitive model evaluation and improvement. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 164–175. Springer, Heidelberg (2006)
5. Chi, M., Koedinger, K. R., Gordon, G. J., Jordon, P., VanLahn, K.: Instructional factors analysis: A cognitive model for multiple instructional interventions (2011)
6. Corbett, A.T., Anderson, J.R.: Modeling the acquisition of procedural knowledge. *UMUAI* 4(4), 253–278 (1994)
7. Gong, Y., Beck, J.E., Heffernan, N.T.: Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In: Aleven, V., Kay, J., Mostow, J. (eds.) ITS 2010, Part I. LNCS, vol. 6094, pp. 35–44. Springer, Heidelberg (2010)
8. Pardos, Z.A., Heffernan, N.T.: KT-IDEM: introducing item difficulty to the knowledge tracing model. In: Konstan, J.A., Conejo, R., Marzo, J.L., Oliver, N. (eds.) UMAP 2011. LNCS, vol. 6787, pp. 243–254. Springer, Heidelberg (2011)
9. Pardos, Z.A., Heffernan, N.T.: Modeling individualization in a bayesian networks implementation of knowledge tracing. In: De Bra, P., Kobsa, A., Chin, D. (eds.) UMAP 2010. LNCS, vol. 6075, pp. 255–266. Springer, Heidelberg (2010)
10. VanLehn, K., Jordan, P.W., Litman, D.: Developing pedagogically effective tutorial dialogue tactics: experiments and a testbed. In: SLaTE, pp. 17–20 (2007)
11. Yudelson, M.V., Koedinger, K.R., Gordon, G.J.: Individualized bayesian knowledge tracing models. In: Lane, H.C., Yacef, K., Mostow, J., Pavlik, P. (eds.) AIED 2013. LNCS, vol. 7926, pp. 171–180. Springer, Heidelberg (2013)

12. Chi, M., VanLehn, K., Litman, D., Jordan, P.: An evaluation of pedagogical tutorial tactics for a natural language tutoring system: a reinforcement learning approach. *int. j. artif. intell. educ.* **21**(1–2), 83–113 (2011)
13. Eddy, S.R.: Hidden markov models. *Curr. Opin. Struct. Biol.* **6**(3), 361–365 (1996)
14. Marcel, S., Bernier, O., Viallet, J.E., Collobert, D.: Hand gesture recognition using input-output hidden markov models. In: *fg*, p. 456. IEEE, March 2000

# **Short Papers**

# “i-Read”: A Collaborative Learning Environment to Support Students with Low Reading Abilities

Nizar Omheni<sup>(✉)</sup> and Ahmed Hadj Kacem

ReDCAD Research Laboratory and Faculty of Economics and Management,  
Sfax University, Sfax, Tunisia  
nizar.omheni@isigk.rnu.tn,  
ahmed.hadjkacem@fsegs.rnu.tn

**Abstract.** Many students suffer of online reading difficulties because of their low abilities of text comprehension. Several educators tried to set strategies to support learners during their online reading. In current work, we present an online reading environment where students can enroll in virtual reading class, to read and annotate their documents. Based on students’ annotation traces, we build their personality profiles which reflect their level of reading performance. Given the students’ reading abilities, we share the annotations of skilled readers with those having problems of text comprehension. The experimental results show the efficiency of the proposed approach to support learners with low reading abilities.

**Keywords:** Online reading comprehension · Collaborative learning · Annotations · Learner’s personality profile

## 1 Introduction

Many students face difficulties in reading because of their poor ability of text comprehension. An individual’s ability to comprehend text means his capacity to read text, process it and understand its meaning. In “face-to-face” reading class, teachers assist students to develop their reading abilities using appropriate pedagogical strategies [1]. In online learning context, instructors and learners are separated physically, so it is challengeable to diversify instructions according to students’ characteristics and abilities. Effectively, we need to implement an effective online instructional system based on proven and sound theories from science of learning that help students to overcome their reading difficulties in online learning context. Annotation activity is viewed as an effective strategy that could be used to improve students’ abilities of reading comprehension [2]. Other method is shown as effective strategy used where the immediate intervention of a teacher is absent in distance learning context, it is the peer learning method where students learn with and from each other [3]. Certain researchers try to combine the two strategies cited previously (annotation and peer learning) in one collaborative reading annotation environments [4]. In present work we consider such approach to improve students’ reading skills. To assess readers’ ability of reading

comprehension we consider the students' personality traits derived through their annotation activities [5, 6]. The presented computational model is a personality-based e-learning system of virtual reading class which we called "i-Read" environment, where students can enroll, read online, annotate and share their annotations. In what follow, we review briefly the related literature. Then, we present the architecture and the functionalities of the proposed reading environment. Thereafter, we present the conducted experimentation to show the effectiveness of our system's functionalities to support students suffering of reading comprehension difficulties. Finally, we discuss our results, we draw some conclusions and we suggest certain possible directions for future works.

## 2 Background

The emergence of reading online technology leads to changing the nature of literacy to comprises the skills and competencies needed for reading, writing and participating on the web which makes understanding reading in the 21<sup>st</sup> century more complicated [7]. For lifelong learners worldwide who aren't enrolled in a traditional institutional frameworks but subscribed in online learning environments where no teachers; no supervision; nor entry requirements; thousands of students in a single course; students teaching each other and grading each others' work, the process of teaching online reading strategies is more challengeable. Further studies show that online collaborative reading is an efficient strategy to improve students' reading comprehension. Indeed, reading in groups provides students with opportunities to develop their abilities to construct meaning and knowledge from text which helps them to achieve a deep understanding of reading material [8].

Several researchers show that using annotations of such experienced readers as experts or senior students may be helpful to those having reading difficulties or seeking for deeper understanding of text [9, 10].

In sum, based on the previous review of the online reading comprehension literature, we saw that the reading strategies used to achieve high level of reading performance vary all depends to different factors such as: students' skills, presence of instructor, online technologies, and context of reading. In present work, we are interested to overcome the shortcomings of online reading comprehension through collaborative reading and annotation strategies.

## 3 i-Read: A Collaborative Online Reading Environment

To overcome the shortcomings of reading online, we propose a collaborative reading environment called "i-Read" where readers can read the same text separately and make separate annotations of the text. The system builds the learners' personality profiles based on their traces of annotation, after which readers will be classified according to their scores of neuroticism and consciousness traits, to good reader, medium reader, and poor reader. Those suffering of reading comprehension difficulties will receive the annotations of skilled peers.

The figure above (Fig. 1) illustrates the interaction between the various modules of "i-Read" system along with the flow of information/data. The system's architecture

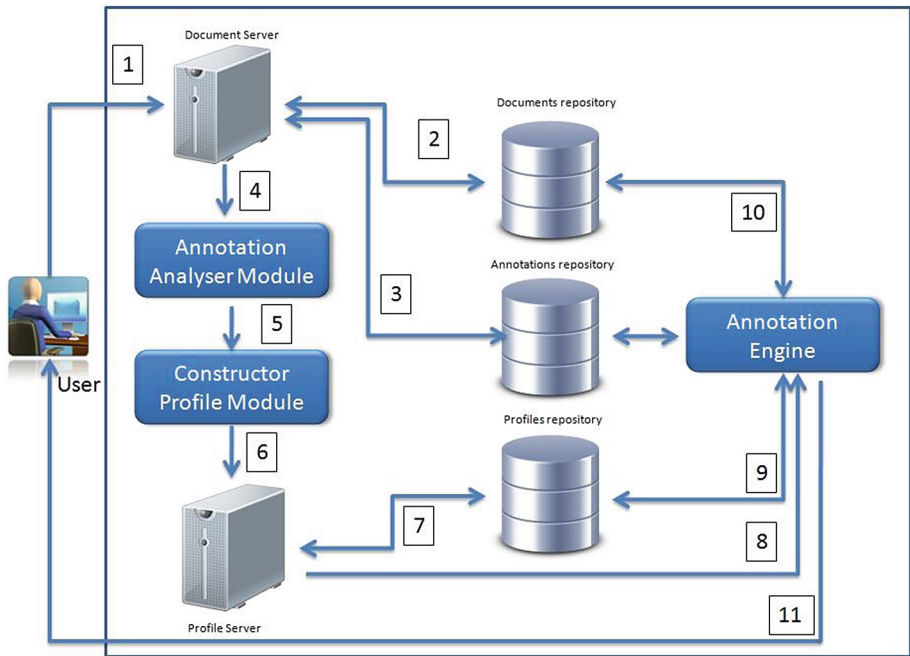


Fig. 1. Architecture of “i-Read”: collaborative online reading environment

consists of user annotation interface, the annotation analyzer module, the profile constructor module, the annotation engine and three databases with two servers. These components contribute to model learners’ personality profiles and derive consequently their reading abilities.

### 3.1 System Operation Procedure

Based on the system architecture (Fig. 1), the functional scenario of “i-Read” system is described and summarized as follows: (1) the learner connects to “i-Read” environment, uploads his/her document and starts reading and annotating; (2) the system saves the document in the Documents Repository; (3) the system saves learner’s annotations in the Annotations Repository; (4) the Annotation Analyser Module captures learner’s annotations and extracts certain features; (5) the Annotation Analyser Module sends the computed information to the Profile Constructor Module to build learner’s personality profile; (6) the Profile Constructor Module considers the received information as an input data to the Multivariate Linear Regression Algorithm used to estimate the scores of learner’s traits; According to the derived scores of learner’s traits, the Profile Module determines his level of reading performance (high, medium, poor); (7) the system saves the modeled user’s profile in the Profiles Repository; (8) for poor readers, the system sends a request to Annotation Engine to search the annotations of skilled readers; (9) the Annotation Engine searches the profiles of skilled readers; (10) the Annotation Engine

searches the annotated documents corresponding to the list of skilled readers compiled previously; (11) the Annotation Engine sends the annotated documents of skilled readers to the active learner who suffer of reading comprehension difficulties.

### 3.2 Evaluation of “i-Read” Environment

To evaluate our system, we invite 32 students (11 males and 21 females). The participants were first-year students from the computer science bachelors programs. They were in the second semester of year one when this study was conducted and were enrolled for a C Language Programming course at the High Institution of Computer Sciences and Management of Kairouan University. The course is a compulsory course for year one students. They had the basic knowledge of C language Programming, which they learned during their study of the first level of C language Programming course.



Fig. 2. Recommendation of skilled readers’ annotations to poor readers.

Based on the students’ annotations we construct their personality profiles and we classify them accordingly into three categories of readers: “Good Reader”, “Ordinary Reader” and “Poor Reader”.

In second step, we are interested to determine whether the reading abilities of poor readers will be improved, as the result of receiving the annotations of skilled readers. To do, we asked unskilled readers, which are judged implicitly by the system based on their personality profile, to summarize the read document.

The written summaries were evaluated by an expert in C language. In fact, the expert graded the summaries of the struggling students on a 0 to 20 point scale. A summary is considered acceptable when it is valued of a grade of 10 or above. The expert testifies if the written summary is concise and accurately represents the author’s



ideas and key points. Thereafter, we recommended annotations of skilled readers to poor readers (Fig. 2). Then, we asked those students to re-summarize the reading content to be evaluated again by the expert, knowing that writing a good summary demonstrates that the student clearly understand the reading document. We utilize the t-test statistical method to study if there is a significant difference between summaries of the first and the second tests.

## 4 Results and Discussion

The results presented in Table 1 indicated that the quality level of the poor students’ summaries written before recommendation of annotations (Mean = 7.38, SD = 2.19) was significantly lower than the quality of their summaries written after they received annotations of skilled readers (Mean = 9.63, SD = 2.7), with t-value (15) = -4.32, and p-value < .001.

**Table 1.** Quality level of readers’ summaries before and after annotations recommendation

Written summaries	Mean	SD	Diff.	t	p
First summaries	7.38	2.19	-	-4.32	0.0006
Second summaries	9.63	2.75	-2.25	-	-

These results are consistent with the findings of many research studies have shown that sharing annotation may foster the exchange of knowledge and learning experiences and has the potential to have a positive effect on reading outcomes. One shortcoming of current work is the sample size which is very limited. We expect in the future, to spread our online reading environment among students of different academic fields to assist them in their online reading activities.

## 5 Conclusion

In this research we present a new tendency to assist students having troubles of reading comprehension in online environment. Our contribution is twofold: first of all we try to assess the students reading abilities based on their personality profiles constructed with reference to their annotation activities. Secondly, we share annotations of expert readers with those suffering of reading problems. The experimental results show the potential role of annotation to enhance students’ learning experiences and their academic achievement, which is very promising and constitute a step forward to overcome students reading difficulties in distance learning context. As a future direction, we expect to zoom more on students’ online reading behaviors as a way to extract certain learning parameters (motivation, style of learning, interest, etc.) that help to assist them during their learning experiences.

## References

1. Zhao, L.: The teaching reform of strategies and skills in perspective of English reading: a case study of Chinese Mongolian students. *Engl. Lang. Lit. Stud.* **5**(1), 102 (2015)
2. Garrett-Rucks, P., Howles, L., Lake, W.M.: Enhancing L2 reading comprehension with hypermedia texts: student perceptions. *CALICO J.* **32**(1), 26 (2015)
3. Spörer, N., Brunstein, J.C.: Fostering the reading comprehension of secondary school students through peer-assisted learning: effects on strategy knowledge, strategy use, and task performance. *Contemp. Educ. Psychol.* **34**(4), 289–297 (2009)
4. Jan, J.C., Chen, C.M., Huang, P.H.: Enhancement of digital reading performance by using a novel web-based collaborative reading annotation system with two quality annotation filtering mechanisms. *J. Hum.-Comput. Stud.* **86**, 81–93 (2016)
5. Omheni, N., Kalboussi, A., Mazhoud, O., Hadjkacem, A.: Modelling learner's personality profile through analysis of annotation digital traces in learning environment. In: 2015 IEEE 15th International Conference on Advanced Learning Technologies, pp. 66–67. IEEE (2015)
6. Omheni, N., Mazhoud, O., Kalboussi, A., HadjKacem, A.: Prediction of human personality traits from annotation activities. In: Proceedings of 10th International Conference on Web Information Systems and Technologies, pp. 263–269. SciTePress (2014)
7. Leu, D.J., Forzani, E., Burlingame, C., Kulikowich, J., Sedransk, N., Coiro, J., Kennedy, C.: The new literacies of online research and comprehension: assessing and preparing students for the 21st century with common core state standards. In: Neuman, S.B., Gambrell, L.B. (eds.) *Quality Reading Instruction in the Age of Common Core Standards*, pp. 219–236. International Reading Association, Newark (2013)
8. Kiili, C., Laurinen, L., Marttunen, M., Leu, D.J.: Working on understanding during collaborative online reading. *J. Lit. Res.* **44**(4), 448–483 (2012)
9. Agosti, M., Ferro, N.: A formal model of annotations of digital content. *ACM Trans. Inf. Syst. (TOIS)* **26**(1), 3 (2007)
10. Marshall, C.C.: Annotation: from paper books to the digital library. In: Proceedings of 2nd ACM International Conference on Digital Libraries, pp. 131–140. ACM (1997)

# Integrating Support for Collaboration in a Computer Science Intelligent Tutoring System

Rachel Harsley<sup>1</sup>(✉), Barbara Di Eugenio<sup>1</sup>, Nick Green<sup>1</sup>, Davide Fossati<sup>2</sup>,  
and Sabita Acharya<sup>1</sup>

<sup>1</sup> Department of Computer Science,  
University of Illinois at Chicago, Chicago, IL, USA  
{rhars12,bdieugen,ngreen21,sachar4}@uic.edu

<sup>2</sup> Department of Math and Computer Science, Emory University, Atlanta, GA, USA  
davide.fossati@emory.edu

**Abstract.** Calls for widespread Computer Science (CS) education have been issued from the White House down and have been met with increased enrollment in CS undergraduate programs. Yet, these programs often suffer from high attrition rates. One successful approach to addressing the problem of low retention has been a focus on group work and collaboration. This paper details the design of a collaborative ITS (CIT) for foundational CS concepts including basic data structures and algorithms. We investigate the benefit of collaboration to student learning while using the CIT. We compare learning gains of our prior work in a non-collaborative system versus two methods of supporting collaboration in the collaborative-ITS. In our study of 60 students, we found significant learning gains for students using both versions. We also discovered notable differences related to student perception of tutor helpfulness which we will investigate in subsequent work.

**Keywords:** Collaboration · Collaborative intelligent tutoring system · Data structures · CS1 · CS2

## 1 Introduction

In recent years, ITS researchers have begun to explore outcomes of ITSs that support collaborative learning. Benefits of collaborative learning include increased group performance as well as individual performance. Moreover, collaborative problem solving is consistently associated with higher order thinking skills including planning, reflection, and metacognition [5]. The field of Computer Supported Collaborative Learning (CSCL) explores how students learn in collaborative settings and how technology can support this collaboration.

There are a plethora of methods for system design regarding pedagogical guidance, group formation, collaboration cues, and student modeling in order for ITSs to accommodate collaboration [3]. Thus, we distinguish collaboration

supported by a CIT in three primary ways: *unstructured* (initiated and maintained by students), *semistructured* or *fully structured* (moderately or strongly supported and guided by the CIT). This paper explores the role of the ITS in structuring collaboration by presenting findings from an empirical study in which students use the *unstructured* and *semi-structured* collaborative adaptations of a traditional ITS. We assess the effectiveness of the systems in terms of student learning gain and perceptions of the system. Findings are presented from a study with 60 students utilizing Collab-ChiQat Tutor, a collaborative ITS for computer science education. Results show that students using the *unstructured* system with minimal collaboration support, and the *semistructured* which provided collaboration feedback, both achieved significant learning gains.

## 2 Background

Longstanding research has shown that both cooperative and collaborative interactions among students are beneficial to learning [6]. However, assigning students to a group and charging them with a task does not ensure that students will engage in effective collaborative learning behavior [9]. Thus, CSCL requires careful construction of the collaboration so that interactions benefit the individual and group. One successful approach to improving collaboration has been the use of visualized group performance and peer assessments [4, 8].

Collaboration is also a core component of CS curriculum and accreditation requirements [1]. It is been utilized in both industry and academia through the growing practice of pair programming. In this methodology, two users share the same computer, keyboard and mouse. One user serves as the driver while the other serves as the navigator. The driver's roles is to write the code and control both keyboard and mouse. The navigator's role is to act as an external metacognizer who thinks about the direction of the code and helps the pair avoid possible pitfalls.

Recently, research efforts have focused on merging the affordances of both ITS and CSCL to capitalize on the benefits of group learning and adaptive support. Several researchers in the CSCL community are exploring how adaptivity, automated analysis, and feedback integrate into CSCL approaches [10]. Similarly, ITS researchers are extending their individual use ITS systems to accommodate collaborative support [7, 11].

## 3 Collab-ChiQat Tutor

This study both reconceptualizes and redevelops a non-collaborative tutoring system for CS Education, ChiQat-Tutor. In particular our work centers on the system's linked lists data structure lesson. A problem is presented to a student in both textual and graphical representation as shown in Fig. 1a. The student is then able to programmatically solve the problem. Moreover, the system provides relevant positive and negative feedback to the student in a manner analogous to the one-on-one human tutoring experience from which the system was derived.

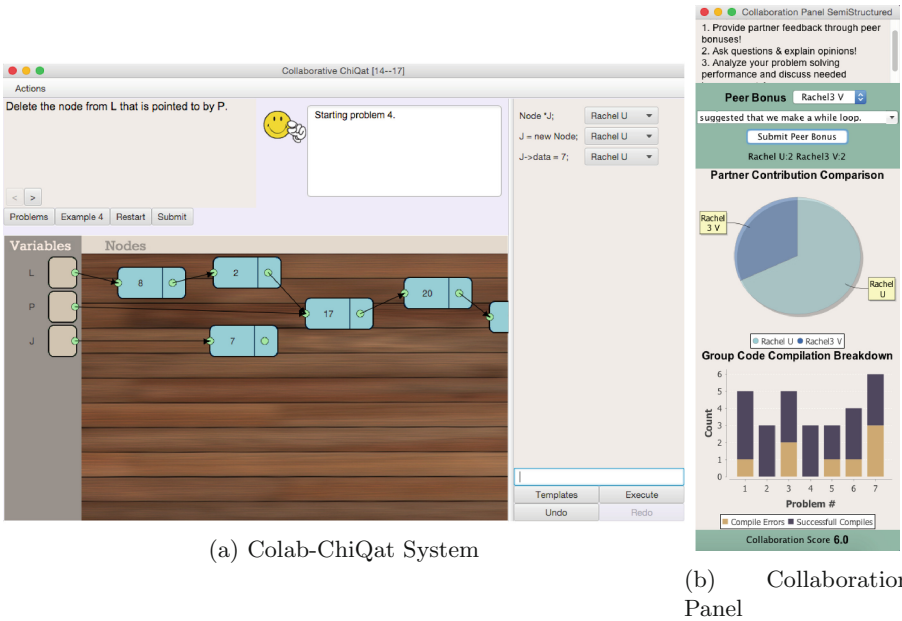


Fig. 1. Collab-ChiQat ITS for computer science education

Collab-ChiQat accommodates learning between pairs of students as they jointly engage with the system in pair programming. Collab-ChiQat maintains all of the major architectural components present in standard ChiQat. However, the collaborative system differs from the standard version in several ways including its student model, graphical user interface, and feedback.

In the *unstructured* version, students focus on CS domain learning with no system-provided support for their collaborative interaction. While in the *semi-structured* version, students focus on CS domain learning and have visualized representation of their participation and performance via the collaboration panel described below. Several newly introduced components for Collab-ChiQat are described below while our prior work sets forth existing components [2].

*Joint Student Model.* The joint student model works as the storehouse of information pertaining to a student's problem solving behavior and the state of the pairs' problem solving. The collection of information available in both the joint and individual student models is used to synthesize relevant and properly timed feedback. Information aggregated in the joint student module includes: history and timing of students' actions, feedback (*i.e.* number of positive/negative proactive/reactive feedback), undo/redo behavior, number of problem attempts and problems solved, individual and collective compile error and success rates, number of spoken utterances, peer bonus information.

*Graphical User Interface.* In *semistructured* Collab-ChiQat, a collaboration panel is introduced. The panel serves as the view for participation and group performance visualization and peer feedback as shown in Fig. 1b. The panel contains the following five components (1) tips on successful collaboration (2) pie chart comparison of number of spoken utterances between partners (3) bar graph comparison of number of compile errors vs successes per problem (4) peer bonus input w/sentence opener (5) overall group collaboration score.

## 4 Empirical Study

An experiment involving human participants was conducted in Fall of 2015 in a second year Computer Science programming course. Our experiments ran over four different sessions of the course. A total of 103 students used Collab-ChiQat during the study.<sup>1</sup> Students chose their own partners. Each pair was stationed at a single workstation and individually equipped with a headset. They were given 40 min to work with the system. Student interaction with the system was continually logged. Students were given an exit survey regarding their perception of Collab-ChiQat and their abilities, their attitudes towards CS and the course, and their understanding of successful pair programming traits.

Students were allowed 12 min to perform pre and post tests individually. Both pre and post tests are identical and derived from prior work analyzing human CS tutoring dialogues. We use the following measure of learning gain to assist in our analysis of learning:

$$gain = postTestScore - preTestScore \quad (1)$$

## 5 Results

Of foremost importance in evaluating the system is the answer to the question of whether or not students learned. In answer to the primary question, the students did learn. Overall, student post test scores were significantly better than pre-test scores ( $p < .05$ ). Moreover, the learning gain in the *unstructured* condition approaches both our best prior results for the single student ChiQat system as well as the human tutoring<sup>2</sup> condition as shown in Table 1. Note, this holds true despite students' higher prior knowledge, given pre-test scores.

Subsequent to learning gains, our aim was to understand student perceptions of the system as captured through the exit survey. We were especially interested in student perception of system helpfulness. Contrary to our hypothesis, we discovered that a greater majority of students in the *unstructured* system condition found the system to be helpful than in the *semistructured* condition.

<sup>1</sup> 43 students had used the non-collaborative ChiQat in a prior experiment. Their data is held out from learning gain analysis and reserved for further work.

<sup>2</sup> The human tutoring condition measured learning gains of students after one 40-minute session of working with an experienced human computer science tutor.

**Table 1.** Learning gains of students

Tutor	N	Pre-test		Post-test		Gain	
		$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$
Human	54	.40	.26	.54	.26	.14	.25
Non-collaborative ChiQat (Best)	23	.41	.18	.55	.22	.14	.17
<i>Unstructured</i> Collab-ChiQat	30	.48	.21	.60	.22	.12	.17
<i>Semistructured</i> Collab-ChiQat	30	.52	.26	.61	.24	.08	.18

Further, student were asked to describe three attributes of a good pair programming partnership. Phrases such as “hard work” and “hard” appeared multiple times in the *semistructured* condition student feedback but did not appear at all in the *unstructured* condition feedback.

## 6 Discussion

The findings indicate that collaborative learning in conjunction with an ITS can enhance student learning. Results showed significant learning for students using both the *unstructured* collaborative system and the *semistructured* condition, which provided collaboration feedback. The findings are a crucial step toward applying known CSCL techniques, including visualized participation and peer feedback, to an ITS. Analysis of student feedback showed that students found the *semistructured* system less helpful and harder to use. There are several possible reasons for this student perception. First, the *semistructured* interface, which visualized individual participation and group performance, may have caused students to experience cognitive overload. Secondly, students may have also been disincentivized to perform well if under the impression that they were given “hard work” by the addition of the collaboration panel.

Future work will incorporate students removed from this study due to their prior exposure to non-collaborative ChiQat. Investigation of their results may shed light on the student’s cognitive overload due to their increased familiarity with the overall system. Fine-grained analysis of interaction data including transcribed student interactions will also provide further insight regarding student perceptions of the system.

## 7 Conclusion

Collaborative Intelligent Tutoring Systems (CITs) offer a promising method to enhance student learning in adaptive and connected ways. In this paper, we detailed the design of an enriched architecture, a CIT for CS Education. In order to gain an understanding of the varying methods for supporting collaboration and their effect on learning, we compared two methods of structuring collaboration in a second year undergraduate CS course and analyzed student

learning gains and system perceptions. We discovered that students found the *unstructured* version of the system, which provided no visualization of collaborative and individual performance, to be more helpful. They also experienced significant learning gains. Similarly, students in the *semistructured* condition experienced significant learning gain, however they found the system to be less helpful despite the additional participation and performance visualization.

Additional research is needed to understand how modes of supporting collaboration affect learning and social participation. Our future work will examine reasons for the learning gain disparity, including the possibility of introduced cognitive overload given the visualized feedback. It will become increasingly important to understand how CITs can provide support for students to effectively collaborate and learn.

**Acknowledgments.** This work was supported by the Abraham Lincoln Fellowship 2015–2016 from the University of Illinois at Chicago, and grant NPRP 5–939–1–155 from the Qatar National Research Fund.

## References

1. Commission, A.C.A.: Criteria for Accrediting Computing Programs, 2016–2017 — ABET. ABET, November 2014
2. Green, N., AlZoubi, O., Alizadeh, M., Di Eugenio, B., Fossati, D., Harsley, R.: A scalable intelligent tutoring system framework for computer science education. In: Proceedings of the 7th International Conference on Computer Supported Education (CSEDU 2015) (2015)
3. Harsley, R.: Towards a collaborative intelligent tutoring system classification scheme. In: Proceedings of the 11th International Conference on Cognition and Exploratory Learning in the Digital Age (Celda 2014), pp. 290–291, Porto, Portugal, October 2014
4. Janssen, J., Erkens, G., Kanselaar, G., Jaspers, J.: Visualization of participation: does it contribute to successful computer-supported collaborative learning? *Comput. Educ.* **49**(4), 1037–1065 (2007)
5. Kaptelinin, V.: Learning together: educational benefits and prospects for computer support. *J. Learn. Sci.* **8**(3–4), 499–508 (1999)
6. Lehtinen, E., Hakkarainen, K., Lipponen, L., Rahikainen, M., Muukkonen, H.: Computer supported collaborative learning: a review. *JHGI Giesbers Rep. Educ.* **10** (1999)
7. Olsen, J.K., Belenky, D.M., Alevan, V., Rummel, N.: Using an intelligent tutoring system to support collaborative as well as individual learning. In: Trausan-Matu, S., Boyer, K.E., Crosby, M., Panourgia, K. (eds.) ITS 2014. LNCS, vol. 8474, pp. 134–143. Springer, Heidelberg (2014)
8. Phielix, C., Prins, F.J., Kirschner, P.A., Erkens, G., Jaspers, J.: Group awareness of social and cognitive performance in a CSCL environment: effects of a peer feedback and reflection tool. *Comput. Hum. Behav.* **27**(3), 1087–1102 (2011)
9. Soller, A.: Supporting social interaction in an intelligent collaborative learning system. *Int. J. Artif. Intell. Educ. (IJAIED)* **12**, 40–62 (2001)



10. Tchounikine, P., Rummel, N., McLaren, B.M.: Computer supported collaborative learning and intelligent tutoring systems. In: Nkambou, R., Bourdeau, J., Mizoguchi, R. (eds.) *Advances in Intelligent Tutoring Systems*. SCI, vol. 308, pp. 447–463. Springer, Heidelberg (2010)
11. Walker, E., Rummel, N., Koedinger, K.R.: Integrating collaboration and intelligent tutoring data in the evaluation of a reciprocal peer tutoring environment. *Res. Pract. Technol. Enhanced Learn.* **04**(03), 221–251 (2009)

# Wheel-Spinning in a Game-Based Learning Environment for Physics

Thelma D. Palaoag<sup>1</sup>(✉), Ma. Mercedes T. Rodrigo<sup>2</sup>,  
Juan Miguel L. Andres<sup>3</sup>, Juliana Ma. Alexandra L. Andres<sup>2</sup>,  
and Joseph E. Beck<sup>4</sup>

<sup>1</sup> University of the Cordilleras, 2600 Baguio City, Benguet, Philippines  
tpalaoag@gmail.com

<sup>2</sup> Ateneo de Manila University, 1108 Quezon City, Metro Manila, Philippines  
mrodrigo@ateneo.edu, alexandralandres@gmail.com

<sup>3</sup> Teachers College, Columbia University, New York, NY 10027, USA  
jla2183@tc.columbia.edu

<sup>4</sup> Worcester Polytechnic Institute, Worcester, MA 01609, USA  
josephbeck@wpi.edu

**Abstract.** We study wheel-spinning behavior among students using an educational game for physics. We attempted to determine whether students wheel-spin, and to build a wheel-spinning detector. We found that about 30 to 40 % of students are unable to successfully complete a level when attempting it 8 times or more, or when working on it for more than 160 s. We also found that past performance is predictive of wheel-spinning, and that persistence increases both the likelihood of success and of wheel-spinning. Finally, we found that wheel-spinning in this context is different from wheel-spinning exhibited in prior work in that it is relatively easy to detect and does not suffer from cold starts.

**Keywords:** Wheel-spinning · Physics playground · Persistence · Detector

## 1 Introduction

Persistence is a disposition, a habit of mind and action. In the context of this study, persistence is the ability to maintain an action, regardless of the person's feelings about achieving the task. It allows a person to keep taking action or pressing on even when he or she feels like quitting. [5] describes grit as tirelessness over the years to accomplish troublesome long-term objectives. Grit is portrayed with respect to stamina, stressing the part of exertion, interest, and passion in keeping focused on the objectives. In academic settings, these elements have a great impact on scholastic accomplishment or achievement.

While persistence is sometimes what distinguishes individuals who are successful and those who fail in an endeavor [5], a student's decision to persist or not is based on many factors. Early success or encouragement may result in the choice to persist in the face of difficult tasks. On the other hand, if a student's efforts are met with early failure, they may choose not to persist, opting instead give up or seek external help.

Persisting despite failure is not always productive or desirable. This type of persistence may in fact be wheel-spinning, a non-learning behavior coined by [3] referring to the failure to achieve mastery in a timely manner. Wheel-spinning denotes continuous, but futile effort. [4] found that wheel-spinning is probably related to knowledge deficits rather than boredom or other affective states. In this paper, we examine both persistence and wheel-spinning in the context of Physics Playground (PP), an educational game for physics. We attempt to (a) determine whether students wheel-spin within PP and, if they do, (b) build a detector for wheel-spinning.

## 2 Methods

### 2.1 Physics Playground

Physics Playground (PP) is a computer game designed to help high school students achieve a non-verbal conceptual understanding of how the physical world operates, characterized by an implicit understanding of concepts related to Newton's three laws: balance, mass, and conservation and transfer of momentum, gravity, and potential and kinetic energy. PP is described in detail in [1, 2].

**Performance Metrics.** Gold and silver badges are awarded to students who manage to solve a level. A gold badge is given to a student who is able to solve the level by drawing a number of objects equal to the particular level's par value (i.e., what the developers consider to be a reasonable number of objects needed to be drawn to solve the level). A student who solves a level using more objects will earn a silver badge. Many levels in PP have multiple solutions, meaning a player can solve the level using different agents.

### 2.2 Participant Profile

Data were gathered from 62 s year high school students, divided between one public school and one private school in Baguio City, Philippines. Participants ranged in age from 13 to 18 years old; 48 % were females and 52 % were males.

Thirty participants were from Bakakeng National High School (BNHS), located at Barangay Bakakeng Sur. The school has 7 instructional rooms and two 2 non-instructional rooms. With a total of 291 students, BNHS has a typical class size of around 42 students. Among the 30 participants, 15 came from honors sections, and 15 from regular sections.

The other 32 students were from University of the Cordilleras Grade and High School (UCGHS), located at Campo Filipino. 10 of the participants came from a star section, and the rest belonged to regular sections. Compared to other schools in the city, the UC grade school has maintained its tradition of academic excellence by winning in different interschool competitions.

### 2.3 Procedure

Participants were divided into batches of 15 to 17. Most batches of students played the game for 120 min; two batches played for only 90 min because they arrived at the testing venue late. As such, only the first 90 min of all the sessions were considered in this analysis.

### 2.4 PP Interaction Logs

We collected interaction log file data from 62 students but data from two students was corrupted. Hence, the succeeding analyses was based on data from 60 students only. These logs captured the following events:

- Menu Focus – an event that indicates the current playground and level,
- Level Start – an event that indicates that the player has begun playing a level,
- Level Restart – an event that indicates that the player triggered a level restart,
- Level End – an event that indicates that the player has finished playing a level,
- Time – an attribute of all events that pertains to the time the event was triggered,
- Object – an attribute of all events that pertains to the number of objects drawn, and
- Badges – an attribute of the Level End event that indicates what badge was given.

From these events, the following features we distilled: time spent on a level and number of restarts, both of which are considered by [6, 7] to be indicators of persistence. The number of restarts was considered to be equal to the number of attempts.

## 3 Wheel-Spinning

In prior work [3, 4], mastery was defined as three consecutive successful attempts at a skill. It was possible to adopt this as a criterion because the skills in the systems discussed in [3, 4] were traditional, structured tutoring systems with steps associated with defined skills. In PP, developers did not specify a mastery criterion. To answer our first research question, we did not attempt to associate specific skills with each level and we considered a level “mastered” once a student received either a gold or silver badge.

Per level, we noted how many attempts and how much time took for each student to achieve mastery. To compute the percentage of students who mastered a level, we counted the number of students who earned a gold or silver badge for a level, and then divided that number by the total number of students who attempted the level. We then computed for the average cumulative percentage of students who demonstrated mastery per number of attempt and over time.

The results are similar to those in [3, 4]. The cumulative percentage of students who achieve mastery plateaus. About 60 % of students master the skill after three attempts. After 8 attempts, there is almost no increase in the number of students who achieve mastery. Additional results show that about 60 % of students achieve mastery after about 80 s of working on a level. After 160 s, the number of students who master a level does not improve. It is therefore reasonable to conclude that at least some of these students were wheel-spinning.

## 4 Wheel-Spinning Detector

The first step in detecting wheel-spinning is to crisply define it for the PP context. Part of the difficulty in the task is that students can acquire either a gold or a silver badge. If a student acquires a silver badge after four attempts and 100 s of work, and struggles for 20 more minutes without earning a gold badge, was he wheel-spinning or not? The student made some progress, but the majority of the time was in engaged in behavior we would probably categorize as wheel-spinning.

For our detector, we adopted the following definition of wheel-spinning:

1. All attempts after 15 min on a level were presumed to be wheel-spinning and were discarded, including any badges attained.
2. All attempts leading up to the first gold badge were not wheel-spinning, as the student made progress.
3. All attempts after the first gold badge were removed, as we were unsure how to score student performance. The student had already maxed out performance from the standpoint of the Tutor, and could have been experimenting for personal knowledge. There were 273 student-level attempts after receiving a gold badge.
4. All attempts leading up to the first silver badge were not wheel-spinning, as the student was making progress.
5. All attempts after the first silver badge that led to a gold badge, would be categorized as not wheel-spinning (this rule is a special case of rule #2).
6. All attempts after the first silver badge that did not lead to a gold badge were categorized as wheel-spinning.

Note that this framework categorizes some student effort on a problem as productive while later work can be wheel spinning. Reusing the example from the first paragraph, the students first four attempts would be considered not wheel-spinning, as it led to the student earning a silver badge. All subsequent attempts, up until a maximum of 15 min, would be labeled as wheel-spinning. For students who received a silver badge, there were 810 additional attempts at a level after. Unfortunately, only 50 of those attempts (6.2 %) were successful at making additional progress and receiving a gold badge. Surprisingly, students were more likely to achieve a gold badge if they had not already achieved a silver badge. One possible explanation is that stronger students realize they are not going to get a gold medal and so restart the level, while weaker students are happy to get any badge.

To predict wheel-spinning, we used the following features:

1. The number of prior attempts the student has made to solve this level.
2. The cumulative amount of time (before this attempt), in seconds, the student has spent on this level.
3. Looking at past performance for this student, the probability he receives a gold medal in less than 5, 10, and 15 min (i.e., 3 features).
4. Similarly, based on past performance, the probability this student has received a silver medal in 5, 10, and 15 min (i.e., 3 features).
5. Whether the student has already earned a silver medal on this level.

To train the model, we used data from 14,232 student attempts at solving a level. We performed 10-fold cross validation, ensuring that each student's data was entirely within one of the folds. We trained a logistic regression model both for its interpretability, and for consistency with [3] in comparing the predictability of wheel-spinning. We experimented with a variety of temporal thresholds (5, 10, and 15 min) in case relatively stronger student performance, indicated by earning medals within 5 min, would be a stronger negative influence on wheel-spinning. However, the probability the student would receive a silver medal within 15 min was the best predictor. Spending additional time on the level unsurprisingly increased the probability of wheel-spinning. Interestingly, earning a silver medal on the level increased the likelihood of wheel-spinning. One explanation is that obtaining a gold medal is fairly difficult, and students attempting it were likely to get stuck.

The detector had fairly strong performance with an AUC of 0.853, and achieved 82.9 % correct in its predictions (vs. a base rate of 73.4 % correct).

The detector does a fairly good job at quickly detecting which students are unlikely to make progress on the current problem. As the student works longer within the problem, the tutor gains some additional information in terms of how long the student has spent, and whether or not he has earned a silver badge on the level. As a practical matter, performance when a student initially starts on a problem is just over 82 %, noticeably higher than the baseline of 73.4 % for guessing majority class. The detector does not suffer from a cold start problem.

## 5 Future Work, Contributions, and Conclusions

Next steps for this work include better analysis of the objects created by the student and level restart behavior. A better understanding of how students interact with the game will aid both detection of wheel-spinning and other pedagogical interventions.

This paper contributes to the ITS literature in at three ways. First, it demonstrates that the incidence of wheel-spinning is about the same within a game-based learning environment as it is in more traditional intelligent tutoring systems. About 30 to 40 % of students require additional intervention in order to help them towards mastery. Second, it shows that past performance is predictive of wheel-spinning and persistence. While increasing likelihood to succeed at a level, past performance also increases the probability of wheel spinning. Third, we identified that wheel-spinning in PP is different compared to wheel-spinning exhibited in ASSISTments and the Scatterplot Tutor [4]. Wheel-spinning in PP is relatively easy to detect, and does not suffer from the cold start problem seen in other work. Therefore, augmenting the tutor with an intervention to discourage students from wasting time should be straightforward.

In conclusion, this paper presents a first attempt at determining whether wheel-spinning behavior exists in Physics Playground. PP is an open-ended environment, and differs greatly from traditional ITS where wheel-spinning analyses have been done previously. We found that wheel-spinning exists, and that its emergence is non-random as it is predictable with our classifier. Determining how to utilize this detector is our next step.

**Acknowledgements.** We thank the Ateneo de Manila University, the Ateneo Laboratory for the Learning Sciences, Ryan S. Baker, Valerie Shute, Matthew Ventura, Matthew Small, Jaclyn Ocumpaugh, Jessica Sugay, Michelle Banawan, Yancy Vance Paredes, and Nicko Regino Caluya. We also thank the Department of Science and Technology’s (DOST) Philippine Council for Industry, Energy, and Emerging Technology Research and Development for the grant entitled “Stealth assessment of student conscientiousness, cognitive-affective states, and learning using an educational game for Physics.”

## References

1. Andres, J.M.L., Rodrigo, M.M.T.: The Incidence and persistence of affective states while playing Newton’s playground. In: 7th IEEE International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (2014)
2. Andres, J.M.L., Rodrigo, M.M.T., Baker, R.S., Paquette, L., Shute, V.J., Ventura, M.: Analyzing student action sequences and affect while playing physics playground. In: International Workshop on Affect, Meta-Affect, Data and Learning (AMADL 2015), p. 24 (2014)
3. Beck, J.E., Gong, Y.: Wheel-spinning: students who fail to master a skill. In: Lane, H., Yacef, K., Mostow, J., Pavlik, P. (eds.) AIED 2013. LNCS, vol. 7926, pp. 431–440. Springer, Heidelberg (2013)
4. Beck, J., Rodrigo, M.T.: Understanding wheel spinning in the context of affective factors. In: Trausan-Matu, S., Boyer, K.E., Crosby, M., Panourgia, K. (eds.) ITS 2014. LNCS, vol. 8474, pp. 162–167. Springer, Heidelberg (2014)
5. Duckworth, A.L., Peterson, C., Matthews, M.D., Kelly, D.R.: Grit: perseverance and passion for long-term goals. *J. Pers. Soc. Psychol.* **92**(6), 1087–1101 (2007)
6. Shute, V.J., Ventura, M., Kim, Y.J.: Assessment and learning of qualitative physics in Newton’s playground. *J. Educ. Res.* **106**(6), 423–430 (2013)
7. Shute, V., Ventura, M.: *Stealth Assessment: Measuring and Supporting Learning in a Video Games*. MIT Press, Cambridge (2013)

# Using Multi-level Modeling with Eye-Tracking Data to Predict Metacognitive Monitoring and Self-regulated Learning with CRYSTAL ISLAND

Michelle Taub<sup>1</sup>(✉), Nicholas V. Mudrick<sup>1</sup>, Roger Azevedo<sup>1</sup>,  
Garrett C. Millar<sup>1</sup>, Jonathan Rowe<sup>2</sup>, and James Lester<sup>2</sup>

<sup>1</sup> Department of Psychology, North Carolina State University,  
Raleigh, NC, USA

{mtaub, nvmudric, razeved, gcmillar}@ncsu.edu

<sup>2</sup> Department of Computer Science, North Carolina State University,  
Raleigh, NC, USA

{jprowe, lester}@ncsu.edu

**Abstract.** Studies investigating the effectiveness of game-based learning environments (GBLEs) have reported the effectiveness of these environments on learning and retention. However, there is limited research on using eye-tracking data to investigate metacognitive monitoring with GBLEs. We report on a study that investigated how college students' eye tracking behavior ( $n = 25$ ) predicted performance on embedded assessments within the CRYSTAL ISLAND GBLE. Results revealed that the number of books, proportion of fixations on book and article content, and proportion of fixations on concept matrices—embedded assessments associated with each in-game book and article—significantly predicted the number of concept matrix attempts. These findings suggest that participants strategized when reading book and article content and completing assessments, which led to better performance. Implications for designing adaptive GBLEs include adapting to individual student needs based on eye-tracking behavior in order to foster efficient completion of in-game embedded assessments.

**Keywords:** Metacognition · Self-regulated learning · Game-based learning · Eye tracking · Process data · Scientific reasoning

## 1 SRL, Metacognitive Monitoring, and Game-Based Learning

Research on self-regulated learning (SRL) has revealed that processes related to metacognitive monitoring and control are effective for learning with advanced learning technologies, such as intelligent tutoring systems (ITSs) and game-based learning environments (GBLEs) [1]. GBLEs have been shown to be effective for learning complex topics during gameplay [2] while keeping students engaged in a learning task, particularly when designed to foster various aspects of SRL [3]. These environments have been developed to afford opportunities to engage in scientific reasoning and problem



solving [4], and studies have found that GBLEs are often more effective than traditional teaching methods, in terms of learning and retention [5, 6].

Despite the growing evidence indicating that GBLEs lead to improved learning outcomes [5], prior research on metacognitive monitoring and SRL within GBLEs has primarily focused on classifying SRL behaviors and relating them to in-game behavior and learning outcomes [3, 6]. In this paper, we aim to integrate how we can use trace data to track students' metacognitive monitoring and SRL to assess performance on concept matrices, an in-game embedded assessment tool within CRYSTAL ISLAND. We investigated if students were using metacognitive monitoring strategies, as indicated by their log file and eye tracking behavior during knowledge construction activities (e.g., reading) related to scientific reasoning to perform successfully on embedded measures of text comprehension (e.g., completing in-game concept matrices), as evidence of SRL and scientific reasoning in GBLEs. Multi-level modeling (MLM) is an ideal analytical technique to assess student learning with GBLEs because it enables statistical analyses of learning events at nested levels of abstraction that do not require restrictive statistical assumptions [7].

## 2 Method

35<sup>1</sup> undergraduate students from North Carolina State University (50 % female), with ages ranging from 18 to 29 ( $M = 20.18$ ,  $SD = 2.38$ ), participated in the study. Prior to beginning the study, the students were randomly assigned to one of three experimental conditions. Students were compensated for their participation in the study, receiving \$10 per hour, up to a total of \$30 for full participation.

CRYSTAL ISLAND is a 3D game-based learning environment designed to foster students' self-regulated learning, problem solving, scientific reasoning, and literacy skills [8]. When participants begin to play CRYSTAL ISLAND, they are informed of an outbreak that has impacted a group of scientists on the remote island. The student's task is to identify the epidemic that has spread amongst the scientists, determine the disease's transmission source, and recommend a treatment and prevention plan for the island's inhabitants. To do so, participants explore the virtual environment from a first-person perspective, navigating between five different buildings on the island: an infirmary, a living quarters, a dining hall, a laboratory, and the lead scientist's residence. These activities contribute to students' engaging in scientific reasoning, which involves hypothesis generation and testing, followed by forming conclusions based upon gathered test results [4]. Participants engaged in scientific reasoning when playing CRYSTAL ISLAND; they generated hypotheses based on the clues they gathered from non-player characters, reading books and articles, viewing posters, and testing their hypotheses about the spreading disease's transmission source. In order to complete the game, participants must submit a correct diagnosis.

---

<sup>1</sup> Data from 25 participants were included in this analysis because the other participants were in the *No Agency* condition (see Results section below).

One assessment tool, the concept matrix, was embedded into gameplay, such that there was a concept matrix to complete with every book or research paper the participant read (Fig. 1). The matrices contained questions regarding the book content in multiple-choice format. Participants were not restricted to answer the questions without returning to the text (i.e., they did not have to memorize the content). In addition, participants were given three attempts at completing the questions in the concept matrix, and if they failed to answer the questions correctly after three attempts, the game auto-filled the responses for participants (to ensure that they were eventually provided with the correct answer and were given potential information needed to help solve the mystery). Concept matrices are used to assess students' understanding of scientific concepts introduced in the reading material within the game.

When students played CRYSTAL ISLAND, we collected multi-channel SRL process data, including (1) software log files and (2) eye-tracking data. The log-file data captured student interactions with the game environment, including timestamp, action type, location, object, and characters involved in the interaction. The eye-tracking data provided gaze patterns and fixation behaviors on predefined areas of interest (AOIs) in the game, such as fixation duration on book content and fixation duration on concept matrices. To code and score the data, the number of concept matrix submission attempts (dependent variable) was calculated from the software log data capturing these events ( $M = 1.03$ ,  $SD = .75$ , across all three conditions). We used three predictor variables for this analysis. The number of books and articles read was extracted from the log files. This variable was calculated based upon the total number of books that participants selected throughout gameplay ( $M = 24.57$ ,  $SD = 8.57$ , across all three conditions). The other two variables were extracted from the eye-tracking data: (1) the proportion of time fixating on book content, and (2) the proportion of time fixating on an associated concept matrix. These variables were calculated by dividing the fixation duration of each activity over the total book fixation duration, yielding one proportion for fixation duration on book content ( $M = .33$ ,  $SD = .22$ , across all three conditions), and one proportion for fixation duration on book concept matrices ( $M = .19$ ,  $SD = .13$ , across all three conditions). Once calculated, these data were used to address the research questions posed for this analysis.

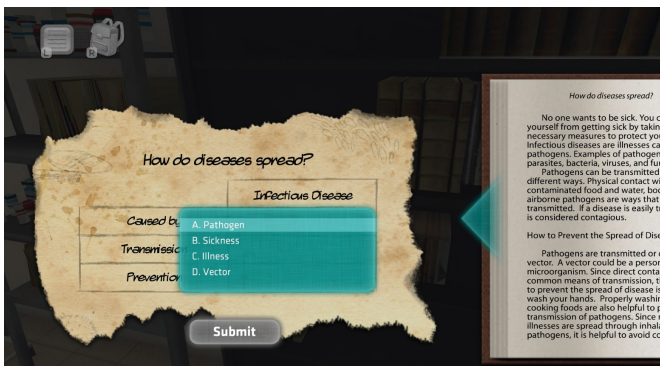


Fig. 1. Screenshot of a virtual book and associated concept matrix in CRYSTAL ISLAND.

### 3 Results

Prior to gameplay, participants were randomly assigned to 1 of 3 experimental conditions (*No Agency*, *Partial Agency*, or *Full Agency*). However, for this study, we included data from only the 2 interactive conditions, because the *No Agency* condition did not allow participants to select books to read, nor did participants in this condition complete concept matrices; students simply watched an expert player perform these activities. Thus, we only analyzed data from 25 participants, with  $n = 12$  for the *Full Agency*, and  $n = 13$  for the *Partial Agency* condition.

For this study we used multi-level modeling [7]. We ran three separate models, each with the same dependent variable: the number of concept matrix attempts. This required only one fully unconditional model to be run. Results from the fully unconditional model revealed there was significant between-subjects ( $\tau_{00} = .03$ ,  $z = 2.01$ ,  $p = .02$ ) and within-subjects ( $\sigma^2 = .52$ ,  $z = 16.10$ ,  $p < .0001$ ) variance in the number of concept matrix attempts, with 5.6 % variance between-subjects, and 94.4 % variance within-subjects. Thus, this model indicated that it was appropriate to continue to run models with predictor variables, as was done for the following research questions.

#### 3.1 Research Question 1: Is There an Association Between the Number of Books Read and the Number of Concept Matrix Attempts?

To address this research question, we ran a means-as-outcomes regression model with constrained slopes, with the number of books as the predictor variable (between-subjects, level 2) and the number of concept matrix attempts as the dependent variable. Results indicated that an increase in the number of books was associated with a decrease in the number of concept matrix attempts;  $\gamma_{10} = -.02$ ,  $t = -5.56$ ,  $p < .0001$ . This model explained 100 % of the between-subjects variance in number of concept matrix attempts. In general, this finding indicates that as participants were selecting more books to read, they were making fewer concept matrix attempts, indicating that as they were reading more books, they were performing better on the concept matrices associated with each book.

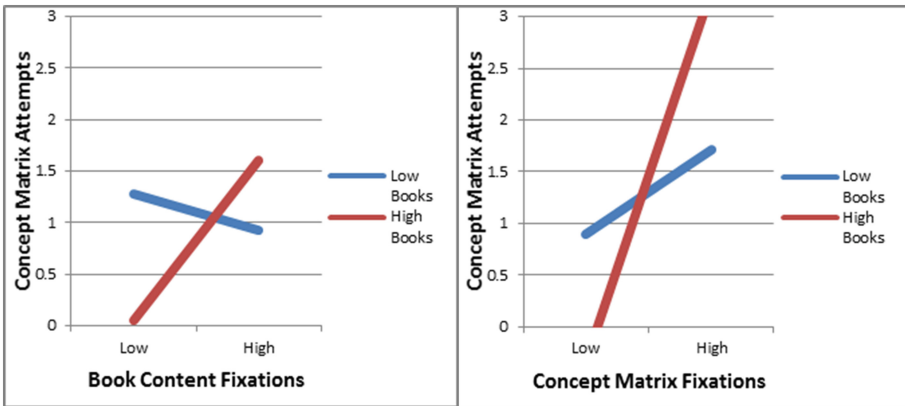
#### 3.2 Research Question 2: Is There a Relationship Between Concept Matrix Attempts and Proportion of Fixations on Book Content, and Does This Relationship Depend on the Proportion of Fixations on Book Concept Matrices?

For this research question, we ran a level 1 moderation model with constrained slopes, with concept matrix attempts as the dependent variable, and the proportion of fixations on book content and the proportion of fixations on book concept matrices as the predictor variables (within-subjects, level 1). Results indicated that the proportion of fixations on book content was not associated with concept matrix attempts ( $\gamma_{10} = .06$ ,  $t = .27$ ,  $p = .79$ ), nor was there an association between the proportion of fixations on book concept matrices and concept matrix attempts ( $\gamma_{20} = .07$ ,  $t = .21$ ,  $p = .84$ ).

However, there was a significant interaction;  $\gamma_{30} = 8.03, t = 6.92, p < .0001$ , such that participants with the fewest concept matrix attempts had the lowest proportion of fixations on book content, as well as on book concept matrices. This model explained 18.1 % of the within-person variance in concept matrix attempts. This finding indicates that a lower amount of fixation durations on both book content and concept matrices resulted in better performance on the concept matrices, such that spending more time reading the content and concept matrices did not result in better performance on the matrices.

**3.3 Research Question 3: Does the Relationship Between Concept Matrix Attempts and Number of Books Read Depend on the Proportion of Fixations on Book Content and on the Proportion of Fixations on Book Concept Matrices?**

This final research question used a 3-way cross-level interaction model with constrained slopes, with concept matrix attempts as the dependent variable and all three predictor variables used in the previous analyses (i.e., number of books – level 2 variable, proportions of fixations on book content and book concept matrices – level 1 variables). Results revealed a significant 3-way cross-level interaction;  $\gamma_{31} = .26, t = 2.16, p = .03$ , such that participants who had the least amount of concept matrix attempts read more books and had lower proportions of fixations on book content (Fig. 2, left) and on book concept matrices (Fig. 2, right). This model accounted for 19.3 % of the within-person variance in concept matrix attempts. Overall, these findings reveal that reading more books led to better performance on the concept matrices, however this was in combination with spending less time reading the content and concept matrices associated with each book.



**Fig. 2.** Interaction between fixations on book content and number of books (left) and book concept matrices and number of books (right), each on the number of concept matrix attempts (Color figure online).

## 4 Conclusions and Future Work

In this study, we used MLM to explore the links between theory (SRL and metacognitive monitoring), data channels (eye tracking and log files), and performance on an in-game assessment tool to examine how students used cognitive and metacognitive processes (reading comprehension) during knowledge construction activities related to scientific reasoning (completing the concept matrix), to provide evidence of SRL and scientific reasoning with the CRYSTAL ISLAND GBLE. Results indicated that these activities did significantly predict the number of concept matrix attempts, such that selecting more books was associated with fewer matrix attempts. However, fixating on more book content and fixating on more concept matrix content was associated with more matrix attempts, with fewer attempts as a more desirable outcome.

From this analysis, we cannot determine the sequence of events, such that we cannot confirm participants were transitioning from looking at the specific questions in the matrix, and finding those responses in particular areas within the text. Therefore, we cannot conclude that students were engaging in strategic reading, based on accurate monitoring, however, future studies will investigate the sequential order of reading books and completing their concept matrices in order to test this hypothesis. In particular, the use of analytical techniques that are amenable to sequence data show especial promise, such as sequence mining [9].

These results have important implications for designing intelligent GBLEs that afford students the opportunities to engage in cognitive, affective, metacognitive, and motivational processes to foster learning and scientific reasoning. Additionally, including adaptive scaffolding can improve the success of these environments in fostering learning during gameplay, such that they can use eye tracking to provide tailored scaffolding based on student strategy use. Improving the intelligence and efficiency of GBLEs can be beneficial to ensure that each student's real-time cognitive and metacognitive learning needs are being met, while still enjoying learning during gameplay.

**Acknowledgments.** This study was supported by funding from the Social Sciences and Humanities Research Council of Canada. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Social Sciences and Humanities Research Council of Canada.

## References

1. Azevedo, R., Aleven, V. (eds.): *International Handbook of Metacognition and Learning Technologies*. Springer, Amsterdam (2013)
2. Lester, J., et al.: Serious games get smart: intelligent game-based learning environments. *AI Mag.* **34**, 31–45 (2013)
3. Sabourin, J.L., Lester, J.C.: Affect and engagement in game-based learning environments. *IEEE Trans. Affect. Comput.* **5**, 45–56 (2014)
4. Spires, H.A., et al.: Problem solving and game-based learning: effects of middle grade students' hypothesis testing strategies on learning outcomes. *J. Educ. Comput. Res.* **44**, 453–472 (2011)

5. Wouters, P., et al.: A meta-analysis of the cognitive and motivational effects of serious games. *J. Educ. Psychol.* **105**, 249–265 (2013)
6. Rodeghero, P., et al.: Improving automated source code summarization via an eye-tracking study of programmers. In: *Proceedings of the 36th International Conference on Software Engineering 2014*, pp. 390–401. ACM, New York, NY (2014)
7. Raudenbush, S.W., Bryk, A.S.: *Hierarchical Linear Models: Applications and Data Analysis Methods*, 2nd edn. Sage, Thousand Oaks (2002)
8. Rowe, J.P., et al.: Integrating learning, problem solving, and engagement in narrative-centered learning environments. *Int. J. Artif. Intell. Educ.* **21**, 115–133 (2011)
9. Kinnebrew, J.S., et al.: Analyzing the temporal evolution of students' behaviors in open-ended learning environments. *Metacognition Learn.* **9**, 187–215 (2014)

# The Mobile Fact and Concept Training System (MoFaCTS)

Philip I. Pavlik Jr. <sup>(✉)</sup>, Craig Kelly, and Jaclyn K. Maass

Institute for Intelligent Systems, University of Memphis, Memphis, TN 38152, USA  
{ppavlik, cnkelly, jkmaass}@memphis.edu

**Abstract.** The effectiveness of Intelligent Tutoring Systems (ITS) research is enhanced by tools that allow researchers to quickly bridge the divide between theoretical and applied work. By providing a common infrastructure to test cognitive and learning science theories in authentic contexts with real students, the Mobile Fact and Concept Training System (MoFaCTS) can aid in accelerating ITS research and real world implementation. MoFaCTS is run from a web browser and allows the teacher or administrator to set up a sequence of units of content. Because the “optimal practice” module is interchangeable, the system allows for the comparison of alternative methods of adaptive practice. To foster faster research progress, data export supports the DataShop transaction format, which allows quick analysis of data using the DataShop tools. Integration with Amazon Turk allows quick and efficient data collection from this source.

**Keywords:** Intelligent tutoring systems · E-learning · Instructional design

## 1 Introduction

MoFaCTS was based on the FaCT system, which was created to make faster progress on laboratory research and its translation to the classroom [1]. MoFaCTS is the latest implementation of the FaCT system, which has new features in addition to running in HTML5, which provides mobility to any common web browser. The framework of MoFaCTS is based on an implicit theory of “chunk” learning [2] which assumes that learning of chunks occurs through discrete “trials” (e.g. a single step problem or fill-in-the-blank sentence). As such it departs from the tradition of model tracing tutors [3], which focus on multistep problems of greater complexity, where the student is learning a sequence of rule applications. The simplified chunk-based approach in MoFaCTS allows the system to focus more clearly on the problem selection aspect of tutoring, and how the selected sequence can be improved. Moreover, from the beginning the system was designed without strong assumptions about the optimal schedule. Because of this the system is easy to adapt to the needs of specific projects. Screenshots of the system in action, see Fig. 1, show a variety of functions, including multiple choice responding, fill-in-the-blank responding with branched feedback, and image-stimuli items.

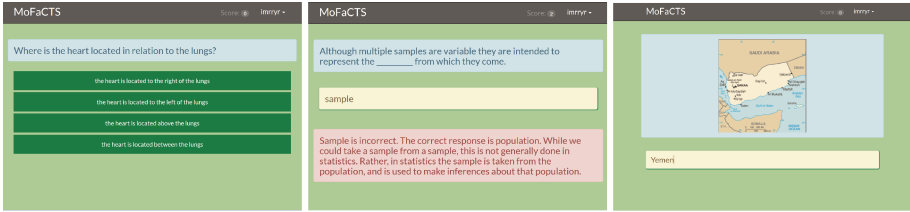


Fig. 1. Example screenshots.

## 2 Client/Server Architecture

MoFaCTS was built using Meteor, a framework based on Node.js which uses a single programming language (JavaScript) for both the client and server logic [4–6]. Communication between the two sides of the architecture is handled transparently by the framework. This architecture conveniently off-loads any complex computations needed to compute practice schedules to the client machine, which allows much larger numbers of users to interact with the system simultaneously.

Since the web is a popular platform for content and application delivery, MoFaCTS is able to leverage a vast body of open source software. Currently this includes the Bootstrap CSS framework developed by Twitter [7]. Bootstrap provides MoFaCTS with a typographic and layout framework. Most importantly, Bootstrap contributes responsive web design, so content appears correctly on both desktop web browsers and mobile devices [8]. The HTML displayed to the user is generated via Meteor’s templating system. This system uses a style known as “reactive programming” [9]. A piece of code can change data on the server and trigger a change on the client. This style of programming works well with the frequent switches in display necessary when sequencing multiple educational content objects.

The server portion of MoFaCTS runs on the Node.js JavaScript server, which gives it access to all of the libraries and asynchronous communication abilities of Node.js. In addition, the Meteor framework simplifies the server code required for an application of this type. When the client contacts the server, Meteor transparently provides data transport, call batching, and automated retry. This functionality leads to excellent performance without the need for more complex software. This architecture is particularly useful given the need for frequent logging to maintain the database of users’ learning histories. This MoFaCTS data is stored in a document-oriented database system named MongoDB [10]. In addition, MongoDB is schema-free, which allows for rapid iteration when designing new features or upgrades, which is often a great advantage in research.

## 3 Functionality

MoFaCTS has two primary unit types, learning and assessment, which define its two main modes of application. Both kinds of units are specified in the control file for each



“tutor”, which is called the tutor definition file (TDF). Each tutor definition file begins with a number of preliminaries, including the initial randomization commands. These randomizations allow the specification of shuffle and swap commands. The shuffle command generates random orders within groups of specified sequences (e.g. “<shuffle> 0–5 6–11” shuffles the first 6 items among themselves and then shuffle the next six items among themselves). The swap command randomizes the order of those specified sequences (e.g. “<swap> 0–5 6–11” randomizes the order of the groups of sequences, so in this simple case it would be either 0–5 6–11 still or 6–11 0–5, while retaining the order of the subsequences). By running a shuffle and then a swap command, complex distributions of stimuli across conditions can be achieved. To enable comparisons of different assessment or learning conditions, the system also automatically randomizes into any number of between-subjects conditions. This choice is recorded in the data for each subject, and reinstated when they begin new sessions from the same root TDF, so multi-session between-subjects comparisons with counter-balancing are easily enabled.

### 3.1 Units

The first main type of unit is the assessment unit, which allows for complex schedules of content, where the TDF author has specified the number of repetitions and the location in the sequence for each repetition of each item. Each repetition may be a test with or without feedback or a passive study opportunity. Assessment units may be used for quizzes in a classroom setting or for experiments looking at practice, forgetting, learning, and/or recall. In an experimental context, the system allows additional sequence level randomization, to make sure blocks of the same items are individually randomized, so that spacing conditions are not predictable. Any number of assessment units can be strung together, which allows pretest, practice and posttest portions to be organized individually to compose a larger experiment. Because the data architecture (described below) saves the state of the learner at all times, assessment sessions are automatically resumed where they were previously stopped, allowing for multiple experimental sessions for the same experiment over days or weeks.

The system also allows the specification of units with dynamic scheduling based on a select function. These units sequence the items according to a mechanism in the select function. This select function could be based on any sort of model of the learning and/or pedagogical rules. Typically, the adaptive learning module would be some version of Pavlik’s optimal learning method [11], which uses a computational model of memory to infer the best item to practice next.

Although assessment and learning units can both provide brief instructional screens prior to practice, an “instructions unit” presents only instructions, with a continue button to move to the next unit. These instructional units can also be configured with a between-subjects randomization into a “lockout condition” where the instruction screen has an active timer that only allows continuation after a specified amount of time.

### 3.2 Supported Practice Types

The system supports two forms of test items: the multiple choice items (which appear in button form for touchscreen responsiveness) and the short-answer items. For multiple choice items, the system allows the researcher to randomly display the order of two or more answer options. These answer options may be specified for each individual question, or be randomly selected from a larger “answer bank.” Feedback can be displayed after incorrect responses. This feedback displays the correct answer for a fixed period of time or until the user hits the spacebar, as specified in the TDF. If the trial is a short answer item (multiple choice branching is in development), more complex branching feedback is enabled which compares the response with a number of wrong responses, each of which has specific feedback text in the stimulus file. This allows the system to provide feedback tailored to particular response errors, hopefully promoting conceptual learning by directly challenging misconceptions in the student’s model of the domain.

Since both the system and the user may be frustrated and deterred in their goals by an incorrectly marked response, the system provides a few ways to identify correct responses with some flaws or ambiguity. These include partial matching using regular expressions, simple Levenshtein proportion errors, or Levenshtein proportion for multiple synonyms. Each of these methods offers different advantages depending on the test type. Regular expressions allow answer specification to pick up the presence of key words for short answer responses, to automatically score relatively complex responses (see the Circulatory System example below). Levenshtein proportion marks an item correct if some proportion of the letters are correct (e.g. 75 %).

Finally, a passive viewing trial type simply presents the stimulus (text, audio, or image) for a fixed number of seconds or until the user hits the spacebar. Normally, a fixed time is used, since unless the user population is intrinsically motivated, the students or participants may truncate these study trials, reducing (possible) learning.

### 3.3 Datashop Export and Amazon Turk Integration

The system provides native export to the PSLC DataShop tab-delimited format style with several custom fields. This functionality means that data collected in the system can be immediately imported into DataShop for analysis, storage, and/or presentation [12]. As part of the new LearnSphere project the DataShop is being expanded to include a graphical workflow analysis tool with multiple methods (<http://learnsphere.org/>). MoFaCTS users will be able to take advantage of these resources immediately. Further, there is a library of prior analyses already shared within the community for DataShop formatted files (<https://pslcdatashop.web.cmu.edu/ExternalTools>).

The system provides integration with Amazon’s Mechanical Turk (MTurk) service. This integration was added to ease the administrative burden often encountered when running experiments with large numbers participants recruited via MTurk. A researcher can oversee the experiment via a management screen within MoFaCTS that shows the current progress of all participants. From the same screen, the researcher may approve payment for a participant’s work and/or pay a post-payment bonus. If using the “lockout conditions” discussed previously, researchers may craft an automated message that the

system will send to Mechanical Turk users when their lockout expires (e.g., email a reminder after a one-week retention interval).

## 4 Research Using MoFaCTS

Described here are three recent experiments (one published as a dissertation, one in preparation, and one submitted for publication, respectively), using MoFaCTS with different experimental designs and stimuli. These large complex experiments demonstrate how flexible the system is for different tasks and goals.

This study assessed the effects of spaced practice on the ability to identify musical intervals. A total of 187 individuals from both a psychology subject pool and MTurk completed a pretest and then practiced identifying six musical intervals, with two musical intervals each randomly assigned to narrow, medium, and wide spacing for each individual. During this practice, the musical intervals were presented at two tone levels and were played as either harmonies or melodies. Participants were randomly assigned to return for a posttest 2 min, 1 day, or 7 days later. All individuals received a posttest of the same six musical intervals from practice at the same tone levels as practice and at a transfer tone level. The posttest also contained both harmonic and melodic trials.

A second experiment which utilized several features of MoFaCTS, presented participants with retrieval practice on questions about the circulatory system. A total of 178 participants, recruited through MTurk, completed the experiment producing valid data. Participants read a text (about the circulatory system), completed a retrieval practice session, and took a posttest. They were randomly assigned to practice retrieval in one of four conditions from a 2 (question depth: factual, applied)  $\times$  2 (answer format: multiple choice, short answer) between-subjects design. Practice consisted of a total of 32 trials (eight questions repeated four times each), followed immediately by 16 posttest trials (16 questions, not repeated). Each practice trial received immediate corrective feedback. MoFaCTS was able to score the short answer responses immediately by matching user type-ins to key words specified via regular expressions. This method was flexible enough to allow us to account for common synonyms and misspellings discovered through pilot testing. After practice, a posttest assessed repetition performance and transfer to a different format, a different depth, and previously unpracticed concepts.

A third experiment involved an arguably even more complex design, which replicated and extended prior work [13], in addition to testing refutation and long-term retention. In this experiment approximately 450 MTurk users filled-in blanks for a collection of 18 fill-in-the blank sentences about statistics to produce complete data. The experiment used a 2  $\times$  3 between-subjects design with 3 levels of retention interval (either 2 min, 1 day or 3 days between 2 sessions of practice) and with 2 levels of feedback (either simple feedback of the correct fill-in or refutational feedback for a portion of the wrong answers). The within-subject design for the experiment crossed 3 levels of spacing (narrow, medium, or wide) with 3 levels of practice repetition (either 2, 4 or 8 repetitions) with 2 levels of fill-in variability during practice (same or random fill repetitions) and 2 levels of fill-in variability during posttest (same or random fill-in repetitions for each of the 9 items in each condition). Order of introduction (random or

fill-in first) for each scheduling condition was counterbalanced in addition to using two different schedule orders, either starting in order from the beginning or from the end (i.e. in reverse) of the schedule. Additional sub-sequence randomization was used to prevent exact repetitions of the spacing of conditions from cueing recall. Posttest practice order tested each of the 18 items in random order with their respective response variability condition for 3 rounds of testing.

## 5 Conclusions

MoFaCTS was created as a research tool to investigate the effect of instructional sequence manipulations. The system is released on bitbucket.org as open source software (<https://bitbucket.org/ppavlik/mofacts/overview>). As development continues we welcome collaborators in building this research accelerator of research. With this in mind, continued development will focus on not only increasing the capabilities in regard to different and more complex types of trials or problems, but also on making the process of creating a student model more streamlined so as to encourage the development of multiple options for student models to explain and control practice in the system.

**Acknowledgements.** This work is supported by the National Science Foundation Data Infrastructure Building Blocks program under Grant No. (ACI-1443068) and the University of Memphis Institute for Intelligent Systems.

## References

1. Pavlik Jr., P.I., Presson, N., Dozzi, G., Wu, S.-M., MacWhinney, B., Koedinger, K.R.: The fact (fact and concept training) system: a new tool linking cognitive science with educators. In: McNamara, D., Trafton, G. (eds.) *Proceedings of the 29th Annual Conference of the Cognitive Science Society*, pp. 1379–1384. Lawrence Erlbaum, Mahwah, NJ (2007)
2. Johnson, N.F.: The role of chunking and organization in the process of recall. *Psychol. Learn. Motiv.* **4**, 171–247 (1970)
3. Anderson, J.R., Pelletier, R.: A development system for model-tracing tutors. *Proceedings of International Conference of the Learning Sciences*, Evanston, IL, pp. 1–8 (1991)
4. Meteor Development Group: Meteor (2015). <https://www.meteor.com/>
5. Node.js Foundation: Node.js (2015). <https://nodejs.org/>
6. Hickson, I., Berjon, R., Faulkner, S., Leithead, T., Navara, E.D., O'Connor, E., Pfeiffer, S.: *Html5* (2014). <http://www.w3.org/TR/html5/>
7. @mdo, @fat: Bootstrap. (2015). <http://getbootstrap.com/>
8. Mohorovicic, S.: Implementing responsive web design for enhanced web presence. In: *2013 36th International Convention on Information and Communication Technology Electronics and Microelectronics (MIPRO)*, pp. 1206–1210 (2013)
9. Bainomugisha, E., Carreton, A.L., Cutsem, T.V., Mostinckx, S., Meuter, W.D.: A survey on reactive programming. *ACM Comput. Surv. (CSUR)* **45**, 52 (2013)
10. MongoDB Inc.: *Mongodb* (2015). <https://www.mongodb.org/>
11. Pavlik Jr., P.I., Anderson, J.R.: Using a model to compute the optimal schedule of practice. *J. Exp. Psychol.: Appl.* **14**, 101–117 (2008)

12. Koedinger, K.R., Baker, R.S., Cunningham, K., Skogsholm, A., Leber, B., Stamper, J.: A data repository for the EDM community: The PSLC datashop. In: Romero, C., Ventura, S., Pechenizkiy, M. (eds.) *Handbook of Educational Data Mining*, vol. 43. CRC Press, Boca Raton (2010)
13. Maass, J.K., Pavlik Jr., P.I., Hua, H.: How spacing and variable retrieval practice affect the learning of statistics concepts. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M. (eds.) *AIED 2015. LNCS*, vol. 9112, pp. 247–256. Springer, Heidelberg (2015)

# Coordinating Knowledge Integration with Pedagogical Agents

## Effects of Agent Gaze Gestures and Dyad Synchronization

Yugo Hayashi<sup>(✉)</sup>

College of Comprehensive Psychology, Ritsumeikan University, 2-150 Iwakura-cho,  
Ibaraki, Osaka 567-8570, Japan  
y-hayashi@acm.org

**Abstract.** This study investigates how pedagogical conversational agents can facilitate learner-learner collaborative learning during a knowledge integration task. The study focuses on (1) how knowledge integration activity can be facilitated by using multiple Pedagogical Conversational Agents (PCAs) with gaze gestures and (2) how dyad coordination influences learners' perspective-changing processes and understanding. In a controlled experiment, dyads were accompanied by multiple PCAs programmed to facilitate learning. Two eye-trackers were used to detect the learner's learning process and coordination. The results show that learners who received facilitation from the PCAs about integrating different perspectives performed better on the task, and if they received gaze gestures, they tended to focus on the relationship of different knowledge as well. Recurrence analysis of gaze patterns show that those who performed well using PCAs synchronized their gaze.

**Keywords:** Pedagogical Conversational Agents · Collaborative learning · Knowledge integration · Recurrence analysis

## 1 Introduction

One important interaction in collaborative learning is knowledge integration [7, 9]. However, for novice learners, such activities are not easy to accomplish, and there is a need to investigate what kind of interaction factors influence coordination. This study examines the effects of Pedagogical Conversational Agents (PCAs) and the interactions that are effective for facilitating learning activities.

### 1.1 Collaborative Knowledge Integration and the Use of PCAs

Research shows that learning through sharing knowledge with others can lead to conceptual changes that brings new knowledge [7]. In addition, discussions based on different perspectives can bring an understanding of the content at a higher level [9]. These studies indicate that collaborative interaction for learning tasks such as integrating different knowledge and concepts is an effective strategy

for collaborative learning. Although such integration is important for developing and understanding new knowledge, it is not easy for novice learners. Considering these points, this study investigates how Intelligent Tutoring Systems (ITS) can be used to support different knowledge integration and abstraction in dyad collaborative learning.

Recent studies on tutoring systems have shown the effective use of PCAs [1, 5]. However, there are few studies that have investigated how a PCA can facilitate the collaborative learning of human-human interaction [4] and knowledge integration activities. In [2], the authors investigated how affective feedback from the PCA facilitates interactions in an explanation task. In addition, studies have shown that a multiplicity of PCAs can facilitate social interactions [3]. These studies indicate that the key to effective PCAs is to provide seamless facilitations that will not interrupt natural learner interactions. One of the challenges of this study is to propose an effective interface design that provides such seamless facilitation, but also draws adequate attention to its advice during knowledge integration activities. This study uses the implications from previous studies such as the use of multiple agents to create a social interaction. It proposes the use of “embodied gestures” by multiple PCAs to provide effective awareness during learning to help synchronize learner perspectives and thus produce better communication. It can be predicted that if learners are biased towards their own perspective, they may not pay attention to others and coordination will become poor. If PCAs are able to produce better coordination, learning performance should improve.

## 1.2 Factors Related to Knowledge Integration

In this study, we set up a knowledge integration task where two concepts are presented on a computer screen such that one is familiar to only one learner and the other is familiar only to the other learner. Given such a task, this study focuses on the use of embodied gestures such as eye gaze, which can seamlessly navigate learner’s attention to different knowledge during a discussion with peer learners. Eye gaze, which can indicate joint attention, is known to be an effective way to affect attention [10]. Because gaze gestures are known to be effective non-verbal communication strategies, they can be effective gestures for drawing a learner’s attention to the different knowledge of others naturally and seamlessly. The present study uses multiple PCAs to facilitate social interactions and it is predicted that the use of multiple gaze gestures indicating the same direction will provide a strong attention signal.

Another challenge of this study is to understand how the performance of communication processes change when using a PCA that provides suggestions for better interaction. The study focuses on the coordination of gazes in a dyad, which is known to be an effective method for capturing how learners interact during a collaborative task [6]. Eye tracking methods are used to capture the nature of coordination. Studies in Learning Science and Computer Supported Collaborative Learning (CSCL) have implied that eye-trackers are effective for understanding the nature of collaborative activities [8]. Communication studies

such as [6] suggest that the degree of gaze recurrence between dyads (speaker-listener) is correlated with collaborative performance such as understanding and establishing common ground. The recurrence of two speakers' eye movements in collaborative learning settings has become widely used to detect the performance of learners in the fields of ITS and CSCL. This index is used to investigate our research goals.

Our main interest is to understand if the use of PCAs is effective in the knowledge integration task in a learner-learner collaborative setting. Therefore, our first goal is to investigate how knowledge integration can be facilitated by multiple PCAs that address different content using gaze gestures. Our second goal is to investigate how dyad coordination influences learners' perspective-changing processes and understanding. Hence, the degree of gaze synchronization and its influence on coordination during learning was investigated.

## 2 Method

### 2.1 Task

This study focuses on the knowledge integration process of dyads (learners) in an experimental setting. The learners consisted of 78 students in a Japanese university majoring in psychology. Participants were formed into dyads and instructed to collaboratively explain two different types of sub-concepts and integrate them to gain a higher understanding of the concepts. The concepts (a) memory processing (sub concepts: long-term and short term memory), (b) knowledge processing (sub concepts: top-down and bottom-up processing), and (c) attention processing (sub concepts: zoom-in and spot-light processing) were used randomly across conditions. To create a situation where learners had different knowledge, before the experiment, participants were required to study and be prepared to explain one of the sub-concepts. Learners sat facing each other with a computer monitor in front of them. Their screen was divided into four areas: (1) sub-concept 1, (2) PCA1, (3) sub-concept 2, and (4) PCA2. A brief description prepared by one of the learners was presented for one of the concepts. Participants were instructed to first explain their familiar sub-concepts and then try to integrate this knowledge to understand the concept from a wider point of view. Participants engaged in the activity for 10 min and received an average of 10 messages from the PCAs.

We used two versions of a PCA developed in a previous study [2,3], where its roles were to provide metacognitive suggestions and facilitate communication. The system was developed in Java and programmed for server-client networks. The rule for responses was based on [2,3], where the system responded to important keywords from the learners (e.g., long-term memory, episodic memory, or implicit memory). When the system detected these words, it provided suggestions to facilitate learner metacognitions. In the current experiment, we did not use the text-based chat from the previous study. Instead, the experimenter listened to the conversations from a different room and input some of the key



phrases into the system. The system then automatically responded using pre-defined rules. Participants were randomly assigned to three conditions (no agent condition:  $n = 26$ , agent condition:  $n = 26$ , and agent+joint condition:  $n = 26$ ). In the no agent condition, no PCAs were presented on the screen. The PCAs in the agent+joint condition provided gaze gestures (e.g., looking at the concepts) while the learners provided metacognitive suggestions.

## 2.2 Dependent Variables

To analyze learning performance, participants were asked to describe the concept in text before and after the task. The answers were scored by 1 (poor) to 5 (good) based on the coding scheme of [2,3]. Two experts then graded the results. Their correlations were 0.65 and they discussed decisions before finalizing them.

The learning process was evaluated based on how often the learners paid attention to the important areas (concepts) on their screens. A high frequency of gaze on both areas indicated an effective integration learning process. The screen was divided into five areas and the number of fixations per area were counted (area 1: concept A, area 2: PCA1, area 3: concept B, area 4: PCA2, and area 5: outside areas 1–4). The gaze plots of areas 1 and 3 were counted for each learner and the following ratio was calculated:

$$b = \frac{|n_1 - n_2|}{n_1 + n_2} \quad (1)$$

where  $n_1$  is the number of fixations on his/her own concept and  $n_2$  is the number of fixations on the other learner's concept. The more  $b$  is around zero, the more the learner looked both at his/her own area and the other area and hence performed well at knowledge integration.

To investigate our second goal of the study, we investigated the amount of synchronized joint attention. Synchronization of the eye plots on the five screen areas plus other fixations was examined using recurrence analysis (chance = 0.166). The analysis used R and was based on [6], which captures the proportion of fixations at the same location for both learners during a typical time period. The recurrence of *phi* observed between the two time series (Learners A and B) was calculated for a specific time state  $k$ . Coefficient *phi*( $k$ ) increases with the frequency of matching recurrences in the same state ( $k; k$ ) and decreases otherwise.

## 3 Results

A 2 (learner: A vs. B)  $\times$  3 (PCA condition: no agent vs. agent vs. agent+joint conditions) ANOVA was conducted on the gaze plot index  $b$ . There was no significant interaction between the two factors ( $F(2, 72) = 0.7639$ ,  $p = \text{ns}$ ,  $\eta_p^2 = .0075$ ). The results of the main effect show that there were no differences between conditions ( $F(2, 72) = 0.8272$ ,  $p = \text{ns}$ ,  $\eta_p^2 = .0053$ ) but there were differences between learners ( $F(2, 72) = 1568.7678$ ,  $p < .001$ ,

$\eta_p^2 = .9561$ ). The mean  $b$  was 0.78 and  $-0.63$  for Learners A and B, respectively. This indicates that the participants preferred to look at their familiar concept regardless of the PCA.

To investigate in more detail the effect of the PCA gaze gestures, we further investigated the relationship between  $b$  and learning performance. Figure 1 shows the relation between the understanding score and gaze plot for the three conditions. A Person’s correlation analysis was conducted for each condition. For the no agent and agent conditions, there were no correlations with performance ( $r = -0.06, p = ns; r = 0.007, p = ns$ ). In contrast, there was a significant negative correlation for the agent+joint condition ( $r = -0.425, p < .01$ ), showing that learners focusing on two concepts ( $b$  close to zero) performed better on the understanding tests. This analysis shows that, depending on the learners’ level of understanding, their learning process changed because of the PCAs and gaze gestures. The advantage of using gaze-gesture PCAs did not appear clearly. However, when learners received addressing gaze gestures from the PCAs, they looked at both concepts, and when they did, they gained a better understanding. In contrast, without such gaze gestures, this tendency did not appear and learners could have been working more individually.

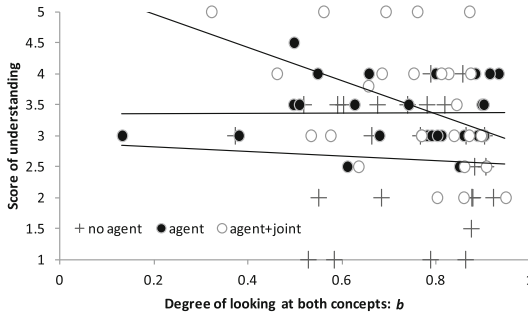


Fig. 1. Correlation of gaze plot and performance.

To analyze synchronization and learning performance, a Person’s correlation analysis was conducted for each condition. However, no differences were found (no agent condition:  $r = -0.3, p = ns$ ; agent condition:  $r = -0.29, p = ns$ ; agent+joint condition:  $r = 0.13, p = ns$ ). Next, we investigate synchronization and learning process. Differences were found in the agent and agent+joint conditions (no agent condition:  $r = -0.33, p = ns$ ; agent condition:  $r = -0.44, p < .01$ ; agent+joint condition:  $r = -0.40, p < .01$ ). This indicates that synchronized gaze was not correlated with learning performance, although it was related to the integration process.

## 4 Conclusions

There were no differences detected with respect to the use of PCAs with gaze gestures. However, learners using PCA with joint attention, when they gazed at both concepts, tended to perform better. In contrast, PCAs without such gestures did not show any relationship. This implies that gaze gestures can be used to direct learners towards important content in a learner-learner centered collaborative learning activity. From the point of providing seamless and natural facilitations to such learners, this study has successfully shown how PCA designs can produce better interaction during knowledge integration tasks. The recurrence analysis results show that learners with high recurrence perform better at learning processes. This result implies the importance of coordination during collaborative learning in a knowledge integration task. The next challenge is to investigate how coordination can be controlled by PCA navigation.

**Acknowledgments.** This work was supported by 2012 KDDI Foundation Research Grant Program, Ritsumeikan University Program for Research of Young Scientists and the Grant-in-Aid for Scientific Research (KAKENHI), No. 25870910, 16K00219.

## References

1. Graesser, A., McNamara, D.: Self-regulated learning in learning environments with pedagogical agents that interact in natural language. *Educ. Psychol.* **45**(4), 234–244 (2010)
2. Hayashi, Y.: On pedagogical effects of learner-support agents in collaborative interaction. In: Cerri, S.A., Clancey, W.J., Papadourakis, G., Panourgia, K. (eds.) *ITS 2012*. LNCS, vol. 7315, pp. 22–32. Springer, Heidelberg (2012)
3. Hayashi, Y.: Togetherness: multiple pedagogical conversational agents as companions in collaborative learning. In: Trausan-Matu, S., Boyer, K.E., Crosby, M., Panourgia, K. (eds.) *ITS 2014*. LNCS, vol. 8474, pp. 114–123. Springer, Heidelberg (2014)
4. Holmes, J.: Designing agents to support learning by explaining. *Comput. Educ.* **48**(4), 523–547 (2007)
5. Kumar, R., Rose, C.: Architecture for building conversational architecture for building conversational agents that support collaborative learning. *IEEE Trans. Learn. Technol.* **4**(1), 21–34 (2011)
6. Richardson, C.D., Dale, R., Kirkham, Z.N.: The art of conversation is coordination. *Psychol. Sci.* **18**(5), 407–413 (2007)
7. Roschelle, J.: Learning by collaborating: convergent conceptual change. *J. Learn. Sci.* **2**(3), 235–276 (1992)
8. Schneider, B., Pea, R.: Toward collaboration sensing. *Int. J. Comput. Support. Collaborative Learn.* **4**(9), 5–17 (2014)
9. Schwartz, L.D.: The emergence of abstract representation in dyad problem solving. *J. Learn. Sci.* **4**, 321–354 (1995)
10. Tomasello, M., Farrar, M.J.: Joint attention and early language. *Child Dev.* **57**(6), 1454–1463 (1986)

# An Investigation of Conversational Agent Interventions Supporting Historical Reasoning in Primary Education

Stergios Tegos<sup>(✉)</sup>, Stavros Demetriadis, and Thrasyvoulos Tsiatsos

School of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece  
{stegos, sdemetri, tsiatsos}@csd.auth.gr

**Abstract.** This work examines the efficiency of an agent intervention mode, aiming to stimulate productive conversational interactions and encourage students to explicate their historical reasoning about important domain concepts. The findings of a pilot study, conducted in the context of primary school class in Modern History, (a) suggest a favorable student opinion of the conversational agent, (b) indicate that agent interventions can help students to engage in a transactive form of dialogue, where peers build on each other's reasoning, and (c) reveal a series of interaction patterns emerging from the display of the agent interventions.

**Keywords:** Conversational agent · Academically productive talk · Computer-supported collaborative learning · Primary education · History education

## 1 Introduction

Although the value of peer dialogue interactions have been repeatedly emphasized in the field of computer-supported collaborative learning (CSCL), it is known that simply placing students in groups and asking them to interact with each other does not guarantee a pedagogically beneficial outcome [1]. According to the classroom discourse framework of academically productive talk (APT), a students' discourse should be: (a) accountable to the learning community, i.e. students should listen to and learn from each other, (b) accountable to accurate knowledge, i.e. students should support the validity of their contributions using explicit evidence, and (c) accountable to rigorous thinking, i.e. students should focus on logically connecting their claims in a reasonable manner [2]. In this perspective, APT highlights a set of valuable discussion practices and conversational moves (Table 1), which aim to induce appropriate forms of students' discourse [1]. Even though such moves are associated with improved learning outcomes and academic achievements in various contexts, their effectiveness appears to depend on factors such as the teacher's authority or the student's educational level [2].

In the past few years, researchers explored the use of conversational agents utilizing APT interventions to scaffold learners' discussions [3]. Despite the limited number of studies conducted in this potentially promising research area, it was shown that APT

**Table 1.** A selection of APT moves.

Intervention Mode	Example	Accountability
1. Agree-Disagree	“Do you agree or disagree with what your partner said about ...? Why?”	Learning community
2. Add-On	“Would you like to add something to ...?”	Learning community
3. Re-voice	“So you are saying that ... Is that right?”	Learning community
4. Build-on-Prior-Knowledge	“How does this connect with what we have discussed in class about ...”	Accurate knowledge
5. Press-for-Reasoning	“What makes you think that?”	Rigorous thinking

agents can increase learning gains and intensify group knowledge exchange [4, 5]. Yet, the findings seem to vary according to the type of the agent intervention employed and the study setting. For instance, the Agree-Disagree intervention mode (Table 1, item 1) was found to perform well in higher education settings [4], while the less demanding mode of Re-voicing (Table 1, item 3), which elicits self-oriented conversational moves, was shown to be only effective for younger novice learners [6].

Drawing on the above line of research, this work presents a pilot study exploring the impact of a Press-for-Reasoning (PR) agent intervention mode (Table 1, item 5) in the context of a collaborative learning activity. Our aim is to collect preliminary evidence regarding the effectiveness of the PR intervention mode in primary education and identify key conversational interaction patterns or behaviors, which could facilitate the future development and configuration of well-targeted agent interventions.

## 2 Method

### 2.1 Participants

A one-group exploratory study was conducted in the context of a 5th-grade primary school class in Modern History. The activity took place in a computer lab. The participants were 32 students (18 females) and their ages ranged from 11 to 12 years old.

### 2.2 An APT Conversational Agent for History Education

For the purpose of this study, the MentorChat conversational agent system [5] was integrated in an online educational game, called HistoryLand. The game was developed as a complementary tool for teaching Modern History in primary education settings.

In the first game phase, students are shown a series of cards relating to the historical period of the current level, namely World War I, World War II or the Balkan Wars. Then, they are asked to individually collect the card item that is incongruous with the prescribed historical period. In the next game phase, students are assigned to dyads and enter a chat (Fig. 1). They are expected to realize that the incongruous card items they

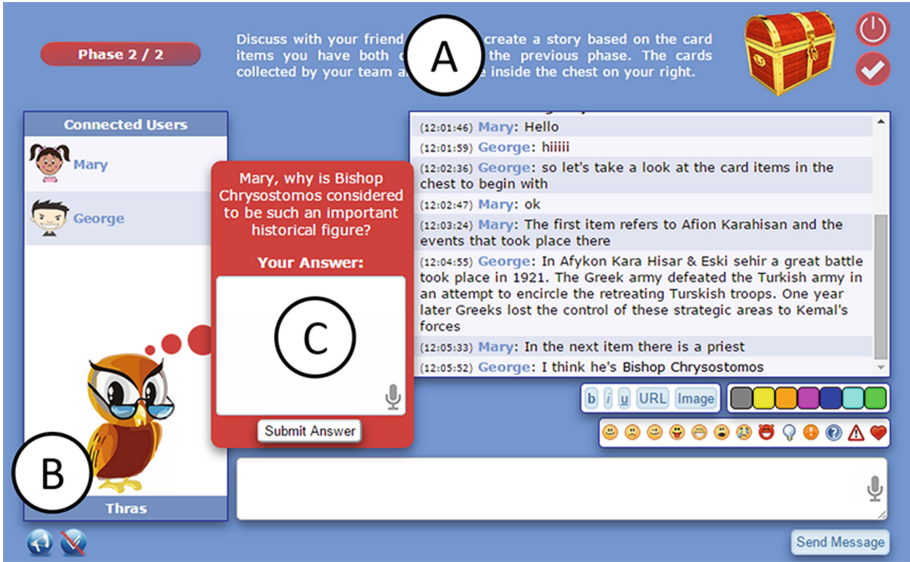


Fig. 1. A screenshot of the chat interface.

have collected refer to the Asia Minor Expedition and asked to create a joint story about it (Fig. 1A). This collaborative phase features an animated conversational agent illustrated as an owl (Fig. 1B), which delivers PR interventions displayed outside the main chat window (Fig. 1C).

The aim of the PR agent interventions is not to introduce additional content but encourage students to externalize their reasoning. Reasoning in the context of history education can be regarded as the process in which a student organizes information about the past to build or explain an interpretative historical case [7]. Simply posting a claim is often not enough without providing the historical reasoning that has led to it. Indeed, it is when students explicate their reasoning in writing that they make it available for others (or oneself) to assess, question or challenge [1]. In this manner, the agent attempts to support accountability to rigorous thinking posing a variety of questions that elicit students’ reasoning about the important historical concepts being discussed (Table 2, row 3). These concepts, which form the agent domain model, are derived from the first game phase where students individually collect some card items.

Table 2. A dialogue excerpt demonstrating an agent intervention.

Time	Name	Message
11:15	Eva	I think we should include what happened in Smyrna in 1922.
11:44	Jim	I do not think this should be our main focus here
11:45	(Agent)	<i>Jim, what do you think caused the destruction of Smyrna in 1922?</i>
12:29	Jim	Oh! Hmmm... Eva do you remember what happened?

(Continued)

**Table 2.** (Continued)

Time	Name	Message
12:51	Eva	I think I remember something
13:19	Jim	Tell me please
14:17	Eva	I think the Greek army fought with the Turkish army there. Turkish forces entered the city in September 1922 and there was fire all over the city.
14:39	Jim	Right! Thanks;-)
16:17	Jim	(Submitted Answer) Thras, Smyrna was destroyed because of the war between the Greeks and Turkish. The war ended in September 1922 when the Turkish troops gained control of Smyrna. At that time, while the city was burning, many Greeks died at the city port while trying to escape in panic
16:47	Jim	I remember that the commander of the Turkish army was Mustafa Kemal
16:48	Eva	That is right. Unfortunately, many people lost their lives then and the whole city was ruined because of the fires and battles...

In a nutshell, the conversational agent (a) analyzes students' contributions to identify relevant key domain concepts or phrases, (b) compares the current students' knowledge representation with the agent knowledge representation using segmentation, stemming and pattern matching algorithms to determine whether an APT intervention is suited, and (c) synthesizes and delivers an agent intervention. As illustrated in Table 2, when a student (Eva) introduces a key domain concept in the group discussion, the agent waits for a maximum of 30 s and, unless some form of reasoning on that concept is detected, the agent displays its intervention addressing their partner (Jim). The aim of this mechanism is to promote reasoned participation by all students.

### 2.3 Procedure

Students were asked to play HistoryLand during their Modern History class. Throughout the first three levels of the game, each student collected card items that displayed historical figures or events. Thereafter, students collaborated in dyads (16) using their card items to compose a joint story about the Asia Minor Expedition. The whole activity lasted about 90 min whereas the collaborative phase, which constitutes the primary focus of this study, lasted approximately 25 min. Following the activity, students were requested to fill in a student opinion questionnaire.

## 3 Data Analysis and Results

The post-task questionnaire measured students' perceptions of the conversational agent using a 5-point Likert scale (1-disagree to 5-agree). A high internal consistency coefficient of reliability was reported ( $C_\alpha = .88$ ). Table 3 presents the questionnaire results.

**Table 3.** Student opinion questionnaire results relating to the conversational agent

Questionnaire Items	M (SD)
The presence of the agent made my team’s discussion more engaging.	4.45 (1.06)
The agent interventions during the discussion were comprehensible.	4.56 (0.88)
Interventions helped me to recall valuable information on topics being discussed.	4.53 (0.94)
I understood the ‘Asia Minor Expedition’ better through responding to the agent.	4.32 (1.23)

A discourse analysis was conducted focusing on dialogue transactivity, which is regarded as a valid indicator of the learning taking place in a conversation [8]. A dialogue can be considered as transactive if learning partners build on each other’s reasoning as the discussion progresses. Students’ contributions were classified by the authors using the discourse analysis scheme (Table 4) presented in Sionti et al.’s [8] study.

**Table 4.** The discourse analysis scheme.

Category	Description
Off-task	Playing a purely social function or not relating to the task (e.g. “Hello”)
Management	Used for task coordination (e.g. “Submit your answer.”)
Assertion	Not displaying any form of reasoning (e.g. “The Eskisehir battle was important.”)
Repetition	Reiterations of previous statements
Non-transactive	Externalizations of reasoning that do not connect with any previously expressed reasoning (e.g. “Asia Minor Expedition ended because of the...”)
Transactive	Externalizations of reasoning that connect with some previous display of reasoning (e.g. “I agree but this is not entirely true since...”)

The discourse analysis revealed 56 agent interventions and 514 distinct students’ contributions (Table 5).

**Table 5.** Discourse analysis results (N = 16 groups).

Contribution Type	Total	Freq.	Mean	St. Dev.
Off-task	56	10.89 %	3.50	2.31
Management	101	19.65 %	6.31	3.96
Repetition	16	3.11 %	1.00	1.15
Assertion	179	34.82 %	11.19	6.12
Non-transactive	56	10.89 %	3.50	1.10
Transactive	106	20.62 %	6.63	2.96



In order to explore the agent impact on students' discussions we identified all agent-induced contributions. These could be either direct responses to the agent or follow-up comments stimulated by the agent intervention. Despite the rather low number of the PR interventions displayed, our analysis revealed that, on average, each agent intervention induced more than one transactive contributions (Table 6, row 3).

**Table 6.** Intervention mode impact on transactivity.

Measurements	Total	%
Agent-Induced Non-transactive Contributions	19	33.93
Agent-Induced Transactive Contributions	67	63.21
Transactivity Induction Ratio (Agent-Induced Transactive Contributions/Agent Interventions)	$67/56 = 1.20$	–

Given that the most important aspects of conversational interactions are based on the temporal sequentiality of students' statements, we also adopted a sequential analytic approach [9]. After examining the interaction flow of each group chat, we performed a line-by-line analysis, which resulted in the identification of the following themes:

*Partner's Impatience.* Students seemed to be impatient while awaiting their partners to submit their agent response and often encouraged them to hurry up. This may be due to the fact that the students' usually spent additional time preparing their answers to the agent as compared to the time they spent responding to their peer.

*Attention Capture.* In contrast to our previous studies involving university students, the younger students never ignored the agent interventions. Even when students could not recall any information about the historical event or figure asked, they replied to the agent using phrases such as "I am sorry but I cannot remember this".

*Amplified Agent Authority.* Although the conversational agent was not designed to have an authoritative communication style or role, the young learners perceived the agent as an authority figure. Students' responses to the agent were far more formal and polite as compared to their responses to their partners' questions.

*Dynamic Configuration of Collaboration Practices.* Students' conversational behavior seemed to change as the discussion progressed and more agent interventions appeared. As the time passed, peers engaged more frequently in question-answering dialogue turns, as if they were trying to mimic the agent role by posing similar questions. Yet, further research is clearly needed to draw any definite inferences.

*Reciprocal Peer Support.* When students could not respond to the agent questions, they asked for the assistance of their partner, who often was more than willing to help.

## 4 Conclusion

We consider the study results to be encouraging suggesting a favorable students' opinion of the conversational agent and a positive effect of the PR intervention mode on the transactive features of students' dialogue. Despite the study limitations, such as

its one group design and small sample size, we expect its findings to provide valuable insights regarding the design, feasibility and potential impact of APT agent intervention modes.

**Acknowledgements.** The authors are grateful to Dimitris Gkoumas, Maria Kioumousidou, Dimitra Kioutsouki, and Maria Vavami for their contributions and the development of HistoryLand.

## References

1. Michaels, S., O'Connor, M.C., Hall, M.W., Resnick, L.B.: *Accountable Talk Sourcebook: For Classroom That Works*. Institute for Learning, University of Pittsburgh, Pittsburgh (2010)
2. Michaels, S., O'Connor, C., Resnick, L.B.: Deliberative discourse idealized and realized: accountable talk in the classroom and in civic life. *Stud. Philos. Educ.* **27**, 283–297 (2008)
3. Stahl, G.: Computer-supported academically productive discourse. In: Resnick, L., Asterhan, C., Clarke, S. (eds.) *Socializing Intelligence Through Academic Talk and Dialogue*, pp. 213–224. AERA Publications, Washington, DC (2015)
4. Adamson, D., Ashe, C., Jang, H., Yaron, D., Rosé, C.P.: Intensification of group knowledge exchange with academically productive talk agents. In: *Proceedings of the 10th International Conference on Computer Supported Collaborative Learning*, pp. 10–17. ISLS (2013)
5. Tegos, S., Demetriadis, S., Tsiatsos, T.: Promoting academically productive talk with conversational agent interventions in collaborative learning settings. *Comput. Educ.* **87**, 309–325 (2015)
6. Dyke, G., Adamson, D., Howley, I., Rose, C.P.: Enhancing scientific reasoning and explanation skills with conversational agents. *IEEE Trans. Learn. Technol.* **6**(3), 240–247 (2013)
7. Van Drie, J., Van Boxtel, C.: Historical reasoning: towards a framework for analyzing students' reasoning about the past. *Educ. Psychol. Rev.* **20**, 87–110 (2008)
8. Sionti, M., Ai, H., Rosé, C.P., Resnick, L.: A framework for analyzing development of argumentation through classroom discussions. In: Pinkwart, N., McClaren, B. (eds.) *Educational Technologies for Teaching Argumentation Skills*. Bentham Science (2010)
9. Stahl, G.: Interaction analysis of a biology chat. In: Suthers, D., Lund, K., Rosé, C.P., Law, N. (eds.) *Productive Multivocality in the Analysis of Group Interactions*, pp. 511–539. Springer, New York (2013)

# Impact of Question Difficulty on Engagement and Learning

Jan Papoušek, Vít Stanislav<sup>(✉)</sup>, and Radek Pelánek

Masaryk University, Brno, Czech Republic  
slaweet@mail.muni.cz

**Abstract.** We study the impact of question difficulty on learners' engagement and learning using an experiment with an open online educational system for adaptive practice of geography. The experiment shows that easy questions are better for short term engagement, whereas difficult questions are better for long term engagement and learning. These results stress the necessity of careful formalization of goals and optimization criteria of open online education systems. We also present disaggregation of overall results into specific contexts of practice.

## 1 Introduction

Making practice suitably challenging is one of the key goals of adaptive educational systems. The general idea that the best activity is neither too easy nor too difficult was formulated as Inverted-U Hypothesis [1]. Lomas et al. [6] found that in the context of their simple educational game easier problems lead to higher engagement, but lower learning. A similar research was done using Math Garden software [2]. The authors compared three conditions and showed that the easiest condition led to the best learning (mediated by a number of solved tasks). Other authors have used more complex experimental techniques to find optimal parameter values (e.g., Bayesian optimization), but they have optimized only with respect to short term engagement [3] or short term transfer [4].

We report results of an online experiment evaluating impact of question difficulty on learning and engagement in the context of declarative knowledge and an open educational system. Specifically, we use a system for an adaptive practice of geographical facts [9] (e.g., names and location of countries or cities); the system is publicly available at <http://outlinemaps.org>. We have reported experiments with question difficulty in this system in previous work [8], but only with respect to engagement. Here we provide more detailed analysis including also learning. The used methodology is similar to a previous work [10] which compared an adaptive and a random construction of questions within the system. Here, we pay more attention to issues related to data aggregation and a conflict between short and long term engagement.

Analyzing data from the experiment containing conditions targeting 5%, 20%, 35%, and 50% error rate, we observe a conflict between learning and long term engagement on one side (more difficult is better), and short term

engagement on the other (easier is better). These results demonstrate the risk hidden in optimizing only short term behaviour of the system (as done in [3, 4]). Our results are also in contrast with previous studies [2, 6], which concluded that easier questions are better (we are, however, using educational system from a completely different domain).

## 2 Experimental Setting

We have performed the evaluation using a randomized trial with four experimental conditions within a widely used adaptive system providing practice of geography. The system estimates learners' knowledge and based on this estimate it adaptively constructs multiple-choice (2–6 options) or open questions of suitable difficulty [9]. The adaptive behaviour of the system is based on models of learners' knowledge. These models provide a prediction of the current knowledge for each learner and item. This part has been described and evaluated in previous work [9], here we use these models as a 'black box'.

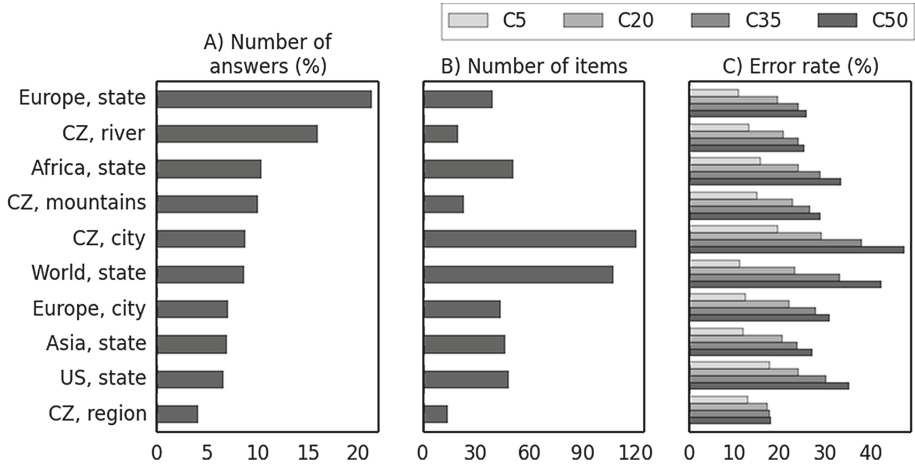
The system uses a target error rate and adaptively constructs questions in such a way that learners' achieved performance is close to this target [8]. In our experiment we evaluate four experimental conditions which differ only in one aspect – the target error rate: 5 %, 20 %, 35 %, 50 %. In the following text we denote the conditions as C5, C20, C35, and C50. Learners were assigned to one of the conditions randomly when they entered the system for the first time. The experiment was performed from November 2015 to January 2016 and we have collected almost 3 300 000 answers from roughly 37 000 learners. To make our research reproducible we make the analyzed data set available<sup>1</sup> (together with a brief description and terms of use).

To evaluate learning within the adaptive system we use "reference questions". The reference questions are open questions about a randomly chosen item from a particular context (independently of the experimental condition). The questions are used periodically (every 10th question is a reference question). The first reference question is the first question within a context, i.e. before the adaptive algorithm has any chance to influence the practice for the given context. A similar approach based on random items has been used for evaluation previously, for example in [4, 10].

An important factor that influences the evaluation and interpretation of results are different contexts within the system. Learners can choose different maps and types of places to practice. These contexts differ widely in their difficulty (prior knowledge) and the number of items available to practice (10–170). Distribution of answers is highly uneven, most learners practice a few popular maps. For the analysis we use 10 contexts with most answers (listed in Fig. 1). More detailed analysis of differences among contexts is available in the full version of the paper [11].

---

<sup>1</sup> <http://www.fi.muni.cz/adaptivlearning/data/slepemapy/2016-ab-target-difficulty.zip>.



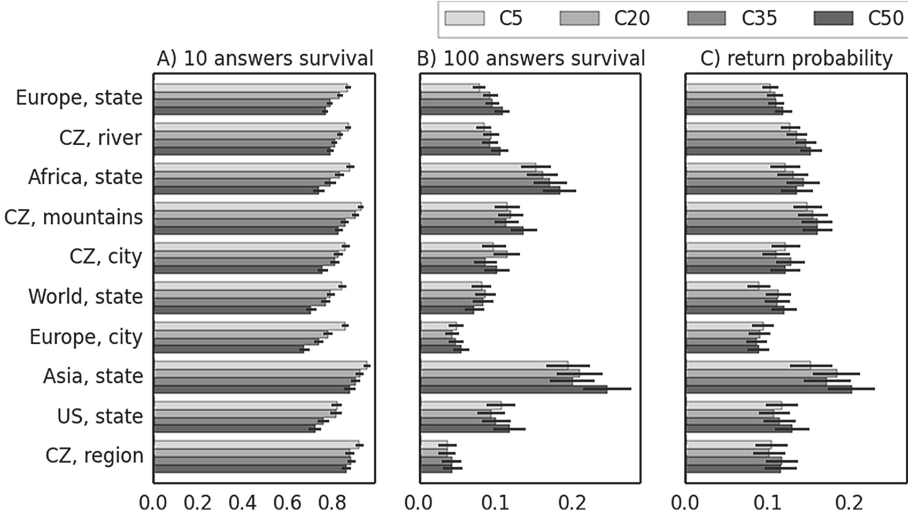
**Fig. 1.** Top 10 mostly used contexts available for learners to practice. (A) percentage of answers in the analyzed data set, (B) number of items, (C) average error rate per experimental condition ignoring reference answers.

### 3 Engagement

To evaluate engagement we consider (1) survival rates (i.e., proportion of learners who answer at least  $k$  questions), and (2) probability of returning to the system (after a delay of more than 10 h; the specific duration of the delay is not important for presented results). While analyzing differences among the conditions, we have identified opposite tendencies with respect to short term and long term engagement. The main trend is that while conditions with easier questions enhance engagement at the beginning, more difficult conditions engage more learners later on.

From the global viewpoint, short term engagement is better in case of easier questions. The survival rate after 10 answers is sorted according to question difficulty (C5: 89.2%, C20: 87.0%, C35: 84.0%, C50: 81.2%, confidence interval  $\pm 0.77\%$ ). The differences are decreasing with the number of answers, survival rates after 100 answers are very similar in all conditions (from 26.0% to 26.5%, confidence interval  $\pm 0.88\%$ ). Note that after 30 or more questions, the conditions C35 and C50 no longer achieve their target error rate in most contexts, since the items from these contexts are already mastered by learners. The return rate increases with the difficulty of questions, the largest difference being between C5 and other conditions (C5: 15.2%, C20: 16.0%, C35: 16.6%, C50: 16.8%, confidence interval  $\pm 0.75\%$ ).

There are quite large differences among individual contexts (see Fig. 2), most likely caused by learners' preferences and implementation details of the system, e.g., the system recommends 6 contexts (e.g., European states) as "quick start" options on the home page, which makes their survival rates lower than survival rates of "self-selected" contexts (e.g., Asian states). The magnitude of differences



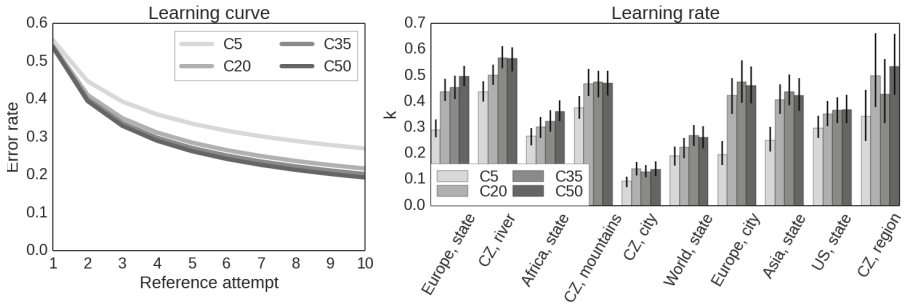
**Fig. 2.** Survival analysis (A, B) and probability of return after 10 h (C) for 10 most practiced contexts and 4 experiment conditions. Error bars represent 95% confidence intervals.

between conditions is mostly aligned with differences in their behaviour in the particular context, e.g. its difficulty or number of items available to practice.

Short term survival (Fig. 2A) differs in all contexts in favour of easier conditions. In case of long term survival (Fig. 2B), the trend is quite opposite, although for individual contexts the differences are typically rather small. This contrast is best seen on European states (the context with most data), where we see a reliable difference between C50 and C5.

## 4 Learning

The evaluation of learning cannot be simply based on the achieved error rate of learners, since this error rate is by definition heavily influenced by the used experimental conditions. For this reason we collect previously described reference answers, which are not affected by any condition, and from these reference answers we construct learning curves. We construct a learning curve [7] in the same way as in [10]. We put together reference answers from all available contexts and compute an average error rate preserving their ordering within contexts (e.g., we put together all the first reference answers from all users and contexts to get the first point of the learning curve). We do not filter any data and users may quit their practice on their own, so for the first point of the learning curve we have more answers than for the second one and so on – the results thus may be influenced by attrition bias, this issue is discussed in the full version of the paper [11]. In accordance with previous research [7, 10] we assume that the learning curve corresponds to the power law, i.e., the error rate can be expressed



**Fig. 3.** Left: Global learning curve based on the power law  $ax^{-k}$ . Right: Learning rate  $k$  for different contexts. Error bars stand for 95% confidence intervals computed using bootstrapping.

as  $ax^{-k}$ , where  $x$  is the number of attempts,  $a$  is the initial error rate, and  $k$  is the learning rate.

When we mix data from all contexts together and analyze learning only on the global level, more difficult practice seems to lead to better learning, see Fig. 3 (left). Figure 3 (right) shows more detailed analysis for individual contexts. Instead of looking at the whole learning curves, we assume that the initial error rate  $a$  is the same for all conditions within the same context and we compare only their learning rate (the parameter  $k$  in the power law). The learning rate differs among some contexts (e.g., Czech cities vs. European states) due to differences in the number of items and other factors. Here, we are mainly interested in the comparison of our experimental conditions within individual contexts. The general trend is the same as in the case of the global learning curve with the largest differences being between C5 and other conditions. The size of differences is related to different behaviour of conditions within individual contexts – number of items available to practice and actually achieved error rate (e.g., European countries are much easier than Czech cities for most of our users).

## 5 Discussion

We performed an experiment with varied difficulty of items in a widely used open online educational system. The most interesting result is the difference between “short term engagement” (not leaving immediately) and “long term engagement” (prolonged usage of the system). Easy questions lead to better short term engagement, whereas difficult questions are better for the long term engagement. We also evaluated learning improvement, which is better for more difficult questions (the main difference being between very simple questions and others). These results are in contrast with previous research [2, 6], which may be due to different learning domain (procedural knowledge in mathematics vs. declarative knowledge in geography). The issue of optimal difficulty thus warrants more attention in research.

These results have specific consequences for the studied system and for closely similar systems (e.g., vocabulary learning) – it seems that the system should start with easy questions “to hook learners up” and then switch to more difficult questions. But more importantly, the results have important methodological consequences for evaluation and optimization of educational systems. It is tempting to use “short term engagement” as a proxy for system quality, because this metric can be easily and quickly measured (as opposed to learning or long term engagement); this has been done for example in [3, 8]. Our results show that this approach can be misleading and that it is important to use a “multi-criteria approach” (using techniques like [5]) since both engagement and learning are important in open online educational systems.

## References

1. Abuhamdeh, S., Csikszentmihalyi, M.: The importance of challenge for the enjoyment of intrinsically motivated, goal-directed activities. *Pers. Soc. Psychol. Bull.* **38**(3), 317–330 (2012)
2. Jansen, B.R.J., Louwerse, J., Straatemeier, M., Van der Ven, S.H.G., Klinkenberg, S., Van der Maas, H.L.J.: The influence of experiencing success in math on math anxiety, perceived math competence, and math performance. *Learn. Individ. Differ.* **24**, 190–197 (2013)
3. Khajah, M.M., Roads, B.D., Lindsey, R.V., Liu, Y.-E., Mozer, M.C.: Designing engaging games using bayesian optimization. In: *Computer-Human, Interaction* (2016)
4. Liu, Y.-E., Mandel, T., Brunskill, E., Popović, Z.: Towards automatic experimentation of educational knowledge. In: *Human Factors in Computing Systems*, pp. 3349–3358. ACM (2014)
5. Liu, Y.-E., Mandel, T., Brunskill, E., Popovic, Z.: Trading off scientific knowledge and user learning with multi-armed bandits. In: *Educational Data Mining*, pp. 161–168 (2014)
6. Lomas, D., Patel, K., Forlizzi, J.L., Koedinger, K.R.: Optimizing challenge in an educational game using large-scale design experiments. In: *SIGCHI Conference on Human Factors in Computing Systems*, pp. 89–98. ACM (2013)
7. Martin, B., Mitrovic, A., Kenneth, K.R., Mathan, S.: Evaluating and improving adaptive educational systems with learning curves. *User Model. User Adap. Inter.* **21**(3), 249–283 (2011)
8. Papoušek, J., Pelánek, R.: Impact of adaptive educational system behaviour on student motivation. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M.F. (eds.) *AIED 2015. LNCS*, vol. 9112, pp. 348–357. Springer, Heidelberg (2015)
9. Papoušek, J., Pelánek, R., Stanislav, V.: Adaptive practice of facts in domains with varied prior knowledge. In: *Educational Data Mining*, pp. 6–13 (2014)
10. Papoušek, J., Stanislav, V., Pelánek, R.: Evaluation of an adaptive practice system for learning geography facts. In: *Learning Analytics and Knowledge* (2016, to appear)
11. Papoušek, J., Stanislav, V., Pelánek, R.: Evaluation of the impact of question difficulty on engagement and learning. Technical Report FIMU-RS-2016-02, Masaryk University (2016)



# Are There Benefits of Using Multiple Pedagogical Agents to Support and Foster Self-Regulated Learning in an Intelligent Tutoring System?

Seth A. Martin<sup>(✉)</sup>, Roger Azevedo, Michelle Taub,  
Nicholas V. Mudrick, Garrett C. Millar, and Joseph F. Grafsgaard

NCSU, Raleigh, NC, USA  
{samarti7, razeved, mtaub, nvmudric,  
gcmillar, jfgrafsg}@ncsu.edu

**Abstract.** This study examined the proportional learning gains attained by 165 college students as they learned about the human circulatory system over two sessions with the intelligent tutoring system, MetaTutor. Results indicated that learners in the prompt and feedback condition, which were afforded the full capabilities of the four pedagogical agents (PAs), attained significantly greater proportional learning gains than learners in the control condition who did not receive the same scaffolding. In addition, we also found that the amount of time spent with each PA produced different types of impacts on the learners, with Sam the Strategizer having the most influence on proportional learning gains. Lastly, results from the revised Agent Persona Inventory (API), administered following the learning session with MetaTutor, revealed key findings regarding learners' overall retrospective affective reactions towards each individual PA. These results have implications for the design of future PAs capable of offering real-time and adaptive pedagogical instruction within Intelligent Tutoring Systems (ITSs).

**Keywords:** Pedagogical agents · Adaptive hypermedia systems · Intelligent tutoring systems · Self-regulated learning · Metacognition · Scaffolding and tutoring

## 1 The Impact of Pedagogical Agents on Learning with MetaTutor

Intelligent tutoring systems (ITSs) are capable of improving the quality, experience, and instructional value of educational environments by supporting a student's learning through the use anthropomorphic virtual characters called pedagogical agents (PAs) [1–3]. One key component of ITSs is that they have been shown to play an important role in facilitating students' self-regulated learning (SRL) and in particular, have been used to scaffold, foster, and support cognitive, affective, metacognitive, and motivational (CAMP) processes [4]. Several existing ITSs utilize PAs to support and foster specific CAMP processes. For example, AutoTutor incorporates conversational

agents to detect and regulate affective processes during computerized tutoring [3]. Betty's Brain uses the learning by teaching paradigm to facilitate metacognitive monitoring during knowledge construction in science-related topics [2]. Other ITSs, like Crystal Island, use virtual agents to foster learner engagement and support affective processes predictive of scientific reasoning [5]. MetaTutor is an intelligent multiagent hypermedia-learning environment that uses four PAs to foster conceptual understanding of the human circulatory system by highlighting the importance of SRL [4].

Utilizing a four-agent system raises several questions pertaining to their ability to cumulatively facilitate SRL effectively. For example, are the unique roles of the PAs effective in enhancing complex learning? If so, is it because the amount of time they are interacting with the learners, or is it perhaps related to the frequency in which they interact with them? Despite the potential benefits of providing timely and adaptive feedback, we also know from our research, and the research of others, that these PAs may also impact learners' reactions toward them following the learning session. These are just a few of the critical questions that are addressed in this study.

## 2 Method

165 college students (52.7 % female) from three North American universities participated in a 2-day laboratory study. The sample ranged in age from 18 to 41 ( $M = 20.4$ ,  $SD = 3.02$ ). Learners were randomly assigned to either the prompt and feedback condition ( $n = 82$ ) or the control condition ( $n = 83$ ). Learners received \$10 per hour, and up to \$40 for completing the study.

MetaTutor is an intelligent multiagent hypermedia-learning environment that helps guide learners through 47 pages of challenging content by setting subgoals and advancing the learners toward those goals [4] (see Fig. 1). While interacting with MetaTutor, learners are guided by four PAs that provide timely scaffolding. Each agent, aside from Gavin the Guide, offers support on one specific component of SRL. Gavin's objective is to provide learners the information necessary to navigate the environment. Mary the Monitor supports learners by helping them monitor what has taken place during the session. Mary recommends the use of metacognitive processes such as content evaluations (CE), feelings of knowing (FOK), judgments of learning (JOL), and monitoring progress toward goals (MPTG). Pam the Planner supports learners by emphasizing planning, activating prior knowledge, and creating relevant subgoals. Additionally, Sam the Strategizer encourages effective cognitive strategy use (i.e., coordinating informational sources, making inferences, summarizing, etc.) as learners progress toward completing their goals.

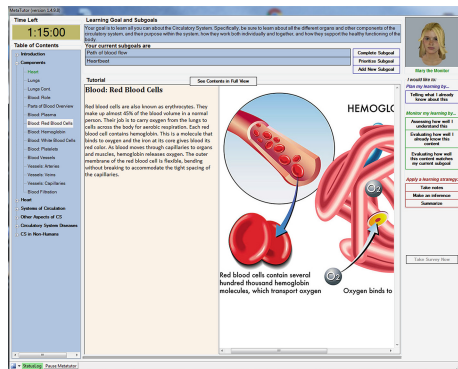


Fig. 1. MetaTutor interface

Learners were also asked to complete pre and posttests, a demographic questionnaire, and several self-report measures, including the revised Agent Persona Inventory (API). The pre and posttest measures were 30-item multiple-choice tests on the human circulatory system. A proportional learning gain score was calculated for each learner based on the results of the pre and posttests.<sup>1</sup> The revised API is a 60-item, four-section questionnaire, adapted from Baylor and Ryu's 2003 Agent Persona Inventory, which was used to assess how learners felt about their interactions with the PAs. Proportion of the session spent interacting with a PA was calculated by summing the total time spent interacting with each PA and dividing by the learner's total session duration. Similarly, summing the number of interactions with a PA and dividing it by the duration calculated the frequency of interacting with PAs.

### 3 Results

#### 3.1 Research Question 1: Does Experimental Condition Impact Proportional Learning Gains During Learning with MetaTutor?

An Independent samples *t*-test was conducted to investigate the impact of experimental condition, control ( $M = 21.74$ ,  $SD = 24.21$ ), and prompt and feedback ( $M = 30.92$ ,  $SD = 20.39$ ), on proportional learning gains. On average, learners in the prompt and feedback condition achieved greater proportional learning gains than learners in the control condition  $t(163) = 2.63$ ,  $p = .01$ ,  $d = .41$ . The results produced a medium effect.

#### 3.2 Research Question 2: Does Proportion of PAs' Time Spent Interacting with Learners Predict Proportional Learning Gains?

Based on the result of research question 1, we were interested in investigating if some PAs had a greater impact on proportional learning gains than others. Gavin was excluded from analyses as his role was helping learners navigate the environment, and therefore, did not provide prompts, scaffolding or feedback. A multiple linear regression was conducted to predict proportional learning gains based on the proportion of session time spent interacting with Sam, Mary, and Pam. The results indicated that the proportion of the session spent interacting with Sam was predictive of learning gains ( $\beta = .29$ ,  $p = .003$ ), whereby, the longer the learner interacted with Sam, the greater their proportional learning gains. However, the proportion of time spent interacting with Mary ( $\beta = .02$ ,  $p = .86$ ) and Pam ( $\beta = .06$ ,  $p = .51$ ), was not predictive of proportional learning gains. The proportion of session time spent interacting with the PAs was predictive of 9 % of the variability in proportional learning gains made during learning with MetaTutor ( $R^2 = .09$ ,  $p = .002$ ). However, the finding that only the proportion of the session spent interacting with Sam is predictive of proportional

---

<sup>1</sup> We used the proportional learning gain formula commonly used by ITS researchers (e.g., Azevedo, D'Mello, Graesser, Graffsgaard).

learning gain raises further questions—e.g., does the frequency of PA-learner interactions and the qualitative nature of overall interactions predict outcomes?

### 3.3 Research Question 3: Does the Frequency of Interactions Between the PAs and Learners Predict Proportional Learning Gains?

Based on the findings of research question 2, another multiple linear regression was conducted to predict proportional learning gains based on the frequency of PA scaffolding and intervention throughout the session. Results indicated that the frequency of interactions for Sam ( $\beta = .22, p = .06$ ), Mary ( $\beta = .002, p = .99$ ), and Pam ( $\beta = -.02, p = .83$ ) were not predictive of proportional learning gains. Results show that, in general, the frequency of interaction (i.e., instances of prompting and scaffolding) with PAs does not predict proportional learning gains. These findings suggest that the qualitative nature of PA-learner interactions may be more predictive than the quantity.

### 3.4 Research Question 4: What are Learners' Posthoc Appraisals of the PAs and are they Related to Learning?

Pearson correlations were conducted to investigate the relationship between learners' affect, proportion of session time spent interacting with the PAs, and feelings about the value of MetaTutor. The results found non-significant correlations between affect and proportional learning gains, where affect was self-reported on the revised API. However, positive and negative feelings of affect toward the PAs were found to be significantly correlated with the proportion of time spent interacting with the PAs.

Results showed general negative affect towards **Sam** with significant positive correlations between the proportion of session time spent interacting with Sam and frustration ( $r(165) = .57, p < .001$ ), anxious ( $r(165) = .38, p < .001$ ), ashamed ( $r(165) = .16, p = .04$ ), hopeless ( $r(165) = .21, p = .01$ ), contempt ( $r(165) = .27, p = .001$ ), and confusion ( $r(165) = .25, p = .001$ ), as well as a significant negative correlation between Sam and enjoyment ( $r(165) = -.16, p = .05$ ).

As for **Mary** the monitor, results indicated significant positive correlations between proportion of session time spent interacting with Mary and pride ( $r(165) = .16, p = .05$ ), frustration ( $r(165) = .27, p = .001$ ), anxious ( $r(165) = .17, p = .03$ ), ashamed ( $r(165) = .27, p = .001$ ), hopeless ( $r(165) = .19, p = .01$ ), and contempt ( $r(165) = .18, p = .02$ ).

There were significant positive correlations between proportion of session time spent interacting with **Pam** and frustration ( $r(165) = .17, p < .03$ ), as well as boredom ( $r(165) = .21, p = .01$ ).

Interestingly, the results indicate that although **Sam** and **Mary** induced negative affect amongst learners, there was also a significant positive correlation between proportion of session time spent interacting with these two PAs and learners reporting that Sam ( $r(165) = .17, p = .03$ ) and Mary ( $r(165) = .16, p = .04$ ) helped them see the value in using MetaTutor to support and foster their self-regulated learning.

## 4 Implications for Designing Intelligent Tutoring Systems with Multiple PAs to Support and Foster SRL

The inclusion of PAs to detect, support, and foster CAMM and SRL processes during complex learning with ITSs continues to pose theoretical, methodological, and analytical issues with implications for designing intelligent PAs. Ideally, adaptive scaffolding needs to be based on a system's accurate inference of the deployment of CAMM processes during real-time learning (e.g., reading speed, revisit to the same content, etc.). However, this remains a challenge for the ITS community as well as others for various reasons (i.e., the collection and interpretation of multichannel data, accurate inferences regarding CAMM processes across multichannel data, translating these inferences into PA intelligent behavior, etc.) [4].

Analyses indicated that learners in the prompt and feedback condition attained significantly greater proportional learning gains than learners in the control condition. Primarily, the difference between these conditions is the access to scaffolding and intervention from the four PAs. As results indicated that only interacting with Sam was predictive of proportional learning gains, further testing is necessary to identify the components of the prompt and feedback condition that contribute to this outcome.

To better understand the influence that the four PAs had on proportional learning gains, it was necessary to examine their impact on an individual level. As such, the proportion of the session spent interacting with each PA, and the frequency of interaction with each PA was examined. Results indicated that as the proportion of the session spent interacting with Sam increased; proportional learning gains did as well. This may be explained by the complexity of Sam's interactions and also from a self-regulatory perspective, since getting learners to use sophisticated strategies is fundamental to complex learning. More specifically, our results indicate that learners' compliance with Sam's prompting of and scaffolding for complex cognitive strategies (i.e., summarize, make an inference, coordinate informational sources, take notes, and re-read) is associated with increases in learning about complex science content. As previously illustrated, the time spent with the other three PAs was not predictive of proportional learning gains. Proportion of the session interacting with Gavin was not expected to be predictive as he simply guides learners through the environment. As for Pam, planning to a certain extent is quite abstract and because of her role, she has fewer and shorter interactions than Mary and Sam. The proportion of time spent interacting with Mary the monitor was also found to not be predictive of proportional learning gains. This may be explained through the literature on metacognition, where it is often found that the quality of metacognitive processes (e.g., making accurate metacognitive judgments) deployed is more impactful than the sheer volume of use [6]. To account for these results, we intend to reconsider from a design perspective, some of the processes, roles, timing, and most importantly, the quality of the scaffolding offered by the other PAs. Highlighting this issue even more, the frequency of interaction with all of the PAs, including Sam, was not predictive of proportional learning gains, thus reaffirming the need to pursue the quality over quantity of interactions with PAs. In total, results led us to consider learner affective response towards PAs as a distinguishing characteristic between the PAs and learner performance.

Results of the API indicated that there was no correlation between feelings of affect toward the PAs and proportional learning gains. However, there were numerous correlations between the proportion of the session spent interacting with PAs and affect. These correlations were predominantly negative in nature. This result may be explained in that the PAs often require the learners to engage in difficult and complex cognitive and metacognitive processes which involve a certain amount of effort and persistence. Moreover, though the system is adaptive, it is not yet capable of providing optimal real-time scaffolding. A major effort among several research communities is to use multichannel data to understand the complex nature of SRL processes in order to build sophisticated CAMM-sensitive ITSs capable of accurately detecting, tracking, modeling, and fostering SRL during complex learning [4]. For example, individual differences will influence learners' CAMM processes depending on prior knowledge, motivation, and regulatory flexibility. More specifically, self-regulating learners will exhibit more dynamic responses throughout their interactions with PAs and utilize adaptive CAMM, SRL, and emotion regulation processes; whereas learners who are not adept at self-regulating with low prior knowledge will require different types of scaffolding and threshold levels than self-regulating learners. Thus, the ability of PAs to efficiently respond to learners' needs in real-time may ultimately be dictated by individual learner characteristics. Future analyses will examine individual PA-learner interactions to understand the dynamics of affect, how students react to individual PA's interventions, the synergy of the multiple agent approach, and how these interactions impact learners' CAMM processes throughout the learning session. Subsequently, these results can be used to modify agent-behavior rules and system logic, as well as create dynamic thresholds that adapt in real-time. These advances will propel ITSs and PAs into the next generation of systems that are more intelligent and effective in supporting and fostering learners' CAMM SRL processes [7].

**Acknowledgements.** This study was supported by funding from the National Science Foundation (DRL 1431552). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## References

1. Azevedo, R.: Metacognition and multimedia learning. In: Mayer, R.E. (ed.) *Cambridge Handbook of Multimedia*, 2nd edn, pp. 647–672. Cambridge University Press, Cambridge (2014)
2. Biswas, G., et al.: Smart open-ended learning environments that support learners cognitive and metacognitive processes. In: Holzinger, A., Pasi, G. (eds.) *Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data*, pp. 303–310. Springer, Berlin (2013)
3. D'Mello, S.K., Graesser, A.C.: AutoTutor and affective autotutor: learning by talking with cognitively and emotionally intelligent computers that talk back. *ACM Trans. Inter. Intell. Syst.* **2**, 23–39 (2012)

4. Azevedo, R., et al.: Using trace data to examine the complex roles of cognitive, metacognitive, and emotional self-regulatory processes during learning with multi-agent systems. In: Azevedo, R., Aleven, V. (eds.) *International Handbook of Metacognition and Learning Technologies*, pp. 427–449. Springer, Amsterdam (2013)
5. Lester, J., et al.: Narrative-centered learning environments: a story-centric approach to educational games. In: Mouza, C., Lavigne, N. (eds.) *Emerging Technologies for the Classroom: A Learning Sciences Perspective*, pp. 223–238. Springer, Manhattan (2013)
6. Winne, P.H., Azevedo, R.: Metacognition. In: Sawyer, K. (ed.) *Cambridge Handbook of the Learning Sciences*, 2nd edn. Cambridge University Press, Cambridge (2014)
7. Taub, M. et al.: The role of pedagogical agents on learning: issues and trends. In: Neto, F., Souza, R., Gomes, A. (eds.) *Handbook of Research on 3-D Virtual Environments and Hypermedia for Ubiquitous Learning*, pp. xx–yy. IGI Global, Hershey (in press)

# Can Peers Rate Reliably as Experts in Small CSCL Groups?

Ioannis Magnisalis<sup>1</sup>(✉), Stavros Demetriadis<sup>1</sup>, and Pantelis M. Papadopoulos<sup>2</sup>

<sup>1</sup> Aristotle University of Thessaloniki, PO Box 114, 54124 Thessaloniki, Greece  
{imagnisa, sdemetri}@csd.auth.gr

<sup>2</sup> Aarhus University, 8200 Aarhus N, Denmark  
pmpapad@tdm.au.dk

**Abstract.** Research on the impact of peer rating (PR) has provided encouraging results, as a method to foster collaborative learning and improve its outcomes. The scope of this paper is to discuss peer rating towards two specific directions that usually are neglected in the CSCL field, namely: (a) coaching of objective anonymous peer rating through a rubric, and (b) provision of peer rating summary information during collaboration. The case study utilized an asynchronous CSCL tool with the two aforementioned capabilities. Initial results showed that peer rating, when anonymous, and guided, can be as reliable as off-line expert/teacher rating, with indications that this process can foster collaboration.

**Keywords:** Peer rating · Expert rating · Computer-supported collaborative learning · Asynchronous forum discussion

## 1 Introduction

Collaborative learning (CL) is important for students both for social and cognitive reasons [1]. Computer-supported collaborative learning (CSCL) is not simply implying the use of technology for communication purposes to enable CL, but aims to improve both CL peer skills and individual/group learning domain products. Efforts to implement reflection tools [2] in CSCL that foster peer interactivity (PI) and improve the collaborative learning process and outcome have been systematically reported in the literature, providing encouraging evidence on the impact of these methods to enhance student learning. In the current work, we study qualitative aspects of PI in a setting where students rate each other's posts in a Moodle forum. Peer rating (PR) is defined as the process through which students monitor and rate the performance of their fellow group members. PR reflection, assisted by visual feedback on PR, is defined as the cognitive and affective activities individuals engage in to explore their experiences and reach new understandings and appreciations of those experiences [3]. Models that capture both activity and domain aspects of PI are described in [4, 5]. In the current study, we use scheme of [4] to classify peer interactions, according to their qualitative characteristics.

Peer rating is a complex skill that does not effectively or efficiently emerge or develop in a spontaneous way [6]. To effectively and efficiently use PR, (a) simple PR tools should be used, and (b) PR process needs to be supported and guided [6].



Solely providing students with a tool to rate posts in a discussion forum, is probably not enough to alter group collaboration balance or change students' rating standards. Visual feedback can support reflection and plays an important role in individual learning processes [7], as well as in collaborative learning processes. Small rating deviations among peers may suggest that group is led to common understanding and awareness. Enhanced group awareness can lead to more effective and efficient collaboration [8–10]. The data from [11, 12] revealed how computer mediation can improve the reliability and validity of peer review activities, while simultaneously improving their functionality. Intra-class correlation coefficient (ICC) [13] was used to assess SWORD's score reliability comparing teachers' and students' ratings.

Rating has been accompanied by visualization techniques in CSCL already. In [14], authors present augmented group awareness tools supporting collaborative learning. The group awareness tool provided to the small groups in one of the experimental conditions was embedded into the online discussion environment. Taking about visualizations, in [15] students, before contributing in the discussion, they gathered information about current balance over participation while in [16] authors tried to boost student motivation by building a positive sense of competition using a representation of average class performance.

Here, we present the study of a technology system that employs visualized peer rating data on posts from a Moodle forum. The main research question of this study can be stated as such: *“To what extent can peer rating be reliable, when compared with expert rating?”*

## 2 Method

### 2.1 Participants, Learning Environment and Procedure

The study was conducted in a Second Chance school in Thessaloniki, Greece. The participants were 176 students (ages 18–50,  $M = 42$ ,  $SD = 3.7$ ), with most of them having low familiarization level with online communication tools; only 13 had used forum/chat tools before -but none of them for educational purposes- with an average computers and information literacy level of 3.8 out of 10 (based on 35 questions, similar to [17], and designed according to the B-Tile [18]). The students were randomly distributed into 44 groups of 4 peers.

In this work, each group member provided anonymous feedback to peers within the same group. Each student had been attributed to a pseudo name in the system and rated posts of peers anonymously. All ratings were calculated, summarized and visualized as feedback to collaborating peers (see Fig. 1, PR at both group and individual levels are depicted). Thus, Moodle was enhanced with: (a) an anonymous peer rating tool (PRT) based on a rubric-based qualitative model, (b) a shared visualization tool (SVT) used as a feedback tool for peer ratings with an intuitive interactive interface supporting both individual and group awareness. PRT allows the group members to rate peer cognitive contribution chunked into posts, and shares this information anonymously with all group members. Rating is the parameter our fPIV (flexible PI visualization) system monitors.

Rating in fPIV is based on the same models applied in [4]. Specific examples were given in a complete guiding manual to peers on how to rate.

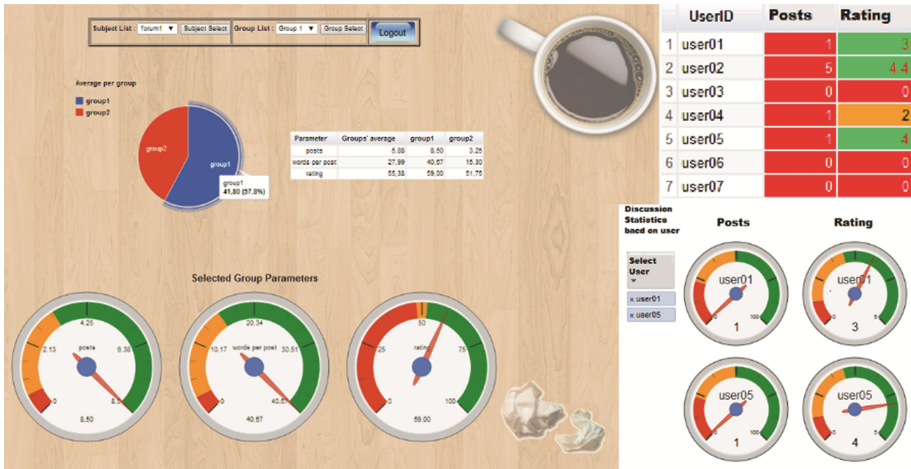


Fig. 1. Data representation of group collaboration in Moodle-forum.

The study lasted 4 weeks (May–June, 2015). The subject was the history of Thessaloniki. The study had 6 phases: pre-test, familiarization, study, discussion, post-test, and interviews. Pre-test was a written test, containing 20 close-type questions (e.g. “When was the White Tower built?”) measuring prior subject knowledge. After pre-test, the students were introduced to the learning environment and they were allowed a 4-week prolonged study period to familiarize themselves with the system and the peer rating process on a test discussion subject. During the study phase that followed and which lasted 4 more weeks, the students had to study 10 most important sightseeing of the city within a presentation deliverable. Collaborative work and deliverable could only be produced within the Moodle forum. Peer rating was obligatory and a grade penalty was introduced for the students that did not rate their peers’ posts. Then, the students filled a post-test containing 20 questions similar to the ones in the pre-test, and a questionnaire focusing on the peer rating process and the tools used. The study was concluded by interviews, in which the students had the opportunity to further elaborate their thoughts on the whole activity.

2.2 Data Collection and Analysis

Logs & expert ratings: Students’ activity within each tool was monitored. Data logs per peer included: the number of posts sent, the number of posts read, time of posts read, access time of resources like forum or visualization tool, time and duration of visualizations viewed. Moreover, three teacher experts rated independently and offline the posts of all collaborating peers a few days after case study and system was closed. These ratings were logged in the same PRT tool we introduced along Moodle-based asynchronous forum used for group discussions.

**Interviews:** All students were interviewed individually for 15 min with focus on deeper understanding students' comments and suggestions. The interviews were transcribed and the classification of conclusions concerning both interviews and open-type question answers was the by-product of interviews analysis.

**Analysis:** Peer ratings were compared to expert ratings. The two-way random average measures (absolute agreement) intra-class correlation coefficient (ICC) was used as a measure of inter-rater reliability. In general, data analysis followed the principles of a mixed evaluation method [19].

### 3 Results

#### 3.1 Logs and Expert Ratings

L1: The students read the received ratings, then they rate and finally post. This is the most common strategy followed by peers when interacting within the fPIV system provided. Students post or rate, after reading PR feedback, 97 % of the time, while they read PR feedback after posting or rating 73 % of the time.

L2: The students are likely to use the PR visual representation tool during the whole activity to monitor their collaboration. The average time spent (per student, per day) on PR feedback information is approximately 5 min (almost 4 % of their activity time when logged in the system).

L3: Students did not need to spend much time in reading guide instructions (average: 5.7 min), and during collaboration they seemed to have "internalized" the rating guidance given (average: 2.3 views).

L4: Providing PR seems to trigger PI (i.e. posts and replies). A new post appears after a student has received peer rating(s). This is related to L1.

L5: ICC was very high for ratings among peers (.91), expert teachers (.99), and peers and teachers combined (.87).

#### 3.2 Interviews

The list below shows the most important findings recorded during the interviews.

I1: Providing PR was an easy task (93 %) because of the guiding rubric used.

I2: The strategy of peers was driven by PR process (83 %). The students first studied the ratings of their peers and then formed their strategy posting, replying and rating. This finding is in accordance with L1 above.

I3: Raw table data should be accompanied by simple visualizations like bar charts preferably (81 % of students). This helps students evaluate the raw data and draw conclusions, promoting self and group awareness during collaboration.

I5: Students wanted to have some statistical data depicting the PR feedback of the whole class (74 %). Thus, PR presentation is covering three levels (see Fig. 1): (a) Individual, (b) Group, (c) All groups working in parallel in an activity.

I6: Students (93 %) opted for anonymous PR (otherwise rating would be biased).

## 4 Discussion-Conclusion

In relation to the major research question posed “*To what extent can peer rating be reliable, when compared with expert rating?*” we can state that:

A tool like PRT, when guided and anonymous, can motivate and support effectively students perform reliable on-line rating (when compared to that of external off-line expert raters)

A tool like SVT can foster collaboration balance in small groups (up to 4 peers).

PRT anonymously shares all perceived and received ratings of peers, in order to make them more aware of the collaboration process and the way peers rate his/her posts (L1, L3, I6). SVT stimulates individual reflection on discrepancies between self and received peer ratings, and stimulates peers to reflect collaboratively upon their group performance. This reflection process allows group members to reach a shared view about what can be referred to as valuable post contributions (see L2, L4, I5).

Because group members’ peer ratings are shared in SVT, all group members receive information on their peers’ contributions. The strength of PRT and SVT emerge from its ability to make implicit aspects of collaboration (e.g., rated posts among peers) explicit for all group members. PRT and SVT enhance students’ awareness of performance, by providing them with explicit information concerning their performance (e.g., contributing low quality work). Based on aforementioned findings we can state that a PR process that is anonymous and guided can provide a good experience to students. Students’ ratings compared to teachers’ ratings exhibited scarce and small deviations. Relevant tests reveal that on-line PR, if performed with these prerequisites, can be as respectable as offline expert rating (L5). That is the key element that allows for building an on-line reflection tool like SVT.

A shared visualization tool helps discussion stay on-task. From log files (see L1), we notice that a peer before writing a post and/or rating peers, he/she studies on PR visual feedback. This is aligned with study [15], where students, before contributing in the discussion, they gathered information about current balance over participation and formed a strategy to achieve a balanced participation over their discussion.

In this work, we reached similar conclusions as in [16]. There, authors tried to boost their motivation by building a positive sense of competition using a representation of average class performance. That is why -in fPIv- we have opted to use information to the student not only for the group he/she is in but also for the class he/she participates in. In our work, we notice that enhancing interpersonal behavior positively affects the group’s balance (L2, L3 and L4) positively.

Overall, the effects of PRT and SVT on group members’ individual behavior and their social group performance look very promising. To our knowledge, there is no concise conclusion in previous research to what extent peer rating assessment and reflection feedback affect group behavior and performance, and what kind of reflection feedback lead to effective reflection processes (e.g., [6, 7]). Therefore, this work contributes as an instigator towards studies that examine a combination of guided and anonymous peer rating supplemented with visualized feedback.

## References

1. Dillenbourg, P.: What do you Mean by Collaborative Learning? *Collaborative-Learning: Cognitive and Computational Approaches*, pp. 1–19. Elsevier, Oxford (1999)
2. Vatrupu, R., Teplovs, C., Fujita, N., Bull, S.: Towards visual analytics for teachers' dynamic diagnostic pedagogical decision-making. In: *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, pp. 93–98. ACM (2011)
3. Boud, D., Keogh, R., Walker, D.: *Reflection: Turning Learning into Experience*. Kongan Page, London (1985)
4. De Wever, B., Schellens, T., Valcke, M., Van Keer, H.: Content analysis schemes to analyze transcripts of online asynchronous discussion groups: a review. *Comput. Educ.* **46**, 6–28 (2006)
5. Veerman, A., Veldhuis-Diermanse, E.: Collaborative learning through electronic knowledge construction in academic education. In: *Collaborative learning, reasoning, and technology*, pp. 323–354 (2006)
6. Prins, F.J., Sluijsmans, D.M., Kirschner, P.A., Strijbos, J.W.: Formative peer assessment in a CSCL environment: a case study. *Assess. Eval. High. Educ.* **30**(4), 417–444 (2005)
7. Chen, N.S., Wei, C.W., Wu, K.T., Uden, L.: Effects of high level prompts and peer assessment on online students' reflection levels. *Comput. Educ.* **52**(4), 283–291 (2009)
8. Janssen, J., Erkens, G., Kirschner, P.A.: Group awareness tools: it's what you do with it that matters. *Comput. Hum. Behav.* **27**(3), 1046–1058 (2011)
9. Druskat, V.U., Wolff, S.B.: Effects and timing of developmental peer appraisals in self-managing work groups. *J. Appl. Psychol.* **84**(1), 58 (1999)
10. Kollock, P.: *The Economies of Online Cooperation. Communities in Cyberspace*, p. 220. Routledge, New York (1999)
11. Cheng, R., Vassileva, J.: Design and evaluation of an adaptive incentive mechanism for sustained educational online communities. *User Model. User-Adap. Inter.* **16**, 321–348 (2006)
12. Cho, K., Schunn, C.D., Wilson, R.W.: Validity and reliability of scaffolded peer assessment of writing from instructor and student perspectives. *J. Educ. Psychol.* **98**(4), 891 (2006)
13. McGraw, K.O., Wong, S.P.: Forming inferences about some intraclass correlation coefficients. *Psychol. Methods* **1**, 30 (1996)
14. Dehler, J., Bodemer, D., Buder, J., Hesse, F.W.: Guiding knowledge communication in CSCL via group knowledge awareness. *Comput. Hum. Behav.* **27**, 1068–1078 (2011)
15. Bachour, K., Kaplan, F., Dillenbourg, P.: *Reflect*: an interactive table for regulating face-to-face collaborative learning. In: Dillenbourg, P., Specht, M. (eds.) *EC-TEL 2008. LNCS*, vol. 5192, pp. 39–48. Springer, Heidelberg (2008)
16. Falakmasir, M.H., Hsiao, I.H., Mazzola, L., Grant, N., Brusilovsky, P.: The impact of social performance visualization on students. In: *IEEE 12th International Conference on Advanced Learning Technologies (ICALT)*, pp. 565–569 (2012)
17. Computer Literacy Survey. [www.fscj.edu/tutorials/media/online\\_skill\\_assess.pdf](http://www.fscj.edu/tutorials/media/online_skill_assess.pdf) (2016). Accessed 2 Jan 2016
18. Beile, P.M.: *Development and validation of the Beile Test of Information Literacy for Education (B-TILED)*. College of Education, Central Florida University, PhD Thesis (2005)
19. Creswell, J.W.: *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, 2nd edn. Sage Publications Inc., London (2002)

# Peer Review in Mentorship: Perception of the Helpfulness of Review and Reciprocal Ratings

Oluwabunmi Adewoyin<sup>1</sup>(✉), Roberto Araya<sup>2</sup>, and Julita Vassileva<sup>1</sup>

<sup>1</sup> University of Saskatchewan, Saskatoon, Canada  
bunmi.adewoyin@usask.ca, jiv@cs.usask.ca

<sup>2</sup> CIAE University of Chile, Santiago, Chile  
roberto.araya.schulz@gmail.com

**Abstract.** Peer review is the main mechanism for quality evaluation and peer-mentoring in the research community. Yet, it has been criticized with respect to its summative function, as being prone to bias and inconsistency and approaches had been proposed to improve it (e.g. double blind review). However, relatively less attention has been paid on how well it meets its formative objective, i.e. providing useful feedback to help the authors improve their quality of work. In our previous work we proposed a modified peer review process, which involved a back-evaluation of reviews by the authors. This paper reports the results of a study of the application of this peer review process to support a group of teachers in Chile engage in group peer mentorship in the context of a summer continuing education course. The objectives are to find out if authors reciprocate their reviews feedback in the back-evaluation given to their reviewers, and if the review length affects the helpfulness and authors' satisfaction with the reviews. Our results showed that peers did not reciprocate their ratings and review length did not affect peers' satisfaction with the reviews.

**Keywords:** Peer review · Continuing education · Collaborative learning · Peer mentorship

## 1 Introduction

Peer review is a veritable means of judging the quality of product or entity by a community of peers [9, 12, 20]. Besides its use in judging the quality of a research work, peer review also helps in mentoring researchers, as authors, to further develop their work and knowledge by providing competent peer-criticism [7]; to develop reviewers' ability to provide fair and constructive criticism of their peer's work by seeing the other reviews of the same paper that they have reviewed. Peer review has been used to evaluate and mentor peers in higher education, teaching, medicine, and accounting [4, 10, 15]. However, research had shown that it is prone to bias and inconsistencies [16]. Proposals to improve the traditional peer review process include blind peer review [18], open peer review model [18], training of reviewers [14] and the back-evaluation of reviews by the authors of the papers that are reviewed [1, 2]. These ideas were proposed to support the two main objectives of peer review – formative and

summative. That is, to assess the quality of scholarly work (summative feedback) and also to provide useful feedback to help the recipients of the review to improve the quality of their work (formative feedback) [2]. However, most of the reported results of the improvements are about the summative objective. Little has been done to investigate if these ideas could support the formative objective of the peer review process [21], i.e. if these ideas could help both authors and reviewers to develop better skills.

In this study, we implemented the peer review framework proposed by [1] to help a group of Chilean teachers to improve their understanding of how the new didactic methodologies can be applied to their classes for implementing the recently changed national curriculum. The objectives are – (1) *to investigate if peers, both as authors and reviewers, perceive that the peer review process helps them learn and improve their understanding skills, and* – (2) *to investigate if peers are inclined to help each other using the peer review system, and not just reciprocating ratings in order to formally fulfill their part in the peer review process.*

The rest of this paper is organized as follows. Section 2 contains the review of related work while Sect. 3 contains the study method. Section 4 describes our case study experiment and results, while Sect. 5 concludes the paper.

## 2 Previous Work

Peer review facilitates learning, teamwork and the development of professional skills [2, 4–6, 8, 10, 11, 15, 17]. In education, peer review is a veritable means of ensuring individual accountability in group work, thereby, discouraging free-riding [10]. Existing research had shown that feedback from peers is a strong motivator for learners to produce higher quality work than feedback from their instructors [2]. Also according to [15], peer review in education enables learners to take control of their learning and helps in boosting their confidence in their growing ability to recognize high quality work when they critique the work of others. To successfully implement peer review in education, [3] suggested setting expectations, involving students in the implementation, periodic formative assessments, preparing learners for feedback, balanced use of anonymous or open reviews, and customizing the process.

Despite its benefits discussed in the literature, peer review is believed to be susceptible to bias, inconsistencies, and can generate a sense of socio-emotional discomfort for peers who have to critique their peers' work [16]. Many ideas have been proposed to improve the peer review process [1, 2, 14, 16, 18]. In [1], we proposed a five-stage peer review procedure including writing, peer feedback on the writing, back-evaluation of the feedback, rewriting and publishing. The reason for adding the additional step of “back-evaluation” or authors' feedback on the received feedback is that it will motivate reviewers to do a better job and allow authors to express and clarify their stance, thus supporting the learning and improvement of both the reviewers and the authors. In [2] we confirmed that by providing feedback to reviewers, peers are motivated to give thorough and helpful reviews. However, this research did not investigate the reliability of the feedback given by peers to their reviews. While we want to ensure that there is reciprocity of learning by making authors back-evaluate the reviews of their work [13], we do not want peers to reciprocate their feedbacks,



i.e. follow a “tit-for-tat” strategy, which would be equivalent to ‘gaming’ the peer-review system. Presently, no existing work measures the reciprocity of feedback in peer review process that involves back-evaluation of reviews. Therefore, this study investigates the relationship between back-evaluation and peer review feedback, which is a measure of the reliability of feedback. We also measure the subjective quality and helpfulness of the feedback as perceived by the peers.

### 3 Study Methods and Tool

In 2013, the Chilean government funded CIAE, the Educational Research Center of the University of Chile, to conduct day-long seminars to help school teachers from the entire country to implement the new national curricula in three disciplines – Mathematics, Music and Language. In these seminars, international and national speakers of each discipline gave lectures on the new didactic methodologies. About half of the four thousand teachers that attended the seminar attended in person, while the rest by video streaming. Then, 168 Mathematics teachers, 73 Music teachers and 43 Language teachers voluntarily engaged in the peer review study, which lasted two months. We provided the modified peer review process with back evaluation [2], which was implemented in a peer-review system designed for the purpose of this event. Back-evaluation is the process whereby peers as authors evaluate the reviews given to their work, as a feedback to the reviewers as well.

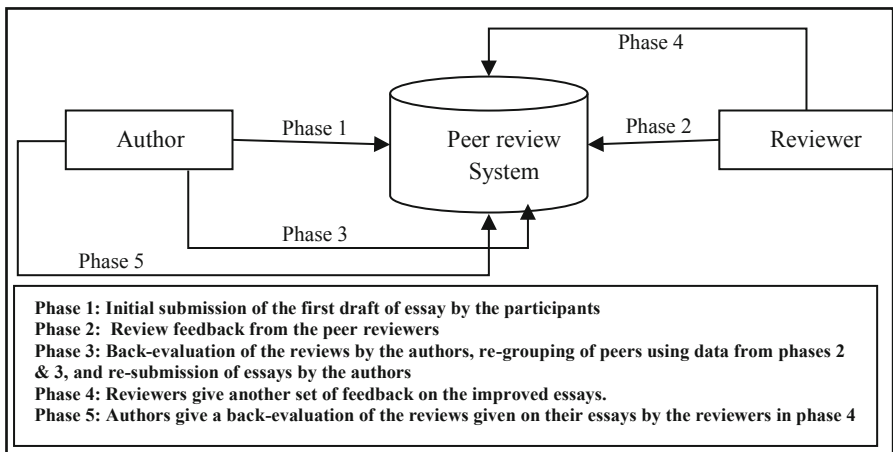
The general goal of the study was to find out if the modified peer review system could support the participants’ understanding of the new didactic methodologies to implement the new curricula. In addition, we set to answer the following specific questions:

1. Can we judge the quality of reviews by their length?
2. What factors determine the success of the peer review session?
3. Does the back-evaluation of reviews encourage reciprocity of ratings in the peer review feedback?

To ensure objective and quality reviews, the modified peer-review process [2] also suggests that review assignments are done not randomly, but in groups of at most 4 peers (to keep manageable reviewing load), with mixed abilities, so that weaker peers can learn from stronger ones by operating within their zone of proximal development [19]. Double blind review is used in each group to avoid bias and reciprocation over different review sessions. At the end of each peer review session, peers evaluate the learning they gained and the general helpfulness of the session in improving their skills both as authors and as reviewers. To implement the modified peer-review process, within each discipline – Mathematics, Language and Music – the participants were classified into groups of four. Within each group, the papers were evaluated through a double-blind peer review process, with each participant in the position of both an author and a reviewer. After the initial draft submission (Phase 1 in Fig. 1, 1<sup>st</sup> Session), the participants were encouraged to give constructive feedback to the drafts assigned for them to review (Phase 2, 1<sup>st</sup> Session). After receiving the reviews, the author of each paper had



to rate the helpfulness of the reviews of their paper (Phase 3, 1<sup>st</sup> Session). In addition, during Phase 3, the reviewers were asked to provide feedback on whether they improved their skills for constructive critiquing from the reviewing session of each paper they reviewed by seeing the other two reviews of the draft paper and from the ensuing discussion. Then peers were invited to revise their drafts, considering the received feedback during Session 1 and to resubmit it again following the same procedure (2<sup>nd</sup> Session). Thus there were two peer review sessions, after which the participants were asked to fill an exit questionnaire about their experience and learning from the two peer review sessions, both as authors and reviewers. We defined five phases in the two peer review sessions (see Fig. 1). For the purpose of this paper, we present our results in the first and second reviewing sessions, Session 1 (including Phases 2 and 3), and Session 2 (Phases 4 and 5).



**Fig. 1.** The implemented modified peer review process

## 4 Results and Discussion

Below are the results obtained presented as they relate to the questions we set out.

*1. Can we judge the quality of reviews by their length?* Since the back-evaluation rating is the author's perception of the quality and helpfulness of their reviews, we want to know if the review length affects these ratings. Table 1 shows the lengths of the reviews and the correlation between the review length (measured by the number of words in it) and the back-evaluation ratings provided by the authors of the essays that were reviewed.

First, we observe that the average review length got shorter in the 2<sup>nd</sup> session. Yet the standard deviation (St. dev) is rather high, so it is hard to speak of a trend. The correlation values vary and are mostly negative (i.e. longer reviews are rated lower), but nearly always close to 0. So for most reviewers, there is no correlation between their back-evaluation ratings and the number of words in their reviews. The one instance of

**Table 1.** Average lengths of the reviews & standard deviation; Correlation btw back-evaluation & length of review; R-Squared ( $R^2$ )

	1st Session		2nd Session		Correlation (r)		R-Squared( $R^2$ )(X 100)	
	Average	St.dev	Average	St.dev	1st Session	2nd Session	1st Session	2nd Session
Mathematics	89.8905	34.4663	85.1688	30.4317	-0.09359	-0.07999	0.00876	0.0064
Music	108.6667	35.4865	103.3945	37.5208	-0.09513	0.04326	0.00905	0.0019
Language	99.4531	32.0495	83.2857	41.9838	-0.03053	-0.20591	0.00093	0.0424

weak negative correlation (-0.20591) for the language group in the 2nd session, was interesting to explore. We calculated  $R^2$  in order to determine the impact of how well the review length could predict the back-evaluation ratings. The R-squared analysis results showed that the review length is a weak predictor of the back-evaluation ratings, with the highest value being  $R^2 = 4.24 \%$ . Therefore, our results show that lengthy reviews are not perceived as being more helpful and with higher quality by the authors.

2. *What determines the success of the peer review session?* We followed the principle that one understands something better when one teaches it to someone else. Therefore, we proposed that every participant is both an author and a reviewer with the hope that after the session, everyone has benefitted and learned from the experience in both roles. In our experiment, teachers as authors are expected to learn how to implement the new curriculum and as reviewers, they are expected to learn how to constructively criticize and help their peers.

At the end of the experiment, we asked participants to provide feedback on their learning experience and the helpfulness of the peer review session, using a 7 – point Likert scale rating (1 is the worst and 7 – the best), which is the typical scale used to grade tests and homework in Chile. These ratings constitute the explicit measure of the success of the peer review session. We did not have feedback from the participants in the Language group, but the participants from the Mathematics and Music groups perceived that they learned from the peer review system and also found the system to be helpful (Table 2). Their comments provide possible factors for the success and suggested that they enjoyed the experience from the peer review session. e.g. *“It is very valuable because you can reach for excellence as it allows you to be corrected by your peers”* (study participant).

We also asked the participants if they would be happy to recommend the peer review system to their schools (Table 2, col. 3). Some of the comments given by the participants showed that they were not questioning the benefits of introducing a

**Table 2.** Average helpfulness and learning (on the scale of 1- min., to 7-max.); % Recommendation for the peer review system

	Helpfulness			Learning			Recommend?	
	Average	Stdev	Conf. Interv.	Average	Stdev	Conf. Interv.	Yes	No
Mathematics	6.0676	1.0248	±0.15496	6.0676	1.275	±0.19279	~97 %	~3 %
Music	6.7308	0.4523	±0.10376	6.4615	0.9047	±0.20753	100 %	0 %

peer-review system to their schools, but whether their country's education policies provide the motivation for teachers to introduce such collaborative task to their students and whether the existing inequalities in the educational system and the preparedness of the teachers can provide a basis for useful peer-feedback within such a system.

3. *Does the back-evaluation encourage reciprocation of ratings?* The authors were encouraged to provide a back-evaluation of the reviews of their essay, given by the reviewers. Participants used pseudonyms in both peer review sessions, in order to mask their identities and also assign ratings from both peer review sessions to their pseudonyms. With the blind-review, we expected that authors would not try to reciprocate the ratings that the reviewers gave to their essay in the back-evaluation. To confirm our expectations, for each of the three groups, we ran a correlation test on the ratings given by reviewers and the back-evaluation ratings given by authors to the reviews they received (Table 3). If the authors were trying to reciprocate reviewers' good ratings and positive comments or retaliate the reviewers' low ratings and critical comments, we would find a positive correlation.

**Table 3.** Correlation between back-evaluation and review ratings (merging reviewers)

	Correlation		R-Squared ( $R^2$ ) (X100)	
	1st Session	2nd Session	1st Session	2nd Session
Mathematics	0.1899	-0.0331	0.03605	0.0011
Music	0.1808	0.2417	0.03270	0.05842
Language	-0.0161	-0.0331	0.00026	0.0011

Our results showed mixed weak correlations ( $-0.03$  and  $+0.18$ ) and one case with a slightly higher correlation value of  $0.2417$  in the 2nd session of the Music group. We calculated  $R^2$  in order to determine how well the review feedback predicts the back-evaluation ratings. The results showed that the review feedback is a very weak predictor of the back-evaluation ratings, in the Mathematics and Music groups, with the highest value being  $R^2 = 5.842\%$  for the 2nd session of the Music group. It seems that authors were not trying to reciprocate the reviews in the back-evaluation, but were only providing helpful and truthful feedback to their peers.

## 5 Conclusion

Many approaches have been proposed to improve the peer review system. One of these approaches is the modified peer review process, which involves the back-evaluation of reviews. However, little has been done on measuring the effectiveness of this modified peer review process in fulfilling its formative objective. Therefore, this study fills this gap by measuring satisfaction with the modified peer review process using the feedback from both authors and reviewers, checking if the back-evaluation phase encourages reciprocity/retaliation by authors to reviewers, and investigating the role of the length of the review as a predictor of the review quality and helpfulness. Our results confirmed

that providing opportunity for authors to back-evaluate the reviews of their work does not necessarily encourage reciprocation of ratings. Instead, it was confirmed that participants were just being helpful and honest with the feedback they provide as back-evaluation of their reviews. Also, we were able to confirm that the quality of the reviews, as perceived by the recipients, cannot be evaluated by their length.

One major limitation to this study is that we could not test our modified peer review process using a controlled experiment in comparison with a standard peer-review process to evaluate if the back-evaluation makes a difference in the learning experience. In the future, we will seek a large group of participants that can be divided into control and experimental groups, where we will be able to compare their review quality, helpfulness of the feedback and general experience for both groups using both log and qualitative data.

**Acknowledgements.** We thank Paulina Sepúlveda for implementing the peer review system and to Abelino Jiménez and Josefina Hernández for initial data preprocessing. We thank Francisco Gutierrez for his help with the translation of user comments from Spanish to English. This research was possible with funding from NSERC Discovery Grants Program to the 3rd author and from Basal Funds for Centers of Excellence Project BF 0003 from the Associative Research Program of CONICYT to the 2nd author.

## References

1. Adewoyin, O., Vassileva, J.: Can online peer-review systems support group mentorship? In: Lane, H., Yacef, K., Mostow, J., Pavlik, P. (eds.) AIED 2013. LNCS, vol. 7926, pp. 737–741. Springer, Heidelberg (2013)
2. Adewoyin, O., Vassileva, J.: Ethics of scientific peer review: are we judging or helping the review recipients? In: IEEE International Symposium on Ethics in Engineering, Science, and Technology, Ethics 2014 (2014)
3. Cestone, C.M., Levine, R.E., Lane, D.R.: Peer assessment and evaluation in team-based learning. *New Directions Teaching Learn.* **116**, 69–78 (2008)
4. Farrell, K.: Collegial feedback on teaching: a guide to peer review. Centre for the Study of Higher Education, The University of Melbourne (2011)
5. Fernandez, C.E., Yu, J.: Peer review of teaching. *J. Chiropractic Educ.* **21**(2), 154–161 (2007)
6. Gutknecht-Gmeiner, M.: Peer review in education. Peer review in initial VET, Leonardo da Vinci project, Austria (2005)
7. Houry, D., Green, S., Callaham, M.: Does mentoring new peer reviewers improve review quality? A randomized trial. *BMC Med. Educ.* **12**(1), 83 (2012)
8. Hutchings, P.: Peer review of teaching. “From Idea to Prototype”. In: AAHE Bulletin, November 1994
9. Kronick, D.A.: Peer review in 18th century scientific journalism. *J. Am. Medical Assoc. (JAMA)* **263**, 1321–1322 (1990)
10. Levine, R.E.: Peer evaluation in team-based learning (2010). [https://training.health.ufl.edu/handouts/FacDev/TBL\\_Chapter9.pdf](https://training.health.ufl.edu/handouts/FacDev/TBL_Chapter9.pdf). (Accessed on 17 October 2015)
11. Pearce, J., Mulder, R., Baik, C.: Involving students in peer review: case studies and practical strategies for University teaching. University of Melbourne, Victoria (2009). [http://www.cshe.unimelb.edu.au/resources\\_teach/teaching\\_in\\_practice/docs/Student\\_Peer\\_Review.pdf](http://www.cshe.unimelb.edu.au/resources_teach/teaching_in_practice/docs/Student_Peer_Review.pdf). (Accessed on 17 October 2015)

12. Ranalli, B.: A prehistory of peer review: religious blueprints from the hartlib circle spontaneous generations. *J. History Philos. Sci.* **5**(1), 12–18 (2011)
13. Sachs, J., Parsell, M.: Peer review of learning and teaching in higher education. *British J. Educ. Technol.* **45**(3) (2014)
14. Schroter, S., Black, N., Evans, S., Carpenter, J., Godlee, F., Smith, R.: Effects of training on quality of peer review: randomised controlled trial. *BMJ* **2004**(328), 673 (2004)
15. Searby, M., Ewers, T.: An evaluation of the use of peer assessment in higher education: a case study in the school of music. *Assess. Eval. Higher Educ.* **22**(4), 371–383 (1997). Kingston University
16. Smith, R.: Peer review: a flawed process at the heart of science and journals. *J. R. Soc. Med.* **99**, 178–182 (2006)
17. Turner, S.; Perez-Quinones, M.A., Chase, J.: Peer review in CS2: conceptual learning. In: SIGCSE 2010 Proceedings of the 41<sup>st</sup> ACM Technical Symposium on Computer Science Education, pp. 331–335
18. Van Rooyen, S., Godlee, F., Evans, S., Black, N., Smith, R.: Effect of open peer review on quality of reviews and on reviewers' recommendations: a randomised trial. *Br. Med. J.* **318**, 23–27 (1999)
19. Vygotsky, L.S.: *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, Cambridge (1978)
20. Wagner, W., Steinzor, R.: *Rescuing Science from Politics Regulation and the Distortion of Scientific Research*, 1st edn. Cambridge University Press, Cambridge (2006)
21. Xiong, W., Litman, D., Schunn, C.: Assessing reviewers' performance based on mining problem localization in peer-review data. In: Proceedings of the Third International Conference on Educational Data Mining (EDM 2010), pp. 211–220 (2010)

# Motivational Gamification Strategies Rooted in Self-Determination Theory for Social Adaptive E-Learning

Lei Shi<sup>(✉)</sup> and Alexandra I. Cristea

The University of Warwick, Coventry, CV4 7AL, UK  
{Lei.Shi,A.I.Cristea}@warwick.ac.uk

**Abstract.** This study uses gamification as the carrier of understanding the motivational benefits of applying the Self-Determination Theory (SDT) in social adaptive e-learning, by proposing *motivational gamification strategies* rooted in SDT, as well as developing and testing these strategies. Results show high perceived motivation amongst the students, and identify a high usability of the implementation, which supports the applicability of the proposed approach.

## 1 Introduction

Social adaptive e-learning proposes that besides receiving personalised content, creating content and interacting with peers can also motivate learning activities. Apart from modeling students themselves, e.g., via knowledge level and preference, social adaptive e-learning also models *their relations*. The social dimension allows for new personalised recommendations, such as which groups to join, or which peers to talk to, and thus mitigates the isolation between students.

Gamification describes an efficient way of utilizing game design elements to motivate learning activities [6]. It is another area that potentially provides motivational benefits in e-learning. As gamification and social e-learning have various common mechanics, such as collaboration, discovery and achievement [12], their combination may have greater impacts. Some studies showed benefits brought by this combination [13], but very few apply these features based on *solid theoretical fundamentals*, with outcomes evaluated against these theoretical concepts. This study addresses this gap by exploring how to approach gamification in social adaptive e-learning *in a systematic way*, based on the theoretical underpinning of the Self-Determination Theory (SDT) [7]. In particular, we propose *motivational gamification strategies* rooted in SDT, for a high level of perceived motivation amongst students.

## 2 Related Work

*Social Adaptive E-Learning*: Learning is known as an intrinsically social endeavour [8], and the social facets of learning are described by a variety of theoretical frameworks [18]. The social learning theory postulates that learning is a cognitive process, which can occur through observation and imitation in a social context, and can be influenced

by intrinsic reinforcement, as a form of internal rewards, such as satisfaction and a sense of accomplishment [2]. Social techniques become increasingly popular in e-learning. They can attract students to interact with peers, and generate trails for peers to follow. Not only can they promote students to participate in various learning activities, but they can also motivate students to create learning content. This study establishes a clear connection to motivational triggers via *grounding in motivational theories* and evaluation of motivational effects.

*Self-Determination Theory (SDT)*: Successful social e-learning requires mechanisms to assist students in directing their own learning and having a high level of motivation to participate in meaningful interactions [10]. In e-learning, motivation initiates and maintains goal-oriented behaviours, to effectively achieve learning goals [1]. This allows students to take a self-motivating role of participating in determining their own learning paths, yet requiring support for a self-determined approach. SDT is widely and is one of the only motivational theories that focuses on the degree to which individual behaviours are self-determined and self-motivated [7], by proffering that individuals become increasingly more self-determined and self-motivated when three basic innate needs are fulfilled: (1) **Autonomy**: a sense of internal assent of one's own behaviours; (2) **Competence**: controlling the outcome and experience mastery; and (3) **Relatedness**: a sense of connection and interaction with others within a community. A social adaptive e-learning system that fulfills all these three basic innate needs is expected to sustainably increase the students' intrinsic motivation, leading to an efficient self-determined learning experience [15]. This study *applies* SDT in the design of the *motivational gamification strategies*, to fulfill students' basic innate needs and thus to foster intrinsic motivation.

*Gamification*: Gamification incorporates game thinking [17] (a game-like approach to aesthetics and usability) and game elements (elements from digital games, e.g., avatars, badges, progression bars, urgent optimism, and behavioural momentum) in a non-game system, and aims to achieve certain goals, such as learning, other than just entertaining players. Preferring to learn through games results from students' motivation in playing games [14], where they enjoy the learning system and like to continue using it. Studies have proposed guidelines for facilitating SDT in gamified e-learning systems [3], showing also positive impacts on learning performances. Yet, gamification has been criticised for its "overjustification effect", which occurs when an expected external incentive demotivates students with already existent high intrinsic motivation [5]. Evidence suggests that an increased extrinsic motivation might reduce the learning performance [4]. Our study explores a "light gamification" approach, rooted in SDT, to promote *intrinsic motivation*, rather than a "full-fledged gamification" approach that may "over-gamify" the e-learning system.

### 3 Motivational Gamification Strategies

Based on the above, the *motivational gamification strategies* defined in this study are classified into three groups, towards respectively fulfilling students' three basic innate needs: *autonomy*, *competence* and *relatedness*, as detailed below.

*To fulfill the Autonomy Need.* Experiencing *autonomy* means feeling in charge of one's behaviour. We suggest providing meaningful, flexible choices, such as learning goals and paths for achieving them, and learning peers to interact with (via various interaction tools), to continuously balance their curiosity, skills and goals against a finite pool of resources. This way, students can feel their behaviour as based on their own intentions, so that they may adhere to desired behaviours in certain contexts. In addition, to reduce the "overjustification effect" and maintain students' intrinsic motivation, it is important to provide intrinsic choices for voluntary behaviour [5], e.g., between competition and collaboration, as students usually tend to quickly notice the loss of autonomy (being controlled), which can demotivate them. To summarise, a system could implement the following *autonomy-related gamification strategies*: **A1.** A set of learning goals with clear descriptions and multiple paths to achieve each; **A2.** Various interaction tools to complete a task; **A3.** Clear, immediate and positive feedback for learning activities; **A4.** Meaningful options with consequences; **A5.** Customizable learning context that can be adjusted by students themselves.

*To fulfill the Competence Need.* Experiencing *competence* means a feeling of achieving mastery of skills and confidence in the current context, where cognition and expectations are consistent with system responses, to obtain further skills and confidence with relative ease. We suggest to provide direct and positive feedback, optimal challenges and freedom of demeaning evaluations. When experiencing enjoyment, students may become so intrinsically motivated that they not even realise completing a complicated task, or achieving a difficult learning goal [5]. We thus suggest offering interesting challenges combining clear rules and goals. We further suggest to "chunk" a learning goal into small and achievable pieces, and gradually increase the difficulty during the learning process, so that students are aware of every 'instant' achievement, feel the increase of skills, and make decisions accordingly and frequently. To summarise, a system could implement the following *competence-related gamification strategies*: **C1.** Reasonable small chunks of learning goals with increasing difficulty; **C2.** Tasks with pleasantly surprising positive feedback; **C3.** Multiple choices for advancing or retracing through the learning paths; **C4.** Frequent decision-making, to keep the learning process moving forward; **C5.** Enjoyable and fun learning activities.

*To fulfill the Relatedness Need.* Experiencing *relatedness* means feeling connected to peers, belonging to communities, and contributing to things 'greater' than oneself. A lower feeling of *relatedness* can reduce the students' motivation to interact with the system, which in turn may affect the satisfaction related to the other two basic innate needs, i.e., *autonomy* and *competence* [16]. *Relatedness* can be supported by various social interactions, such as tagging, rating, commenting and sharing with a learning



community; additionally, the visualisation of social status and reputation, via levels, badges and leaderboards, helps situating students within a meaningful community, with similar interests and preferences [10]. With *relatedness* feelings, we suggest that even if other rewards may be boring or meaningless for them, students may still retain motivation, if they enjoy the community. To summarise, a system could implement the following *relatedness gamification strategies*: **R1**. Opportunities to discover and join learning communities; **R2**. Connections of interest and goals between students and communities; **R3**. Various tools for interaction, collaboration, discussion and mutual assistance; **R4**. Visualisations of social status, reputation and contribution; **R5**. Supporting the display of appreciation to/of others (such as “like”).

## 4 Implementation

The proposed *motivational gamification strategies* were applied to implement Topolor 2 [11], a social adaptive e-learning system, which overhauls the previous version [9] with new *motivational gamification features*. This section maps each *motivational gamification strategy* onto concrete *motivational gamification features*, as described below (more details on Topolor 2 and its other features can be found in [11]). As explained, each strategy is supported by a wealth of different features.

*Structured and Chunked Goals with Increasing Challenges*. In Topolor 2, a *course* is composed of structured *topics*, so students have various “layers” of goals, with a learning path that can be accessed in different ways (**A1**). They have a *long-term goal* to complete the *course*, a *medium-term goal* to finish each *topic*, and a *short-term goal* to achieve each *objective*. Topics have reasonably short descriptions, although more resources of various sizes can be added to them (**C1**). They (normally) cannot jump goal layers, but they can decide which unlocked topic to learn next, or even access locked topics (**A5**), as many times as they wish (**C3**). Besides, a higher-level goal is usually more difficult and complicated (**C1**), so students can incrementally master new skills, and practice before they demonstrate mastery.

*Immediate and Positive Feedback with Guidance for the Next Step*. Topolor 2 provides clear, immediate and positive feedback for learning activities, to fulfill the need for *autonomy* and *competence*. For example, after finishing the pre-test of a course, Topolor 2 shows “congratulations” and encourages students to start the course (**A3** & **C2**) and offers thus the opportunity to join its learning community (**R1**). When a student shares a new post, such as an image or video, a reminder shows the number of the new post(s), similar to [twitter.com](https://twitter.com), so that the student can click on it to update the post list (**A3** & **C4**). Students need to continuously decide what to study next, and they can use various mechanisms to do that - e.g., learning path, filters of resources, etc. (**C4**). After submitting a test, Topolor 2 immediately shows the result and recommends the topics that the student may need to review (**A3** & **A4**).

*Visualisation of Social Status, Comparisons, and Learning Progress*. Topolor 2 supports various visualisations of individuals and communities for students to feel *competent* and

*related*. For example, the comparison of performance and contribution potentially encourage students to contribute more to the learning community (C5, R2 and R3), as seeing each other’s status may simulate imitation and competition. Students can also “like” an image, a video, etc. shared by others (C3, R3 and R4).

## 5 Evaluation

Two studies were conducted during two real-life university courses using Topolor 2 in two countries, with students of MSc and BSc levels. In the first course, “Dynamic Web-Based Systems”, 15 MSc students took part. They were learning the topic “Collaborative Filtering”, at the University of Warwick, UK, in 2013. The study included two time-controlled one-hour learning session (students sat in a classroom) and a non-time-controlled learning session (students accessed Topolor 2 at their preferred time and location). Ten completed the optional online survey, after the learning sessions.

A second course on “Management” was run in 2014 with 20 BSc students, learning the topic of “Control”, in the Sarajevo School of Science and Technology, Bosnia and Herzegovina. One online session took one and half hours. Then students further used Topolor 2 to revise the covered materials, for two weeks. After that, the students were asked to complete an optional online survey. Fifteen completed the survey.

**Table 1.** Statement and score of the *Perceived Motivation* questionnaire

#	Statement	$\mu$	$\sigma$	Category
1	I felt in control of my learning process.	0.60	0.50	<i>Autonomy</i> $\mu: 0.89$ $\sigma: 0.65$
2	I felt interested in using Topolor.	0.76	0.60	
3	I felt confident to use Topolor.	<b>1.12</b>	<b>0.78</b>	
4	I felt my learning experience was personalised.	1.08	0.70	
5	I felt having fun when using Topolor.	0.80	0.65	<i>Competence</i> $\mu: 0.99$ $\sigma: 0.57$
6	I felt I only needed a few steps to complete tasks.	0.64	0.57	
7	It was easy to understand why I received recommendations.	<b>1.36</b>	<b>0.49</b>	
8	It was easy to find the content I need.	<b>1.16</b>	<b>0.55</b>	
9	It was easy to share content with peers.	0.52	0.51	<i>Relatedness</i> $\mu: 0.77$ $\sigma: 0.86$
10	It was easy to access shared resources from peers.	0.76	0.60	
11	It was easy to tell peers what I like/dislike.	0.80	0.65	
12	It was easy to discuss with peers.	1.00	0.58	

The *Perceived Motivation Questionnaire* developed in [10] was adopted, targeting SDT's three basic innate needs: *autonomy*, *competence* and *relatedness*. It contained 12 statements on a five-point Likert scale (-2: strongly disagree ~ 2: strongly agree). Table 1 shows the statements and scores from the questionnaire. *Cronbach's  $\alpha$*  of the scores is 0.81, indicating a reliable internal consistency. The *means* ( $\mu$ ) range between 0.52 and 1.36, and the *standard deviations* ( $\sigma$ ) of the results are between 0.49 and 0.78. All the *means* are greater than 0 (the neutral response; overall  $\mu = 0.88$ ; overall  $\sigma = 0.60$ ), suggesting that the proposed *motivational gamification strategies* can provide a positive to high level of perceived motivation amongst students.

Among the statements, statement 7 obtained the highest score ( $\mu = 1.36$ ,  $\sigma = 0.49$ ). This is not surprising, as Topolor 2 explains each recommendation. For example, in the course structure view (learning path recommendation; not shown here due to lack of space), icons explain if a topic has been learnt, and if the student is eligible to learn it. Statement 8 received the second highest score ( $\mu = 1.16$ ,  $\sigma = 0.55$ ). This can be due to the new filtering tool implemented in Topolor 2. Statement 3 gained the third highest score ( $\mu = 1.12$ ). This further supports the *autonomy* goal.

Overall, as seen in Table 1, all motivational goals are achieved. The *competence* goal is supported by the features the most, and the *relatedness* goal the least (with statement 9 receiving the lowest score). The latter is possibly due to the fact that most students preferring to be "consumers" and not "producers" (a situation often observed in social media: about 80 % are "readers" or "consumers" and the rest "authors" or "producers"). Yet, the content sharing issue may need further investigation.

Additionally, the SUS (System Usability Scale) score for Topolor 2 is 76.1 out of 100 ( $\sigma = 12.36$ , *Cronbach's  $\alpha$*  = 0.98), suggesting a high usability of the system. This indicates the applicability of the proposed *motivational gamification strategies*.

## 6 Conclusion

To tackle the challenge of designing e-learning systems able to keep students highly motivated, we propose *motivational gamification strategies*, rooted in SDT. We provide means to concretely implement SDT-rooted motivational features in e-learning systems. We recommend using these strategies to guide the development and enhancement of general e-learning systems. We also suggest a method for exploring the impact of gamification on social adaptive e-learning – a measure of the perceived motivation amongst students.

## References

1. Abrami, P., et al.: Interaction in distance education and online learning: using evidence and theory to improve practice. *J. Comput. High. Educ.* **23**(2–3), 82–103 (2011)
2. Bandura, A., McClelland, D.C.: Social learning theory, pp. 1–46 (1977)
3. Banfield, J., Wilkerson, B.: Increasing student intrinsic motivation and self-efficacy through gamification pedagogy. *Sci. J.* **7**(4), 291–298 (2014)
4. Gillet, N., et al.: Intrinsic and extrinsic school motivation as a function of age: the mediating role of autonomy support. *Soc. Psychol. Educ.* **15**(1), 77–95 (2012)

5. Groh, F.: Gamification: state of the art definition and utilization. In: Proceedings of the 4th seminar on Research Trends in Media Informatics, pp. 39–46 (2012)
6. Kapp, K.M.: The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education. Wiley, New York (2012)
7. Ryan, R.M., Deci, E.L.: Overview of self-determination theory: an organismic dialectical perspective. In: Ryan, R.M., Deci, E.L. (eds.) Handbook of self-determination research, pp. 3–33. University of Rochester Press, Rochester (2002)
8. Saurabh, S., Sairam, A.S.: Professors – the new YouTube stars: education through Web 2.0 and social network. *Int. J. Web Based Communities* **9**(2), 212–232 (2013)
9. Shi, L., et al.: A social personalized adaptive e-learning environment: a case study in Topolor. *IADIS Int. J. WWW Internet* **11**(3), 13–34 (2013)
10. Shi, L.: Scaffolding for social personalised adaptive e-learning (doctoral dissertation). Retrieved from <http://wrap.warwick.ac.uk/67201> (2014)
11. Shi, L., Cristea, A.I.: Designing visualisation and interaction for social e-learning: a case study in Topolor 2. In: Rensing, C., de Freitas, S., Ley, T., Muñoz-Merino, P.J. (eds.) EC-TEL 2014. LNCS, vol. 8719, pp. 526–529. Springer, Heidelberg (2014)
12. Shi, L., Cristea, A.I.: Making it game-like: Topolor 2 and gamified social e-learning. In: The 22nd Conference on User Modeling, Adaptation and Personalization (UMAP 2014), pp. 61–64, Aalborg, Denmark (2014)
13. Simões, J., et al.: A social gamification framework for a K-6 learning platform. *Adv. Hum.-Comput. Interact.* **29**(2), 345–353 (2013)
14. Squire, K.: Video games in education. *Int. J. Intell. Games Simul.* **2**(1), 49–62 (2003)
15. Street, H.D.: Factors influencing a learner’s decision to drop-out or persist in higher education distance learning. *Online J. Distance Learn. Adm.* **13**, 4 (2010)
16. Vansteenkiste, M., et al.: Autonomy and relatedness among Chinese sojourners and applicants: conflictual or independent predictors of well-being and adjustment? *Motiv. Emot.* **30**(4), 273–282 (2006)
17. Werbach, K., Hunter, D.: For the Win: How Game Thinking can Revolutionize your Business. Wharton Digital Press, Philadelphia (2012)
18. Zimmerman, B.J.: A social cognitive view of self-regulated academic learning. *J. Educ. Psychol.* **81**(3), 329–339 (1989)

# Adaptive Training of the Metacognitive Skill of Knowledge Monitoring in Intelligent Tutoring Systems

Tiago Roberto Kautzmann, Talvany Carlotto, and Patrícia A. Jaques 

Programa Interdisciplinar de Pós-Graduação Em Computação Aplicada (PIPICA),  
Universidade Do Vale Do Rio Dos Sinos (UNISINOS),  
São Leopoldo, Brazil

tkautzmann@gmail.com, talvanynet@gmail.com, pjaques@unisinos.br

**Abstract.** This paper investigates the effects of training the metacognitive skill of knowledge monitoring when metacognitive instruction is adapted to the characteristics of students in intelligent tutoring systems. An animated pedagogical agent that trains knowledge monitoring was developed and integrated into a step-based tutoring system that helps students in solving algebraic equations. The training provided by the agent encourages learners to reflect on their knowledge and has its content and frequency of intervention adapted to the characteristics of the student. Related work has not adapted the metacognitive instruction to the characteristics of the student, nor has it aimed at investigating the effects of knowledge monitoring training specifically. Results of a classroom study suggest that students who received metacognitive training improved their knowledge monitoring skill and performed better on tests.

**Keywords:** Knowledge monitoring skill · Metacognition · Pedagogical agent · Intelligent tutoring systems

## 1 Introduction

The metacognitive skill of knowledge monitoring is the ability people have of identifying what they know and what they do not know about a given subject. This ability is fundamental to the acquisition of other metacognitive skills, and it influences academic achievement [1]. Students who are able to accurately identify their knowledge in a subject are prone to work harder in the development of their deficit areas [2], to seek help when necessary [3] and to study more strategically [1].

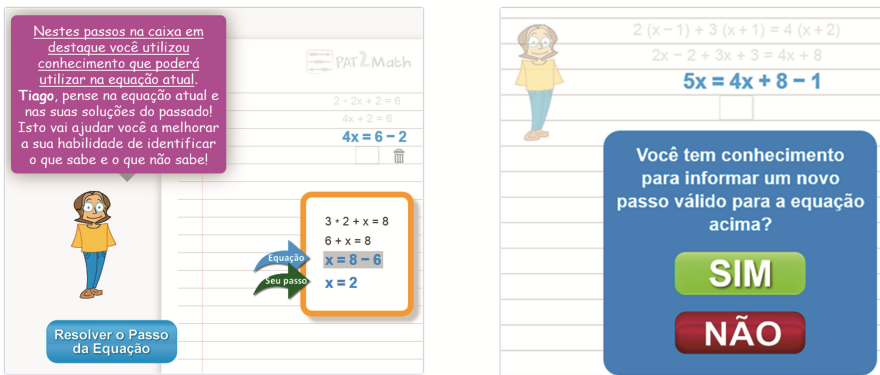
Several studies have investigated how learning systems can improve students' metacognitive skills. Although some of these works have specifically encouraged students to reflect on their knowledge at a given moment [4–8], they have not studied the knowledge monitoring skill exclusively. Therefore, the isolated effects of computer-aided instruction aimed at encouraging students to reflect on their knowledge are still unknown. Additionally, the metacognitive training has not been adapted to the characteristics of the learners in these works.

The present work aims at investigating the specific effects of knowledge monitoring training in learning systems that adapt system content and frequency of intervention to

the students' metacognitive skill level, task performance, and problem solving history. An animated pedagogical agent that trains the metacognitive skill of knowledge monitoring in an algebraic step-based ITS was developed to achieve that goal. The training provided by the agent encourages learners to reflect on their knowledge during problem solving processes.

## 2 Knowledge Monitoring Training Agent

The animated pedagogical agent is represented by a feminine animated character that can move around the screen and present idle behavior such as breathing and blinking (Fig. 1). It trains knowledge monitoring through speech balloons and text messages that encourage students to reflect on their knowledge. The following instructional strategies were adopted: (1) encourage the student to identify what the problem asks; (2) encourage the student to dedicate some time to reflect on their knowledge before trying to solve the next step; and (3) encourage the student to reflect on similar, previously solved problems. These strategies and their importance during problem solving are discussed in [4, 10–12].



**Fig. 1.** (a) the agent delivering a prompt that shows a similar, previously solved step, (b) the agent asking the users to assess their knowledge to solve the current step

The adaptive training is an important characteristic of the agent. Both training content and the agent's frequency of intervention are adapted to the following student info: (1) their current level of knowledge monitoring; (2) their domain knowledge; and (3) their problem solving history. The agent may be integrated into ITSs that: (1) provide step-by-step assistance during problem solving; (2) keep a problem solving record; and (3) are able to identify the knowledge applied by learners in each step and the knowledge that might be applied in next steps (such as model tracing tutors).

This study used an instrument called Knowledge Monitoring Assessment (KMA), which measures learners' ability of monitoring their own knowledge. The KMA compares the student's assessment of their own knowledge to solve a problem with their actual performance on the same problem, generating a total of four possible

scores. In one of the scores, for instance, the student assessed she knows how to solve the problem, and she did solve it correctly; in another score the learner assessed she did not know how to solve the problem, but she did solve it correctly; and so on. The agent uses these scores to generate the KMA index, which measures the discrepancy between the knowledge assessed by the students and the knowledge demonstrated by them. The Hamman Coefficient formula is used to calculate the KMA index, as suggested by [1].

The KMA index is a real number between  $-1$  and  $+1$ , in which  $+1$  indicates precision in knowledge monitoring. It was classified into two categories: satisfactory and unsatisfactory. Values greater than  $+0.5$  represent satisfactory KMA indices.

The agent has two operating mechanisms: (1) the outer loop and (2) the inner flow.

The **outer loop** is responsible for activating the inner flow (which is responsible for the actual knowledge monitoring training), and it is always executed before the student makes an attempt at solving a problem step. Two strategies are used to decide whether the inner flow should be activated or not. The first strategy makes decisions based on student's metacognitive level, and it is used when the current KMA index of the learner is unsatisfactory, or when he or she has started the training too recently. In this strategy, the greater the KMA index, the smaller the probability of activation of the inner flow, i.e. the frequency of activation of the inner flow is adapted to the metacognitive level of the student. The second strategy makes decisions based on the student's knowledge, and it is used while the student keeps a satisfactory KMA index. This mechanism uses the information from the ITS' student model, more specifically the probability that the learner masters the knowledge units that are necessary to solve the next step, to generate a relevance index for the next step, which is a real number between 0 and  $+1$ . Relevance indices close to 1 suggest that the student neither completely know neither completely do not know the solution of the next step. The greater the relevance index, the bigger the probability of activation of the inner flow.

The **inner flow** is responsible for knowledge monitoring training. Initially, if the KMA index is unsatisfactory, the mechanism selects a prompt that encourages students to reflect if they have the knowledge required to solve the current step of the problem and waits for the solution. If learners are very reactive, i.e., if they go to the next step very quickly, a message is delivered, indicating inadequate behavior. Before entering a new step, students must assess whether they have the knowledge to solve the step by choosing "YES" or "NO". Next, the mechanism compares learners' self-assessment with their performance on the current step and updates the student's KMA index. Additionally, the mechanism may choose to deliver a self-explanation activity in which students fill a form describing the reasons for the assessment. Before each new step, the pedagogical agent can also inform the learners of their current level of knowledge monitoring.

Prompts are text messages that are activated by the inner flow and encourage students to reflect on their knowledge. The prompts were grouped into four levels. The first level encourages learners to reflect on their knowledge using the description of the problem. The second level makes students reflect on knowledge they have already demonstrated to master. The third level makes them reflect on solutions that they previously entered, which required knowledge that can also be applied to the current step. The fourth level

shows a similar, previously solved step. Prompt levels are selected according to students' KMA index. The smaller the current KMA index, the greater the prompt level that is selected (i.e., closer to level 4).

### 3 Evaluation Study

The goal of the evaluation study is to verify the effects of explicitly instructing students to reflect on their knowledge when the content and the frequency of the metacognitive intervention are adapted to the learners' current level of knowledge monitoring, their task performance and their problem solving history. We aim at answering the following research questions: (1) Does the adaptive training improve students' knowledge monitoring skill? (2) Does the adaptive training improve learning? (3) Is there a correlation between students' level of knowledge monitoring and their performance on tests when they receive adaptive metacognitive training?

The hypothesis of the authors of this study for the first research question is that the adaptive training effectively improves the knowledge monitoring skill, because it encourages learners to reflect on their knowledge, making them act less reactively, and also because it makes them aware of the importance of this metacognitive skill in their studies. Additionally, both frequency and content of the intervention are adapted to the characteristics of students. The benefits of this adaptive characteristic of the metacognitive training have already been verified in classroom [9]. In the context of computer-aided instruction, existent research have not investigated the isolated effects of the students' reflection on their own knowledge. They also have not adapted the instruction to the learners' current level of knowledge monitoring. The hypothesis for the second research question is that the metacognitive training does improve learning, since the actions of the agent lead students to be less reactive. Finally, the hypothesis for the third research question is that there exists a positive correlation between learning and knowledge monitoring skill in students receiving the metacognitive training. This correlation has already been found in classroom [1], but not in ITSs.

An experimental evaluation with a control group was carried out with classes of seventh grade students in four private schools in the south of Brazil. One hundred seven students (ranging from 12 to 14 years old) participated in the study and were randomly assigned to either the experimental group or the control group. The animated pedagogical agent was integrated into PAT2Math (<http://pat2math.unisinus.br>), a step-based ITS that provides step-by-step assistance for students in the process of solving linear equations (Fig. 1).

The experiment was composed of a total of six to seven sessions. In the first session, students received training in PAT2Math without the agent. In the second session, students completed a pretest and a metacognitive self-assessment instrument. In the following sessions, students solved equations using two different versions of PAT2Math with the agent. In the experimental condition, the animated pedagogical agent did deliver metacognitive instruction. In the control condition, the agent was modified to not deliver metacognitive actions; it only provided hints related to the domain, which were also delivered by the agent of the experimental group. In the last session, students completed



a posttest and a metacognitive self-assessment instrument. Each session lasted for 50 min and all sessions had an interval of one week between each other.

From the 107 students who participated in the experiment, we could only consider the data of the 63 students who returned the consent form signed, and completed both pretest and posttest. Thirty-four students were from the control group (44 % of boys and 56 % of girls), and 29 students were from the experimental group (55 % of boys and 45 % of girls). In the pretest and posttest sessions, students' performance on tests assessed their learning and students' metacognitive indices (KMA) assessed their knowledge monitoring skills.

## 4 Results

An independent t-test ( $\alpha = .05$ ) comparing the means of KMA index obtained in the posttest by the experimental group ( $\mu = .800, \sigma = .251$ ) and by the control group ( $\mu = .605, \sigma = .413$ ) found a significantly higher mean ( $t(61) = 2.214, p = .015$ ) in the experimental group. A second independent t-test compared KMA index gains (i.e., difference between KMA index after and before the experiment) of the groups. A statistical result marginal to the significant level ( $t(61) = 1.588, p = .059$ ) was found, indicating that KMA index gains in the experimental group ( $\mu = .231, \sigma = .373$ ) were higher than the gains in the control group ( $\mu = .061, \sigma = .463$ ). A dependent t-test comparing KMA indices between the pretest ( $\mu = .544, \sigma = .325$ ) and the posttest ( $\mu = .605, \sigma = .413$ ) did not find any statistically significant differences ( $t(33) = .766, p = .225$ ) in the control group. However, a significantly higher mean ( $t(28) = 3.333, p < .01$ ) was found in the experimental group ( $\mu_{pre} = .569, \sigma_{pre} = .381; \mu_{pos} = .800, \sigma_{pos} = .251$ ).

An independent t-test ( $\alpha = .05$ ) compared the grades of the students in both conditions in the posttest. A higher mean ( $t(61) = 2.327, p = 0.012$ ) was found in the grades of the experimental group ( $\mu = 8.621, \sigma = 1.741$ ) when compared to the control group ( $\mu = 7.441, \sigma = 2.205$ ). Additionally, another independent t-test compared gain score (i.e., difference between post and pre-tests) between the groups. The gain mean in the experimental group ( $\mu = 2.414, \sigma = 2.147$ ) was higher than the gain mean in the control group ( $\mu = 1.941, \sigma = 1.825$ ), but this difference was not statistically significant ( $t(61) = .945, p = .174$ ). Results of Pearson's correlation coefficient analysis found a statistically significant ( $\alpha = .05$ ) positive correlation between the KMA index and the students' posttest scores in the experimental condition.

## 5 Conclusions

This work investigated the specific effects of adaptive training of the metacognitive skill of knowledge monitoring in an ITS. An animated pedagogical agent that trains knowledge monitoring in step-based ITSs was implemented. The agent encourages students to reflect on their knowledge during problem solving and adapts content and frequency of the interventions to students' current level of knowledge monitoring, their knowledge in the domain and their problem solving history in the ITS.

In the evaluation, the animated pedagogical agent was integrated into PAT2Math, a step-based ITS that helps students in solving linear equations. The results of the

statistical tests presented evidence that supports the hypothesis that the training provided by the agent improves the students' skill of knowledge monitoring. The study also found evidence that supports the hypothesis that the metacognitive instruction of the agent improves performance on tests, although the results of the statistical tests were not so statistically solid. This study also found evidence of a strong positive correlation between knowledge monitoring skill and learning by students that received training from the agent. This evidence supports the results found in classroom, as described in [1]. However, these results had not been verified in computer-aided instruction, and more specifically in ITS, yet. We believe that the experiment results would be more significant if students could have an extended use of the ITS, given that one of the fundamental principles to achieve success in metacognitive instruction is that the training of the metacognitive skills must be prolonged [13]. Students only had three or four sessions of ITS use due to the schools scheduling restrictions.

**Acknowledgments.** The present work has received financial support of the following Brazilian research funding agencies: CNPq and FAPERGS.

## References

1. Tobias, S., Everson, H.T.: Knowing what you know and what you don't: further research on metacognitive knowledge monitoring. In: College Board Research Report, pp. 1–25 (2002)
2. Fogarty, R.: How to Teach for Metacognitive Reflection. Pearson, Glenview (1994)
3. Stavrianopoulos, K.: Adolescent's metacognitive knowledge monitoring and academic help seeking. *Coll. Student J.* **41**(2), 444–453 (2007)
4. Gama, C.A.: Integrating Metacognition Instruction in Interactive Learning Environments. Thesis (PhD), University of Sussex, Brighton (2004)
5. Aleven, V., McLaren, B., Koedinger, K., Rool, I.: Toward metacognitive tutoring: a model of help-seeking with a cognitive tutor. *IJAIED* **16**, 101–130 (2006)
6. Park Woolf, B., et al.: A general platform for inquiry learning. In: Cerri, S.A., Gouardères, G., Paraguaçu, F. (eds.) ITS 2002. LNCS, vol. 2363, pp. 681–697. Springer, Heidelberg (2002)
7. Kramarski, B., Michalsky, T.: Student and teacher perspectives on IMPROVE self-regulation prompts in web-based learning. In: Azevedo, R., Aleven, V. (eds.) *International Handbook of Metacognition and Learning Technologies*. Springer International Handbooks of Education, vol. 28, pp. 35–51. Springer, Heidelberg (2002)
8. Azevedo, R., Feyzi-Behnagh, R.F., Duffy, M., Harley, J.M., Trevors, G.: Metacognition and self-regulated learning in student-centered learning environments. In: *Theoretical foundations of student-center learning environments*, Routledge, vol. 2, pp. 171–197 (2012)
9. Azevedo, R., Hadwin, A.F.: Scaffolding self-regulated learning and metacognition. *Instr. Sci.* **33**(5–6), 367–379 (2005)
10. Fogarty, R.: How to Teach for Metacognitive Reflection. Pearson, Glenview (1994)
11. Verschaffel, L.: Realistic mathematical modelling and problem solving in the upper elementary school: Analysis and improvement. In: Hamers, J.H.M., van Luit, J.E.H., Csapo, B. (eds.) *Teaching and Learning Thinking Skills*, pp. 215–240. Swets & Zeitlinger (1999)
12. Polya, G.: How to Solve It. Princeton University Press, Garden City (1957)
13. Veenman, M.J., Van Hout-Wolters, B.H.M., Afflerbach, P.: Metacognition and learning: conceptual and methodological considerations. *Metacognition Learn.* **1**(1), 3–14 (2006)

# Persuading an Open Learner Model in the Context of a University Course: An Exploratory Study

Blandine Ginon<sup>1</sup>(✉), Clelia Boscolo<sup>2</sup>, Matthew D. Johnson<sup>1</sup>,  
and Susan Bull<sup>3</sup>

<sup>1</sup> School of Engineering, University of Birmingham, Birmingham, UK  
b.ginon.l@bham.ac.uk

<sup>2</sup> Department of Modern Languages, University of Birmingham,  
Birmingham, UK

<sup>3</sup> Institute of Education, University College London, London, UK

**Abstract.** The LEA's Box open learner model (OLM) allows learners to try to persuade the system to make changes to their learner model by challenging evidence or providing justifications. This aims to help make the OLM more accurate, and provides a means for learners to satisfy themselves that the model does indeed reflect their current state of learning. We report an exploratory study with 15 university students, with learner model data coming from quizzes in a Learning Management System. Students generally claimed to understand the approach of learner model persuasion, how it is useful, how it relates to their learning, and identified cases when they could use persuasion.

**Keywords:** Open learner model · Learner model persuasion

## 1 Introduction

Open learner models (OLMs) are learner models that can be accessed in a user-understandable form [3]. Some are interactively maintained by both system and student, helping increase the accuracy of the model, supporting reflection, facilitating planning and self-monitoring, and affording the learner a greater level of control over the learner model data [2, 10]. Those that allow users to directly edit, and therefore fully control the contents of their OLM (e.g. [4, 8, 12]) may be particularly appropriate when learners are known to be accurate, and are also confident in self-assessment. It has been suggested that learners may feel more confident if the model changes are *validated* by another stakeholder [12] such as a teacher or the system. OLMs can also be updated through the student contribution of additional information (e.g. [6, 10, 17]), an evidence based approach [18], enabling the OLM to benefit from user-given data, but without handing full control to the learner as in editable models.

In contrast to the above, negotiated learner models allow learners to challenge learner model data, with separate representations retained if the learner and system cannot agree on a representation [1, 9, 11]. Persuadable OLMs also allow learners to request and justify changes to their model, e.g. by answering additional questions

[7, 12, 14, 15] or selecting from teacher-defined reasons [5]. If the system is convinced, the model will be updated. However, in this case the system retains control of the learner model if the student does not successfully justify their reasons for changing representations. Both negotiation and persuasion aim to help overcome possible learner reticence of not having validation for the model content in OLMs that they can edit or add information to, without challenge (as suggested in [12]), whilst ensuring some responsibility for OLM content is retained by the learner – an important aspect that OLMs aim to support [3, 10].

In our context, all OLM evidence originates from external data sources. Such approaches have also been investigated with other OLMs (e.g. [6, 13, 16]), since today’s learners now use a range of learning applications. However, a potential limitation of such situations is that the data from other sources may be of different granularity, may not be equally representative of student learning, or may simply not be regarded by students as equally valid. Therefore, adding the facility to allow users to try to persuade the learner model to update any data that they believe does not adequately represent their skills, aims to help overcome these limitations. Students may offer information that can help increase the accuracy of their learner model in this context, while retaining the system control offered by persuasion approaches, and also the validation as considered important by some students [12]. Our initial findings with a persuadable OLM are likely to apply also in some negotiated learner modelling contexts.

## 2 The LEA’s Box Persuadable Learner Model

The LEA’s Box OLM offers ten visualisations [5], both simple (e.g. skill meter, radar plot) and more complex (e.g. network), see Fig. 1, and the OLM can be constructed from a range of activities and multiple data sources (based on [6]). As in some other OLMs (e.g. 10]), the persuasion feature allows learners to view evidence underlying their learner model. In addition, it allows users the opportunity to try to persuade the system to make changes if disagreement occurs, e.g. by challenging evidence or providing justifications for their own assessment of their skills. Table 1 (extended from [5]) details the moves available to the system and learner.

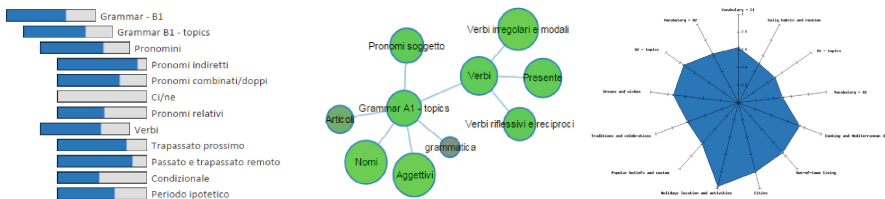
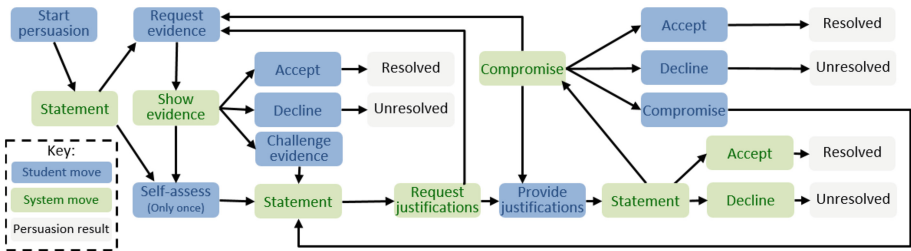


Fig. 1. Example LEA’s Box visualisations: skill meters; network; radar plot.

**Table 1.** Persuasion moves for each stakeholder.

	Student	System
Accept/agree	Agree with the system’s evidence; accept a compromise	Agree with the student’s justifications; accept a compromise
Decline	Decline system proposed compromise	Decline (e.g. too recent)
Compromise	Propose a compromise between the system’s the student’s self-assessment	Propose a compromise between current level and self-assessment
Request evidence or justifications	Request evidence for current level	Request justifications for a self-assessment
Provide evidence or justifications	Provide justifications (e.g. homework, further study, external factors)	Provide evidence (e.g. learner model evidence)
Self-assess	Proposition of a new OLM state	×
Challenge evidence	Disagreement with item of evidence	×
Statement	×	Statement of fact about the OLM



**Fig. 2.** Workflow for persuasion.

The first step of the persuasion workflow (Fig. 2) displays the student’s current level for a competency (statement). The student may then request evidence or self-assess, to try to change the value. Requesting evidence is available throughout persuasion, and details how the current competency is calculated, taking into account all evidence associated with the competency and its sub-competencies (Fig. 3). Evidence may, for example, be a score in a quiz, a teacher assessment, or the result of a past persuasion. The modelling process gives more recent evidence a higher weight. Following a student self-assessment, the system requires justifications to validate the increase or decrease to the value in the learner model. Using teacher defined parameters [5], the system accepts or declines the proposed change, or may propose a compromise. If the student accepts a compromise or the system accepts the student’s proposition, the model is updated with an additional piece of evidence stating the new value. In that case, older evidence no

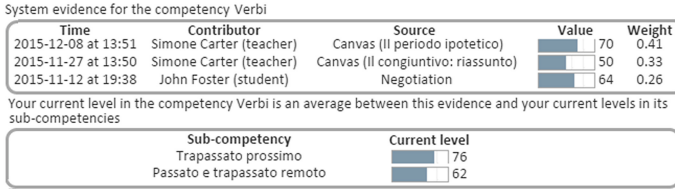


Fig. 3. Example display of system evidence.

longer contributes to the modelling process, but remains available for reference. If a self-assessment or a compromise is declined, the model is not updated as the system, parameterised by teacher, ultimately retains the control, as in other persuadable learner models [7, 14, 15].

### 3 User Perceptions of the LEA’s Box Persuadable OLM

The LEA’s Box OLM was used by 15 volunteers studying Italian at the University of Birmingham. The exploratory study investigated whether students claim to understand OLM persuasion and find it useful, and their motivations for why it might be used in their learning. OLM evidence came from short answer quizzes imported daily from the course Learning Management System (LMS). The quizzes take about 30 min to complete, can be repeated, and cover 133 teacher defined grammar and vocabulary topics.

At the start of the course, students were given a demonstration of the OLM and its persuasion facility using a test account with sample data. The OLM was available for two months (the first week and last two weeks of which were during term time). All OLM usage was logged. At the end of the period, participants completed a 5-point Likert scale questionnaire, and individual semi-structured interviews took place with 5 volunteers during an optional lab session. The interviews lasted about 10 min, and were audio recorded and transcribed. They took place in front of the student’s OLM, and focused on participants’ perceptions/attitudes towards OLM persuasion, including whether it was used, why it might be used and why it might not be used.

Most interaction occurred during the term (start and end of the period). 7 participants used persuasion, in 12 OLM discussions: 3 were resolved (i.e. with system-student agreement and a model update); one was discontinued as it was too soon after a previously resolved persuasion; 8 terminated after viewing evidence. All persuasion attempts were self-assessments higher than the value in the OLM. 11 participants returned questionnaires (Fig. 4): 3 used persuasion, 8 did not. Those who attempted persuasion indicated that they disagreed with their OLM, whilst those who did not try to persuade, indicated agreement. None of the latter claimed not to understand persuasion. One indicated that they refrained from persuasion because it was not summatively assessed. All who used persuasion wished to make the OLM more accurate, and wished to explain their viewpoint and understand the evidence behind the model. The 5 interview transcripts showed reasons for using/not using persuasion as relating to information, time, precision and attitudes (Table 2 states the

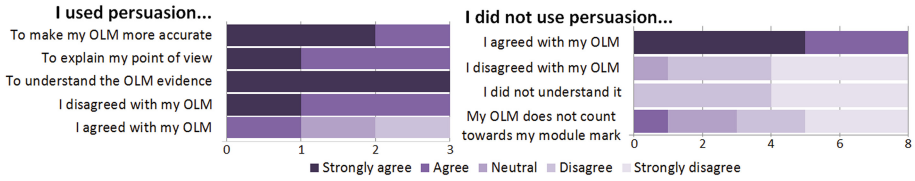


Fig. 4. Questionnaire responses: reasons to use/not use the persuasion feature.

Table 2. Themes from interviews about expected use of persuasion feature.

Persuasion	Would use	Would not use
Informational	<p>[5] course is large. Not all competencies have info</p> <p>[3] only completed most relevant part of quiz</p> <p>[2] wish to restore the model to a previous state</p> <p>[2] quiz content broad. Only part completed quiz</p>	<p>[2] more evidence is required first.</p> <p>[2] difficult to add self-assessment values</p>
Temporal	<p>[3] quizzes take a long time to complete.</p> <p>[3] student ran out of time, quiz submitted early</p> <p>[1] wanted immediate feedback, quiz not complete</p>	<p>[2] takes time to complete persuasion.</p> <p>[2] not exam period</p>
Inference precision and level of interaction	<p>[3] answer incorrectly marked e.g. part of sentence not typed, but still correct</p> <p>[3] no half marks. e.g. case sensitive responses</p> <p>[2] setup error causes incorrect marking</p> <p>[2] quiz platform interaction problems</p> <p>[1] right answers placed in wrong boxes</p>	<p>[4] do not feel have done enough quizzes yet to use persuasion effectively</p> <p>[3] already accurate</p>
Attitudes/strategies	<p>[1] learning strategy leads to lower level showing in the OLM, e.g. use of trial and error</p>	<p>[2] not technology confident</p> <p>[1] no summative mark</p>

[number of participants] who mentioned each theme). For OLM accuracy, participants indicated persuasion might be needed where they have short term goals (e.g. part completion of quizzes), because of limitations with the LMS (e.g. incorrect marking, multiple right answers) or because of more transient constraints (e.g. out of time to interact with the LMS, early quiz submission). Participants also indicated OLM persuasion may not be a priority because the model was already perceived as accurate, not

enough course content had been covered, or it was not at the point of the course where it was of most use (“during the summer exam period”). Two participants added that they wished use persuasion after they had completed more quizzes.

## 4 Discussion and Conclusions

OLMs are designed to represent learners’ current skills, knowledge, competencies, etc. Usually they are assumed to be as accurate as is necessary for the purpose of personalising teaching. In this study, as also described in other research (e.g. [16]), the activities providing data were from a LMS. In our case, the data was transferred to the OLM each day. This meant that there was more scope for the OLM to be outdated, and perhaps, more reason for students to try to persuade the system to change values. Against this, however, is the fact that OLMs are typically updated dynamically as students interact with a learning system, and so students may have regarded the delay as too cumbersome to engage fully. Our aim, therefore, was to explore students’ reasons to choose to use or not use an OLM persuasion feature in this context.

Interaction logs, questionnaires and interviews indicated that learners could see how persuasion related to their learning, and many participants said that they agreed with their model, so there was no reason to try to update it. Some stated that it was perhaps the wrong time in their learning to use persuasion, potentially because of the size of the course, time taken to complete (or partially complete) quizzes, or because they may wish to wait until upcoming summative assessment before more intense engagement. Of those who claimed to have started model persuasion, each had an interest in seeing evidence behind their OLM. This may suggest that a core foundation to OLM persuasion is understanding the evidence’s origin and context, in order for the learner to think about the differences and similarities between this and their perceptions of OLM accuracy, in line with other calls to show learner model evidence [10]. Participants showed awareness of some limitations of the LMS quiz engine, such as stringent scoring, human error, or using it with their own learning strategies (e.g. working on only small parts of course content), leading to the OLM underestimating competency. This presents an interesting case for keeping the model accurate, and for OLM persuasion, away from the more usual use of OLMs in intelligent tutoring systems where dynamic modelling is at the core of the system.

Some of our findings may generalise to other contexts: university students appear to understand how OLM persuasion applies to their learning, when it may be useful, and are willing to challenge evidence if they disagree, explaining their point of view. Such persuasion allows them opportunities to try to influence the model data, and could give them more control over their learning in, for example, an ITS where teaching is personalised according to the learner model. This control may be further increased in contexts learner model negotiation techniques are used.

**Acknowledgments.** This project is supported by the European Commission (EC) under the Information Society Technology priority FP7 for R&D, contract 619762 LEA’s Box. This document does not represent the opinion of the EC and the EC is not responsible for any use that might be made of its contents.



## References

1. Bull, S., Pain, H.: Did I say what I think I said, and do you agree with me? Inspecting and questioning the student model. In: Greer, J. (ed) Proceedings of World Conference on Artificial Intelligence and Education. AACE, Charlottesville, VA, pp. 501–508 (1995)
2. Bull, S.: Negotiated learner modelling to maintain today's learner models. RPTTEL (in press)
3. Bull, S., Kay, J.: Open learner models as drivers for metacognitive processes. In: Azevedo, R., Aleven, V. (eds.) International Handbook of Metacognition and Learning Technologies, pp. 349–365. Springer, New York (2013)
4. Bull, S., Dong, X., Britland, M., Guo, Yu.: Can students edit their learner model appropriately? In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 674–676. Springer, Heidelberg (2008)
5. Bull, S., Ginon, B., Boscolo, C., Johnson, M.: Introduction of learning visualisations and metacognitive support in a persuadable open learner model. In: LAK 2016 (in press)
6. Bull, S., Johnson, M., Alotaibi, M., Byrne, W., Cierniak, G.: Visualising multiple data sources in an independent open learner model. In: Lane, H., Yacef, K., Mostow, J., Pavlik, P. (eds.) AIED 2013. LNCS, vol. 7926, pp. 199–208. Springer, Heidelberg (2013)
7. Bull, S., Mabbott, A., Abu-Issa, A.: UMPTEEN: named and anonymous learner model access for instructors and peers. IJAIED **17**(3), 227–253 (2007)
8. Czarkowski, M., Kay, J., Potts, S.: Web framework for scrutable adaptation. In: Kay, J., Lum, A., Zapata-Rivera, D. (eds.) Learner Modelling for Reflection to Support Learner Control, Metacognition and Improved Communication, AIED Workshop, pp. 11–18 (2005)
9. Dimitrova, V.: StyLE-OLM: interactive open learner modelling. IJAIED **13**(1), 35–78 (2003)
10. Kay, J.: Learner know thyself: student models to give learner control and responsibility. In: Halim, Z., Ottomann, T., Razak, Z. (eds.) ICCE 1997. AACE, pp. 17–24 (1997)
11. Kerly, A., Bull, S.: Children's interactions with inspectable and negotiated learner models. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 132–141. Springer, Heidelberg (2008)
12. Mabbott, A., Bull, S.: Student preferences for editing, persuading, and negotiating the open learner model. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 481–490. Springer, Heidelberg (2006)
13. Morales, R., Van Labeke, N., Brna, P., Chan, M.E.: Open learner modelling as the keystone of the next generation of adaptive learning environments. In: Mourlas, C., Germanakos, P. (eds.) Intelligent User Interfaces. ICI Global, London (2009)
14. Tchétagni, J., Nkambou, R., Bourdeau, J.: Explicit reflection in prolog tutor. IJAIED **17**(2), 169–215 (2007)
15. Thomson, D., Mitrovic, A.: Preliminary evaluation of a negotiable student model in a constraint-based ITS. RPTTEL **5**(1), 19–33 (2010)
16. Tongchai, N.: Impact of self-regulation and open learner model on learning achievement in blended learning environment. IJIET **6**(5), 343 (in press)
17. Van Labeke, N., Brna, P., Morales, R.: Opening up the interpretation process in an open learner model. IJAIED **17**(3), 305–338 (2007)
18. Zapata-Rivera, D., Hansen, E., Shute, V.J., Underwood, J.S., Bauer, M.: Evidence-based approach to interacting with open student models. IJAIED **17**(3), 273–303 (2007)

# Blinded by Science?: Exploring Affective Meaning in Students' Own Words

Sarah E. Schultz<sup>1</sup>(✉), Naomi Wixon<sup>1</sup>, Danielle Allesio<sup>2</sup>,  
Kasia Muldner<sup>3</sup>, Winslow Burseson<sup>4</sup>, Beverly Woolf<sup>2</sup>,  
and Ivon Arroyo<sup>1</sup>

<sup>1</sup> Worcester Polytechnic Institute, Worcester, MA, USA

{seschultz,mwixon,iarroyo}@wpi.edu

<sup>2</sup> University of Massachusetts, Amherst, MA, USA

allessio@educ.umass.edu, bev@cs.umass.edu

<sup>3</sup> Carleton University, Ottawa, ON, Canada

kasia.muldner@carleton.ca

<sup>4</sup> New York University, New York, NY, USA

wb50@nyu.edu

**Abstract.** This work addresses students' open responses on causal attributions of their self-reported affective states. We use qualitative thematic data analysis techniques to develop a coding scheme by identifying common themes in students' self-reported attributions. We then applied this scheme to a larger set of student reports. Analysis shows that students' reasons for reporting a certain affect do not always align with researchers' expectations. In particular, we discovered that a sizable group of students externalize their affect, attributing perceived difficulty of the problem and their own negativity as lying outside of themselves.

## 1 Introduction

When an adaptive tutor, MathSpring, asked a student to explain the reason for her self-reported affect of high interest, she typed in “because i think i will learn a lot of new information in this website about math. [*sic*]” This student was explaining that her interest stems from anticipating progress in her learning. This type of open-response data, however, is not typically collected in studies on student affect. Instead, as described in [4], existing methods in the Intelligent Tutoring Systems (ITS) community generally focus on classifying affect categorically, such as “interest,” “frustration,” “excitement,” and so on. While this information is valuable, the way that students define these terms may not be the same as what researchers believe the terms mean [3], so it will be valuable to find out *why* students report feeling a certain way.

To be scalable, the classification of student open-ended responses would need to be done automatically by relying on natural language processing (NLP) techniques. However, before investing in the design of such technologies, it is necessary to determine how much of an advantage examining these reports truly offers.

In this work, we examine data from two populations of students interacting with the MathSpring system (formerly Wayang Outpost). The system periodically asked the

students to (1) self-report on their affect using a Likert scale and (2) explain why they were feeling that way using an open-ended response dialog box (Fig. 1). We use a qualitative approach to create and apply a coding scheme to students' self-reported explanations and then use data mining techniques to explore what type of student responses different prompts and reports of affect were likely to elicit. Additionally, we seek to determine whether students' reports of reasons for feeling a certain way align with our expectations, as researchers, of what types of attributions occur with different types of affect.

## 2 Methods

This research was conducted using data from two studies using the MathSpring ITS (formerly Wayang Outpost) [1]. The studies were run in 2011 with 7th and 8th grade students ( $N = 123$ ), and 2015 with 7th grade students ( $N = 209$ ), in Massachusetts and California respectively.

To obtain in situ information about student affect, MathSpring prompted reports every five minutes or every eight problems, whichever came first without interrupting a problem. Students were asked to report on a target emotion (e.g., interest, excitement) via a 5-point Likert scale, and to explain why they were feeling that way (Fig. 1).

Please tell us how you are feeling.  
Based on the last few problems tell us about your level of  
Confidence in solving math problems

Not at all Confident  
 A little Confident  
 Somewhat Confident  
 Quite a bit Confident  
 Extremely Confident

Why is that?

OK

**Fig. 1.** Student self-report of affect. Open prompt (bottom) asked students to explain why they felt that way.

We use two types of qualitative thematic data analysis; open coding (phase 1), where coders independently code student report data with little direction, and axial coding [2], where core categories are developed based on coders' open coding schemes (phase 2). Our third phase consists of validating those sub-categories (which we use as our tags) through inter-rater reliability as measured with Cohen's kappa.

**Phase 1: Open Coding.** Our first step was "open coding," [2] wherein coders parse and reflect on data with the goal of naming and categorizing phenomena that occur within. Here, a set of 450 randomly selected open responses was gathered from a dataset collected in 2011 and given to the five coders (the first four authors, and one

additional coder). The coders were told how the data was collected, but were not given a coding scheme apart from the directive to independently arrive at a set of approximately 10 categories that would encompass approximately 70 % of the responses. Coders were instructed that they could tag a response with multiple tags if they felt they were applicable.

**Phase 2: Axial Coding.** Once all five coders created their schemes and tagged the data, we entered the “axial coding” phase [2], in which the 2<sup>nd</sup>, 3<sup>rd</sup>, and 4<sup>th</sup> authors reviewed the independently devised coding schemes to determine simple commonalities among them. Where similar categories were created by most coders, they were merged into a single category. Ten final categories were determined, as follows, with examples of responses that were tagged with each attribution and abbreviations in parentheses:

- IDK (idk) – doesn’t understand why they feel the way they feel (thus “IDK” for “I don’t know”) or doesn’t want to tell us why. (e.g., “????????????????????”, “meh”).
- Boring (bor) – describes something as boring (e.g., “math is boring”).
- Easy (easy) – says the material is easy (e.g., “too easy,” “its simple” [sic]).
- Hard (hard) – says the material is hard or difficult (e.g., “it is a little confusing and hard”, “it gives me a good challenge”).
- Internal (int) – attributes their feelings to internal causes (e.g., “i am smart” [sic]).
- External (ext) – attributes their feelings to external causes (e.g., “It is kind of fun”).
- Positive (pos) – the valence is positive (e.g., “I like this program”).
- Negative (neg) – the valence is negative (e.g., “I hate math”).
- Supportive (sup) – feels supported (e.g., “It is fun but it also helps me learn a lot.”).
- Unsupportive (unsup) – does not feel supported (e.g., “is not helpful”).

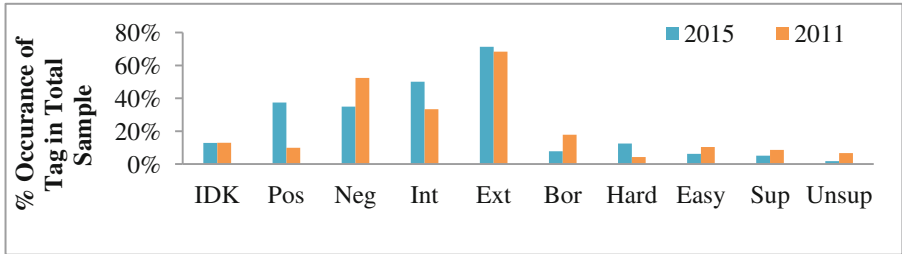
**Phase 3: Application and Validation of Tags.** The coding scheme was applied by the first four authors to the 2015 dataset, coding each response. After tagging student responses, inter-rater reliability was determined by Cohen’s kappa. The highest agreement between any two authors is displayed in Table 1. Overall, the second author had highest agreement with other authors, so their tags were used for analyses.

**Table 1.** Best inter-rater reliability kappa values

Code	idk	bor	easy	hard	int	ext	pos	neg	sup	unsup
Kappa	0.80	0.79	0.9	0.78	0.84	0.69	0.84	0.78	N/a	0.55

### 3 Results

First we report the frequency of each tag in all self-reports (Fig. 2). Then we examine the frequency of each tag given a particular self-reported affect state (e.g., “high frustration”). Forced choice Likert response of <3 was categorized as low, while >3 was categorized as high. Neutral responses (3 on the Likert scale) were not included in this analysis. Finally, we examine frequent combinations of tags over all affect reports.



**Fig. 2.** Frequency of each code out of a total sample group (2015 N = 449; 2011 N = 464) (Color figure online)

The second dataset, from 2015, was collected after several improvements were implemented in MathSpring, e.g., a screen showing student progress. Perhaps in part due to the improvements in MathSpring, the 2015 participants' reports (on average) were more often positive and less often negative in valence. Additionally, their reports were more frequently attributed to internal causes. They were less likely to report boredom, and were more likely to describe their work as hard or challenging. Figure 2 shows how frequently each tag occurred in each dataset.

Table 2 shows the frequency of each tag for each affect report (note: this includes only reports for which students included both a non-neutral affect rating and a text response). In 2015, only prompts on excitement and interest were given. Table 3 shows how often each tag occurred for each type of report, similar to Table 2.

**Table 2.** Percentage of tagged reports of affect containing each tag 2011

	IDK	Pos	Neg	Int	Ext	Sup	Unsup	Easy	Hard	Bor
Low Exc.	3.1	1.9	29.0	14.5	30.9	3.8	3.8	1.9	0.8	10.3
High Exc.	0.0	18.9	5.4	18.9	37.8	5.4	0.0	5.4	8.1	0.0
Low Int.	7.2	1.3	23.8	11.5	31.5	3.4	3.4	5.5	1.7	10.6
High Int.	8.7	21.7	8.7	21.7	26.0	8.7	0.0	0.0	4.3	0.0
Low Conf.	8.9	0.9	28.6	18.8	25.9	4.5	3.6	0.9	1.8	6.3
High Conf.	10.5	12.1	8.9	15.3	29.0	1.6	0.0	13.7	2.4	6.5
Low Frustr.	4.9	12.3	9.9	19.8	29.6	2.5	0.0	12.3	1.2	7.4
High Frustr.	3.6	0.0	33.1	13.0	31.4	5.3	5.3	0.0	2.4	5.9

Many of the tags appear frequently where expected, for example the “positive” tag appears most frequently in reports of high interest and high excitement. Some others are more surprising, such as the “external” tag, which is frequent in all reports and most frequent in reports of high excitement in 2011 and low interest in 2015.

Additionally, the modular nature of attributions for emotions can be combined to provide more complex meanings. For example a reason such as “Math is easy!” would get the tags “ext, easy” or external attribution and easy. Table 4 shows the most frequent tag combinations for each dataset. Reports where the student chose the neutral

**Table 3.** Percentage of tagged reports of affect containing each tag 2015

	IDK	Pos	Neg	Int	Ext	Sup	Unsup	Easy	Hard	Bor
Low Exc.	4.8	5.4	23.1	20.4	29.4	4.2	1.5	1.8	5.7	3.6
High Exc.	5.2	31.4	2.9	22.9	29	2.4	0.0	2.9	3.3	0.0
Low Int.	5.0	5.9	20.8	17.1	29.5	3.7	0.9	2.2	7.8	7.1
High Int.	6.4	27.9	3.0	23.2	28.3	5.2	0.0	3.9	2.1	0.0

**Table 4.** Most frequent tag combinations and frequency of occurrence for 2015 and 2011 datasets

Code combination	pos int ext	neg int ext	idk	ext easy	neg ext bor	neg ext	pos ext	pos int
Freq 2015	18.1	12.5	11.4	4.2	3.1	2.9	7.3	6.6
Freq 2011	3.3	11.4	13.7	4.1	2.6	14.4		
Code combination	ext hard	No code	ext bor	neg ext unsup	neg int	idk neg	neg int ext hard	Total (except no code)
Freq 2015	4.7						3.1	73.8
Freq 2011		5.7	4.5	3.1	2.5	1.4		62.8

“3,” or did not choose a rating of affect, but did include an open response are included here. If a code combination appears in only one dataset, its frequency is left blank in the other column.

While there are a total of 511 possible combinations, the given complex tag combinations were able to cover 62.8 % of all possible instances for 2011, and 73.8 % for 2015.

Some of the most common instances were “pos, int, ext” and “neg, int, ext” which often included a reference to a relationship between self and an external entity with an associated valence (e.g., “I like this program” or “I hate math”).

One interesting observation is that while “negative” co-occurs with “external” in many of the above combinations, including those two alone, it never co-occurs with “internal” unless “external” is also part of the combination. This indicates that negative attributions are more often blamed on external reasons, such as the software or the domain, rather than on one’s self (unless it is a relationship between self and other).

Finally, the other very frequent category was “idk” or “I don’t know.” When this appeared, it was most often the only tag given to the statement (e.g., “because I’m just not”), but it also sometimes co-occurs with “negative;” in these cases, students may offer no explanation in a manner that is hostile towards the system (e.g., “None of your business!”).

## 4 Discussion and Future Work

In general, hand coding and analyzing student text about specific affect self-reports has enabled us to explore reasons/attributions that further describe particular affective states. We have found similarities across disparate datasets for students that seemed to have had somewhat different experiences; not only in expected areas such as valence, but also with regard to intrinsic vs. extrinsic attributions and difficulty. For example, we were surprised that in one dataset students were more likely to report that the material was “hard” when reporting high interest or excitement than when reporting low interest or excitement, while in the other dataset the opposite was true.

One of the most overwhelming findings of this work is the prevalence of students who externalize their affect, especially negative valence emotions. It is important that we recognize the existence of this sizable group of students. In the future, we plan to look more in-depth at each of these areas in order to understand each of these groups of students better, as well as the relationship between their emotions and their reasons for them.

An advantage to our coding scheme is that it has also prompted us to think about possible new affective constructs. We can attempt to build models predicting these reason tags using highly contextualized features, instead of looking at emotions labels. This would imply inspecting the relationship between attributions and students' contextualized performance and behaviors to see if these attributions may be responsible for different behaviors, and vice versa.

**Acknowledgements.** Thanks to Jaclyn Ocumpaugh for all her support and help and to Samantha Tapia for helping as an additional coder. This research was funded by the National Science Foundation, #1324385, Cyberlearning DIP, Impact of Adaptive Interventions on Student Affect, Performance, and Learning; Burlison, Arroyo and Woolf (PIs). Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

## References

1. Arroyo, I., Beal, C.R., Murray, T., Walles, R., Park Woolf, B.: Web-based intelligent multimedia tutoring for high stakes achievement tests. In: Lester, J.C., Vicari, R.M., Paraguaçu, F. (eds.) ITS 2004. LNCS, vol. 3220, pp. 468–477. Springer, Heidelberg (2004)
2. Corbin, J.M., Strauss, A.: Grounded theory research: procedures, canons, and evaluative criteria. *Qual. Sociol.* **13**(1), 3–21 (1990)
3. Wixon, D.A., Ocumpaugh, J., Woolf, B., Burlison, W., Arroyo, I.: La Mort du Chercheur: how well do students' subjective understandings of affective representations used in self-report align with one another's, and researchers'? In: International Workshop on Affect, Meta-Affect, Data and Learning (AMADL 2015), p. 34 (2015)
4. Porayska-Pomsta, K., Mavrikis, M., Pain, H.: Diagnosing and acting on student affect: the tutor's perspective. *User Model. User-Adap. Interact.* **18**(1–2), 125–173 (2008)

# A Framework for Parameterized Design of Rule Systems Applied to Algebra

Eric Butler<sup>(✉)</sup>, Emina Torlak, and Zoran Popović

Department of Computer Science and Engineering, University of Washington,  
Seattle, WA 98195, USA

{edbutler,emina,zoran}@cs.washington.edu

**Abstract.** Creating a domain model (expert behavior) is a key component of every tutoring system. Whether the process is manual or semi-automatic, the construction of the rules of expert behavior requires substantial effort. Once finished, the domain model is treated as a fixed entity that does not change based on scope, sequence modifications, or student learning parameters. In this paper, we propose a framework for automatic learning and optimization of the domain model (expressed as condition-action rules) based on designer-provided learning criteria that include aspects of scope, progression sequence, efficiency of learned solutions, and working memory capacity. We present a proof-of-concept implementation based on program synthesis for the domain of linear algebra, and we evaluate this framework through preliminary illustrative scenarios of objective learning criteria.

**Keywords:** Intelligent tutoring systems · Program synthesis · Automated domain modeling · Artificial intelligence

## 1 Introduction

Creating an appropriate domain model (i.e., a set of rules capturing expert behavior) is an integral part of designing intelligent tutors. In general, the domain rules are considered intrinsically rigid and tied to the content to be learned, while the student model accounts for variability and specialization. Prior work on creating domain models relies heavily on expert modeling. This affects the expense of tutor development, estimated at 200–300 h per hour of content [9]. There has been some work on learning domain rules semi-automatically in the context of Intelligent Tutors [4, 7]. Both manual and semi-automatic processes, however, assume the domain model is defined by one canonical set of rules.

We postulate that the domain rule set is not a fixed entity, but one that can be specialized for each learning context by considering factors related to scope of coverage and sequence of progressions. Furthermore, instead of separating all student factors into the student model, we explore the effects of incorporating student population traits in the design of the domain rules. For example, in determining the optimal rule set for introductory algebra, we consider the relative complexity of rule trigger conditions, and the working memory demands



related to the number of rules students need to remember. By considering more factors in the process of the domain model creation, we aim to more precisely target the domain model to the specific scope and sequence goals as well as student traits.

This paper presents RULESYNTH, a framework for distilling the domain model that is optimal for a specific set of learning objectives. Given an objective function and an initial (suboptimal) set of rules, RULESYNTH produces a new set of rules that collectively optimize the given learning objective. We focus on creating a domain model in terms of condition-action rules (akin to production rules in a cognitive tutor), so that it can be applied to any existing or new rule-based tutoring system. RULESYNTH creates new rules using DSL-driven inductive program synthesis.

The key contributions of this work are a new framework for customizing the domain model that optimizes certain feature properties, and a preliminary evaluation of a proof-of-concept implementation in the domain of introductory algebra. Our evaluation shows that different learning objectives lead to dramatically different rule sets and that we can do so efficiently enough for customized intelligent tutoring systems to become a reality in the near future.

## 2 System Description

In this paper, we focus on domain models for solving linear algebraic equations. We represent a domain model as a set of *condition-action rules* (akin to production rules in a cognitive tutor). Each rule has a *condition* for when the rule may be applied, and an *action* to perform the rule. Our goal is to automatically produce a domain model (i.e., set of rules) optimizing some learning criteria.

To work towards this goal, we built a proof-of-concept system, RULESYNTH, that produces the best rule set given an initial domain model for algebra problems and an objective function. The objective function captures student constraints (such as limited working memory) and goals (such as solving problems in a few steps). Our starting set of rules is listed in Table 1; we call these rules *axioms*. Our axioms, along with backtracking search, are sufficient to solve a large class of linear algebra problems. However, while simple to state and suitable for automated problem solving, this set of rules is difficult for humans to use, as it leads to inefficient solutions with many steps. RULESYNTH uses the axioms to synthesize a large set of *macro rules* that lead to shorter solutions, and it uses the objective function to select the best rule set from the resulting pool of rules.

## 3 Evaluation

To evaluate our framework, we used it to generate several novel rules for solving algebra problems, which are shown in Table 2. We then investigated a few hypothetical scenarios and objective functions on these synthesized rules. Our objective functions and cost models were hand-crafted, but, in principle, could

**Table 1.** The set of axioms used as input for our system.

Label	Description	Example
<b>A</b>	Additive Identity	$x + 0 \rightarrow x$
<b>B</b>	Adding Constants	$2 + 3 \rightarrow 5$
<b>C</b>	Multiplicative Identity	$1x \rightarrow x$
<b>D</b>	Multiplying by Zero	$0(x + 2) \rightarrow 0$
<b>E</b>	Multiplying Constants	$2 * 3 \rightarrow 6$
<b>F</b>	Division Identity	$\frac{x}{1} \rightarrow x$
<b>G</b>	Canceling Fractions	$\frac{2x}{2y} \rightarrow \frac{x}{y}$
<b>H</b>	Multiplying Fractions	$3 \left( \frac{2x}{4} \right) \rightarrow \frac{(2*3)x}{4}$
<b>I</b>	Factoring	$3x + 4x \rightarrow (3 + 4)x$
<b>J</b>	Pushing Negatives	$-(3x) \rightarrow (-3)x$
<b>K</b>	Expanding Negatives	$-x \rightarrow -1x$
<b>L</b>	Adding to Both Sides	$x + 4 = 2 \rightarrow x + 4 + -4 = 2 + -4$
<b>M</b>	Dividing Both Sides	$3x = 2 \rightarrow \frac{3x}{3} = \frac{2}{3}$
<b>N</b>	Multiplying Both Sides	$\frac{x}{3} = 2 \rightarrow 3 \left( \frac{x}{3} \right) = 2 * 3$

**Table 2.** A sample of the macro rules found and synthesized by our system, with example applications. Several are common rules taught in algebra such as combining like terms (**IB**) or moving a constant’s opposite to the other side of an equation (**LBA**).

Pattern	Example
BA	$x + 2 + -2 \rightarrow x$
LBA	$x + 2 = 3 \rightarrow x = 2 + -3$
MG	$3x = 6 \rightarrow x = \frac{6}{3}$
LBAMG	$3x + 2 = 1 \rightarrow x = \frac{1 + -2}{3}$
NHG	$\frac{x}{4} = 2 \rightarrow x = 2 * 4$
IB	$2x + 4x \rightarrow 6x$
LJIBD	$x + 3y = 2 + 3 \rightarrow x = 2 + 3 + -(3y)$
KMG	$-x = 5 \rightarrow x = \frac{5}{-1}$
BLBA	$3 + x + 2 = 1 \rightarrow x = 1 + -5$

be based on student data or generically defined based on cognitive principles. These sample scenarios are not exhaustive nor intended to be exemplars for what would be used in a real tutor. Rather, they are intended to illustrate how a variety of objective functions can produce different domain models for different situations, all starting from the same synthesized rule set. Based on our results, we believe that, through the crafting of appropriate objective functions (which may depend on student models and live data), tutors could use RULESYNTH to automatically adapt their domain models to particular learning situations.

### 3.1 Balancing Solution Size and Rule Set Size

Our first scenario considers balancing the size of the rule set and the efficiency of solutions. Given a set of rules  $\mathcal{R}$  and example problems  $\mathcal{E}$ , we define the objective function to be a weighted sum of the *rule-set cost* and the *solution cost*, subject to the constraint that all problems in  $\mathcal{E}$  are solvable with the chosen rule set  $\mathcal{R}' \subseteq \mathcal{R}$ . Thus, our objective function takes the form

$$C(\mathcal{R}', \mathcal{E}) = \arg \min_{\mathcal{R}' \subseteq \mathcal{R}} \alpha R(\mathcal{R}') + (1 - \alpha) S(\mathcal{R}', \mathcal{E}) \tag{1}$$

where  $\alpha \in [0, 1]$  is a weighting term,  $R(\mathcal{R}')$  is the rule-set cost, and  $S(\mathcal{R}', \mathcal{E})$  is the solution cost.

We define the rule-set cost to be the sum of the costs of its rules, i.e.,  $R(\mathcal{R}') = \sum_{r \in \mathcal{R}'} \text{cost}(r)$ . The cost of a rule is itself a weighted sum of costs of its condition and action. The condition cost measures the size of the condition expression, thus estimating the amount of work required to evaluate the condition. The action cost is defined as the number of elements that are added or removed to the equation during the application of a rule. For example, moving a term to the other side of an equation costs two: one to remove the term and one to add it to the other side. Intuitively, macros tend to have more expensive conditions (i.e., they are harder to apply) but lower action costs than the axiom subsequence they replace (because they compress the replaced actions into fewer steps).

The solution cost  $S(\mathcal{R}', \mathcal{E})$  minimizes the average solution cost over all example problems  $\mathcal{E}$ . That is,  $S(\mathcal{R}', \mathcal{E}) = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \min_{\{r\}_n \in \text{solns}(\mathcal{R}', e)} \sum_{i=1}^n \text{cost}(r_i)$ . The function  $\text{solns}(\mathcal{R}', e)$  is defined as the set of all finite sequences of rules from  $\mathcal{R}'$  which solve the problem  $e$ . We therefore take the cost of a problem  $e \in \mathcal{E}$  to be the sum of the cost of every rule in the shortest solution to that problem.

We considered two different versions of the optimization problem defined by Eq. 1, which set the weighting term  $\alpha$  to nearly 1 and to 0.3, respectively. The first version ( $\alpha \approx 1$ ), which we call Optimization A, represents an objective function that tries to find all of the rules that are used in the shortest solutions to  $\mathcal{E}$ , discarding only unused rules. The second version ( $\alpha = 0.3$ ), which we call Optimization B, represents a trade-off between having efficient solutions and keeping the total size of the rule set small.

Table 3 compares the results of running each optimization on our synthesized rules. For Optimization A, which considers only average solution cost, the rule set includes a large number of rules based on macros. This is because almost every macro makes at least one example more efficient to solve. Also, there are very few axioms. Most of them, while generally applicable, are obsolesced by one or more macro rules. On the other hand, Optimization B, which balances average solution cost with total rule cost, contains many more axioms and fewer macros. The axioms, while making solutions more expensive since more steps are required, are more broadly applicable since they can be used in combination. However, some macro rules which are themselves very broadly applicable (e.g., combining like terms) remain in the optimal solution. Making this weighting dynamic in a live tutor would enable the tutor to adjust the domain model along the spectrum of maximizing solution efficiency or minimizing rule set size.

**Table 3.** Optimization results for an objective function minimizing average solution cost and total rule cost. Optimization A considers only average rule cost whereas Optimization B balances the two.

Rules from Optimization A		Rules from Optimization B	
A	KMG	A	L
C	LBAMGLBA	B	IB
D	LJIBDLBA	C	NHG
MC	LJIBD	D	MG
F	IB	MC	LBA
G	NHG	F	
J	LBAMG	G	
BLBA	MG	H	
KNHGMG	LBA	J	
NHGMBHGMG		K	

### 3.2 Adapting Rule Sets to Teacher-Specified Problem Sequence

As another example scenario, suppose that we have a sequence of problems we wish to use (perhaps provided by a teacher), broken up into discrete units. RULESYNTH can automatically find a sequence of rule sets that cover the entire sequence of problems, introducing only the minimal number of rules when needed for each unit. That is, for each unit of problems, we would like the minimal set of rules (with respect to the cost defined in Eq. 1) that covers these problems and is a superset of the rule set for the previous sequence. Given a sequence of problem sets  $\mathcal{E}_1, \dots, \mathcal{E}_n$ , we solve  $n$  sequential optimizations, where the  $i^{\text{th}}$  objective function is  $C(\mathcal{R}_i, \mathcal{E}_i)$ , subject to the constraints that all of  $\mathcal{E}_i$  are solvable with  $\mathcal{R}_i$  and either  $i = 1$  or  $\mathcal{R}_i \supseteq \mathcal{R}_{i-1}$ . We chose such an example problem sequence and ran this optimization (with  $\alpha = 0.3$ ), showing the results in Table 4. As can be seen, RULESYNTH finds a small number of rules to add for each successive unit of problems. We only chose a few basic features for this optimization, but with a richer domain model, future versions of our framework can have a more sophisticated function for choosing progressions of rules.

**Table 4.** Optimization results for generating a sequence of rules. Each column is a successive unit of example problems. This tables shows which new rules (in addition to all previous columns) are required to cover the new set of problems.

Percent Coverage	25 %	50 %	75 %	100 %
New Rules Added	LBA, MB, BLAB D, C, A	NHG, LJIBD, E	IB, J	KMG, KNHGMG NHGHMHGMG

## 4 Related Work

There is a long history of work in learning within cognitive architectures [5]. In some of these architectures, there is a concept of “chunking” rules to create new rules. Our system explicitly is performing a similar kind of chunking by finding *macros* of the given set of rules, and synthesizing conditions and actions for these macros to create novel rules. More recently, researchers have looked at methods to help automate authoring domain models in tutors, including rule learning [4]. Closely related to our system is SimStudent, which is capable of inductively learning rules (for primarily algebra but also other domains) using Inductive Logic Programming (ILP) [7] and unsupervised learning of deep domain features with probabilistic grammars [8]. Other work used ILP to search for rules given example applications from experts [3]. Our work is similarly inductive but uses program synthesis techniques. Other research has explored approaches to adapting content on the fly to students, by, for example, using multi-arm bandits for problem selection [1]. We are specifically concerned with choosing rules and domain models instead of problems. Inductive Programming / Synthesis has been applied to problem and solution generation [2], hint/feedback generation [6, 11], and rule generation [10]. Previous work in rule learning focused on learning individual rules, while we explore adapting rule sets to given learning criteria.

## 5 Conclusion

This paper presents RULESYNTH, a framework for generating custom domain models that optimize desired learning objectives. RULESYNTH employs discrete optimization to select the best set of rules from a pool of axioms and synthesized macros, according to a desired objective function. Our proof-of-concept implementation for algebra is able to synthesize several novel macro rules and produce optimal rule sets for example objective criteria. Our plans for future work include expanding to other domains to evaluate the generality of our approach, and exploring the impact of this system in the tutor design process.

## References

1. Clement, B., Roy, D., Oudeyer, P.Y., Lopes, M.: Multiarmed bandits for intelligent tutoring systems. *J. Educ. Data Min.* **7**(2), 20–48 (2015)
2. Gulwani, S.: Example-based learning in computer-aided stem education. *Commun. ACM* **57**(8), 70–80 (2014)
3. Jarvis, M.P., Nuzzo-Jones, G., Heffernan, N.T.: Applying machine learning techniques to rule generation in intelligent tutoring systems. In: Lester, J.C., Vicari, R.M., Paraguaçu, F. (eds.) *ITS 2004. LNCS*, vol. 3220, pp. 541–553. Springer, Heidelberg (2004)
4. Koedinger, K.R., Brunskill, E., Baker, R.S., McLaughlin, E.A., Stamper, J.: New potentials for data-driven intelligent tutoring system development and optimization. *AI Mag.* **34**(3), 27–41 (2013)

5. Langley, P., Laird, J.E., Rogers, S.: Cognitive architectures: research issues and challenges. *Cogn. Syst. Res.* **10**(2), 141–160 (2009)
6. Lazar, T., Bratko, I.: Data-driven program synthesis for hint generation in programming tutors. In: Trausan-Matu, S., Boyer, K.E., Crosby, M., Panourgia, K. (eds.) *ITS 2014. LNCS*, vol. 8474, pp. 306–311. Springer, Heidelberg (2014)
7. Li, N., Cohen, W., Koedinger, K.R., Matsuda, N.: A machine learning approach for automatic student model discovery. In: *Educational Data Mining 2011* (2010)
8. Li, N., Schreiber, A.J., Cohen, W., Koedinger, K.: Efficient complex skill acquisition through representation learning. *Adv. Cogn. Syst.* **2**, 149–166 (2012)
9. Murray, T.: Authoring intelligent tutoring systems: an analysis of the state of the art. *Int. J. Artif. Intell. Educ.* **10**, 98–129 (1999)
10. Schmid, U., Kitzelmann, E.: Inductive rule learning on the knowledge level. *Cogn. Syst. Res.* **12**(3), 237–248 (2011)
11. Singh, R., Gulwani, S., Solar-Lezama, A.: Automated feedback generation for introductory programming assignments. In: *Proceedings of the 34th ACM SIGPLAN Conference on Programming Language Design and Implementation* (2013)

# Cognitive Tutors Produce Adaptive Online Course: Inaugural Field Trial

Noboru Matsuda<sup>1(✉)</sup>, Martin van Velsen<sup>2</sup>, Nikolaos Barbalios<sup>1</sup>, Shuqiong Lin<sup>1</sup>,  
Hardik Vasa<sup>3</sup>, Roya Hosseini<sup>4</sup>, Klaus Sutner<sup>2</sup>, and Norman Bier<sup>5</sup>

<sup>1</sup> College of Education and Human Development, Texas A&M University, College Station, USA  
mazda@tam.u.edu, nbarmpalios@gmail.com, vcmore.lin@gmail.com

<sup>2</sup> School of Computer Science, Carnegie Mellon University, Pittsburgh, USA  
vvelsen@gmail.com, sutner@cs.cmu.edu

<sup>3</sup> School of Information Sciences, University of Pittsburgh, Pittsburgh, USA  
hnvasa@gmail.com

<sup>4</sup> Intelligent Systems Program, University of Pittsburgh, Pittsburgh, USA  
hosseini.ec@gmail.com

<sup>5</sup> Open Learning Initiative, Carnegie Mellon University, Pittsburgh, USA  
nbier@cmu.edu

**Abstract.** We hypothesize that when cognitive tutors are integrated into online courseware, the online courseware can provide a new type of adaptive instructions, such as impasse-driven adaptive remediation and need-based assessments. As a proof of concept, we have developed an adaptive online course on the Open Learning Initiative (OLI) platform by integrating four new instances of cognitive tutors into an existing OLI course. Cognitive tutors were created with an innovative cognitive tutor authoring system called WATSON. To evaluate the effectiveness of the adaptive online course, a quasi-experiment was conducted in a gateway course at Carnegie Mellon University. The results show that the proposed adaptive online course technology is robust enough to be used in actual classroom with mixed effect for learning.

**Keywords:** Adaptive online course · Active learning · Cognitive tutors · Authoring by demonstration · SimStudent

## 1 Introduction

In the face of challenges in access, quality, and cost in higher education, many have looked to technology as a mechanism for controlling costs and increasing student success. There is strong evidence that an adaptive and personalized approach to online courseware increases the quality of learning for diverse learner populations. However, there is substantial literature on barriers to adoption of information technologies in higher education [1–3]. Among the barriers, cost to create and maintain technology-enhanced learning interventions is a key factor [4–6].

This paper explores questions of the expedient development and integration of adaptive learning technologies into larger learning environments. In particular, we argue that an efficient authoring facility for creating cognitive tutors that can seamlessly integrate

into online courseware provides a solution for the cost-effective construction of adaptive online courseware with high-quality individualized instruction. The resulting adaptive online courseware provides students with rich multimedia instruction and multimodal activities; e.g., different types of formative assessments ranging from a multiple choice question to a guided-problem solving driven by a cognitive tutor.

## 2 Overview of Adaptive Online Course

We hypothesize that to provide adaptive support, the system needs to have a *skill model* that represents a set of skills that students must learn [7]. In the proposed adaptive online course, there are two types of skill models—a skill model embedded in a cognitive tutor and a skill model defined in an online course. In this paper, we call the former skill model a *production skill model* and the latter a *literal skill model*.

In the current paper, we use Open Learning Initiative (OLI) maintained by Carnegie Mellon University as an example online course platform. OLI is currently unique in offering the platform with an integrated literal skill model [8]. Individual skills are tied to learning objectives and assessments (both formative and summative), supporting a learner model that is updated as learners interact with assessments.

There are different kinds of “adaptive” supports studied for online courses [9]. In the current context, we use the word “adaptive” to mean four things: (1) timely and contextualized scaffolding for learning by doing (aka tutored problem-solving), (2) the optimal problem selection, (3) the impasse-driven adaptive remediation supported by dynamic linkage between a learning-by-doing and online course material, and (4) the need-based adaptive assessment.

The first two adaptive supports are realized by adaptive instructions provided by cognitive tutors [10, 11]. In other words, integrating cognitive tutors to online courseware will provide the adaptive problem selection and adaptive problem-solving scaffolding within the individual cognitive tutors.

The third adaptive support, the *impasse-driven adaptive remediation*, is realized by the Wheel Spinning Detector and Hybrid Student Model. *Wheel spinning* in this context occurs when cognitive tutor fails to stop posing problems in the face of a student’s inability to meet the pre-defined level of mastery [12]. The Wheel Spinning Detector is a neural network-based classifier [13] that classifies a sequence of student’s attempts on cognitive tutors into two categories: a sequence of attempts that will eventually meet the mastery and the one that will never meet the mastery.

Once wheel spinning is detected on a particular skill, corresponding instructional material on the online courseware is identified and presented to the student. This can be done by the Hybrid Student Model that represents the relationship between production skills and literal skills. We speculate that there are different ways to formalize the relationship between two types of skills. One idea would leverage the predicates used in production rules (to represent conditions and operators) as indices for domain concepts. For example, one of the production rules to convert an English sentence (e.g., if p or q then r) into a propositional logic (e.g.,  $(q \vee r) \rightarrow r$ ) says “if a token in the English sentence is a conjunction connective, then enter the conjunction symbol.” Here, the predicates



“conjunction connective” and “conjunction symbol” are domain concepts that must be encoded as literal skills. We can therefore associate this production to two literal skills each corresponds to a concept of conjunction connective and conjunction symbol respectively.

The fourth adaptive support, the *need-based adaptive assessment*, is realized by an extension of the knowledge-tracing technique. When a cognitive tutor detects a mastery on particular skills, the system identifies assessment items that are most closely related to the mastered skills based on the Hybrid Student Model, and hide them from students. Cognitive tutors in this design also appear as formative assessment items (with adaptive scaffolding) just like other types of formative assessments embedded in online course. In other words, we design the “mastery” learning distributed across multiple cognitive tutors and formative assessment items. Since cognitive tutors on an adaptive online course can be designed to share skills, cognitive tutors can be dynamically dropped just like other formal assessment items.

### 3 WATSON: Web-Based Cognitive Tutor Authoring System

WATSON (**W**eb-based **A**uthoring **T**ool for adaptive tutoring **S**ystems on **O**nline courses) is a web-based cognitive tutor authoring system that has (1) a web-browser based WYSIWYG front-end interface builder and (2) an automated expert model builder back-end.

A cognitive tutor applies task independent adaptive instructions—i.e., immediate feedback and just-in-time hint—given an expert model. Therefore, WATSON provides a support for (1) creating a *tutoring interface* in which students solve problems and (2) creating an *expert model*.

The front-end interface builder allows authors to create a *tutoring interface* in a web-browser. Since our new authoring environment is used to create online content, we design the authoring tools to work within a web browser. The current version is equipped with a components toolbar, a drawing canvas, and the component properties panel (Fig. 1). The description of the student interface is represented in an XML format, which is then rendered as HTML5 by a JavaScript renderer in real time.

Once the tutoring interface is created, the author builds an *expert model* by teaching a teachable agent called SimStudent [14] on the tutoring interface that is just created. SimStudent is a machine-learning agent that learns cognitive skills sufficient to solve target type problems. The underlying technology is programming by demonstration in the form of inductive logic programming [15]. To apply the SimStudent technology to WATSON, we have developed application programming interfaces (APIs) that connect the tutoring interface to SimStudent. For example, steps performed on the tutoring interface by the author are sent to SimStudent as demonstrations of a target task. The steps performed by SimStudent are shown on the tutoring interface and the author can provide feedback on their correctness.

A formal evaluation that we conducted in a past showed that a cognitive model generated by SimStudent can accurately model trace 99 % of 2900 solutions that (real) students made during an algebra study.

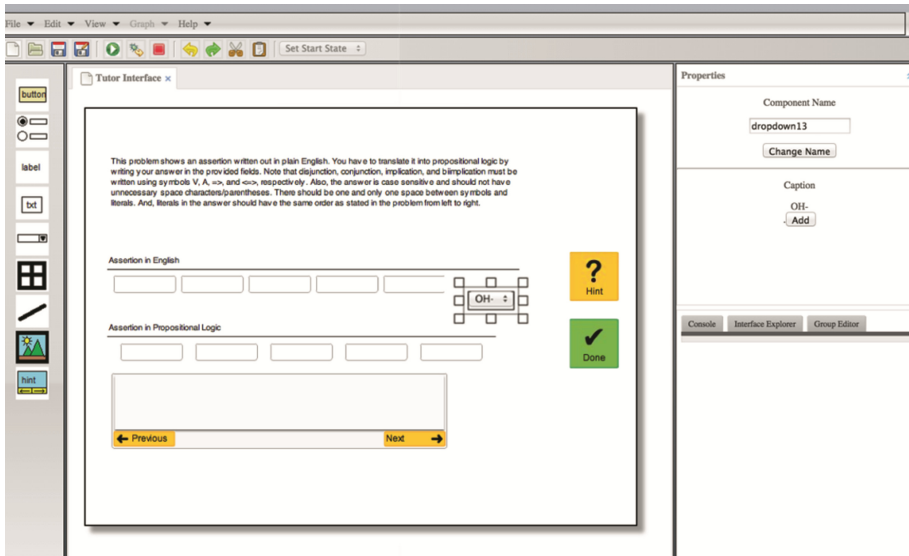


Fig. 1. An example screenshot of the WATSON web-based authoring tool

## 4 Evaluation Study

As a proof of concept, we developed an instance of adaptive online course by modifying an existing OLI course, Discrete Mathematics Primer (DMP for short), and used it for a gateway course (blended class) offered at Carnegie Mellon University for the Fall 2015 semester. The course was open exclusively to CS majors, and 133 out of 147 freshmen participated. The study ran for 3 weeks in August 2015.

Due to a schedule constraint on the system development, we have only implemented the knowledge-tracing engine for the current study. Cognitive tutors integrated into the DMP course adaptively provided feedback and just-in-time hint, but problems were hard-coded and all students were exposed to the same sequence of problems.

The classroom evaluation study was a quasi-experiment with a lack of control for specific study variables. To evaluate the impact on student's learning, we compare two versions of DMP: the original 2014 instance (DMP 2014), and the cognitive-tutor enhanced 2015 version (DMP 2015). We developed four cognitive tutors using WATSON (as described below) and integrated them into the first module of DMP 2015.

The first cognitive tutor (CT1) teaches to convert an assertion in informal language (e.g.  $p$  and  $q$  or  $r$ ) into an assertion in the propositional logic (i.e.  $p \wedge q \vee r$ ). The second cognitive tutor (CT2) teaches truth tables. The third cognitive tutor (CT3) teaches to push the negation in the given formula (e.g.  $\sim(\sim p \wedge q)$ ) all the way inside (i.e.,  $p \vee \sim q$ ). The fourth cognitive tutor (CT4) teaches to convert the given formula (e.g.  $\sim p \Rightarrow q$ ) into a negation normal form (i.e.,  $p \vee q$ ). Figure 2 shows an example screenshot of this cognitive tutor running on DMP 2015.

In this problem you have to convert the given formula into NNF, in a sequence of steps, each justified by one of the rules that you have learned so far. You are done after one step only if the conversion requires one rule. Note that disjunction, conjunction, implication, and bimplication must be written using symbols  $\vee$ ,  $\wedge$ ,  $\Rightarrow$ , and  $\Leftrightarrow$ , respectively. Also, the answer is case sensitive and should not have unnecessary space characters/parentheses. There should be one and only one space between symbols and literals. And, literals in the answer should have the same order as stated in the problem from left to right.

Formula

---

~p

>

q

Negation Normal Form

---

?  
Hint

✓  
Done

**Fig. 2.** An example screenshot of a cognitive tutor integrated in the Discrete Math Primer OLI course

We hypothesized that if DMP 2015 is more effective than DMP 2014, then students show steeper learning for DMP 2015 that results in larger slopes in a logistic regression model for learning-curve prediction [16]. Since skill names were redefined from 2014 to 2015, to make this comparison, we identified a mapping between skills for each year. The results show that there are two (out of the total of six) skill-mappings that show a steeper slope in DMP 2015 than DMP 2014; the difference in slopes in a logistic regression for these two mapped skills were 0.32 ( $Z = 6.47$ ,  $p < 0.0001$ ) and 0.52 ( $Z = 2.09$ ,  $p < 0.05$ ) respectively.

## 5 Conclusion

In this paper, we demonstrated that the WATSON technology allows online-course engineers to create cognitive tutors and to integrate them seamlessly into online courses. We used the Open Learning Initiative (OLI) as an example online learning platform and modified an existing online course with newly created cognitive tutors. Our classroom evaluation study shows the robustness of the adaptive online course, and data confirmed that the adaptive online course is better than or equal to the existing OLI course.

Our pilot study focused on the first module of the DMP course; integration of additional cognitive tutors into other two modules is currently underway, with use and evaluation planned for Summer 2016. The future study will also have incremental inclusion of technologies that was not implemented this time (e.g., wheel-spinning detector and hybrid student model).

The proposed integration of adaptive learning technologies and online learning provides a model for future, next generation online course. Such courses will integrate

a range of learning activities that span multiple modalities, but also span a range of development costs and a range of technical sophistication. Such a range will allow us to focus on efficacy, deploying less sophisticated (and less costly-to-develop) activities when approaches, but also create more sophisticated activities in efficient ways when necessary. This range of activities will provide a more effective and more efficient learning experience. This focus on efficacy will support cost and attainment goals, but will require additional research on how and when to best deploy more sophisticated technology and when pedagogically rich but technologically simple approaches will be as (or more) effective in supporting learners.

**Acknowledgement.** The research reported here was supported by National Science Foundation Award No. DRL-1418244.

## References

1. Reid, P.: Categories for barriers to adoption of instructional technologies. *Educ. Inf. Technol.* **19**(2), 383–407 (2012)
2. Bascow, L.S., et al.: *Barriers to Adoption of Online Learning Systems in U.S. Higher Education*. ITHAKA S+R, New York (2012)
3. Griffiths, R., et al.: *Interactive Online Learning on Campus: Testing MOOCs and Other Platforms in Hybrid Formats in the University System of Maryland*. ITHAKA S+R, New York (2014)
4. Parthasarathy, M., Smith, M.A.: Valuing the institution: an expanded list of factors influencing faculty adoption of online education. *Online J. Distance Learn. Adm.* **12**(2), 9 (2009)
5. Thille, C., Smith, J.: *The Open Learning Initiative: Cognitively Informed E-Learning*. The Observatory on Borderless Higher Education, London (2004)
6. Gannon-Cook, R., et al.: Motivators and inhibitors for university faculty in distance and e-learning. *Br. J. Educ. Technol.* **40**(1), 149–163 (2009)
7. Koedinger, K.R., Corbett, A.T., Perfetti, C.: The knowledge-learning-instruction framework: bridging the science-practice chasm to enhance robust student learning. *Cogn. Sci.* **36**, 757–798 (2012)
8. Bier, N., Strader, R., Zimmaro, D.: An approach to skill mapping in online courses. In: *Learning with MOOCs*, Cambridge, MA (2014)
9. EGA: *Learning to Adapt: A Case for Accelerating Adaptive Learning in Higher Education*. Education Growth Advisors, Boston (2013)
10. Ritter, S., et al.: Cognitive tutor: applied research in mathematics education. *Psychon. Bull. Rev.* **14**(2), 249–255 (2007)
11. Corbett, A.T.: Cognitive computer tutors: solving the two-sigma problem. In: Bauer, M., Gmytrasiewicz, P.J., Vassileva, J. (eds.) *UM 2001. LNCS (LNAI)*, vol. 2109, pp. 137–147. Springer, Heidelberg (2001)
12. Beck, J.E., Gong, Y.: Wheel-spinning: students who fail to master a skill. In: Lane, H., Yacef, K., Mostow, J., Pavlik, P. (eds.) *AIED 2013. LNCS*, vol. 7926, pp. 431–440. Springer, Heidelberg (2013)
13. Matsuda, N., Chandrasekaran, S., Stamper, J.: How quickly can wheel spinning be detected? In: *Proceedings of the International Conference on Educational Data Mining* (under review)

14. Matsuda, N., Cohen, W.W., Koedinger, K.R.: Teaching the teacher: tutoring SimStudent leads to more effective cognitive tutor authoring. *Int. J. Artif. Intell. Educ.* **25**, 1–34 (2015)
15. Muggleton, S.: Inductive logic programming. *New Gener. Comput.* **8**(4), 295–318 (1991)
16. Koedinger, K.R., Mathan, S.: Distinguishing qualitatively different kinds of learning using log files and learning curves. In: *Working Notes of the ITS 2004 Workshop on Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes* (2004)

# Optimizing Pattern Weights with a Genetic Algorithm to Improve Automatic Working Memory Capacity Identification

Jason Bernard<sup>1</sup>(✉), Ting-Wen Chang<sup>2</sup>, Elvira Popescu<sup>3</sup>,  
and Sabine Graf<sup>1</sup>

<sup>1</sup> Athabasca University, Athabasca, Canada  
c.j.bernard@ieee.org, sabineg@athabascau.ca

<sup>2</sup> Beijing Normal University, Beijing, China  
tingwenchang@bnu.edu.cn

<sup>3</sup> University of Craiova, Craiova, Romania  
popescu\_elvira@software.ucv.ro

**Abstract.** Cognitive load theory states that improper cognitive loads may negatively affect learning. By identifying students' working memory capacity (WMC), personalized scaffolding techniques can be used, either by teachers or adaptive systems to offer students individual recommendations of learning activities based on their individual cognitive load. WMC has been identified traditionally by dedicated tests. However, these tests have certain drawbacks (e.g., students have to spend additional time on them, etc.). Therefore, recent research aims at automatically detecting WMC from students' behavior in learning systems. This paper introduces an automatic approach to identify WMC in learning systems using a genetic algorithm. An evaluation of this approach using data from 63 students shows it outperforms the existing leading approach with an accuracy of 85.1 %. By increasing the accuracy of automatic WMC identification, more accurate interventions can be made to better support students and ensure that their working memory is balanced properly while learning.

**Keywords:** Working Memory Capacity · Student modeling · Genetic algorithm

## 1 Introduction

Working memory capacity (WMC) is a cognitive trait that influences the learning process, in terms of learning speed, memorization of learned concepts and effectiveness of skill acquisition [1]. WMC enables us to keep active a limited amount of information ( $7 \pm 2$  items) for a brief period of time [2]. Exceeding the WMC limit can reduce students' learning performance, reduce transfer of learning or increase the amount of time needed to learn [3, 4]. By identifying WMC, cognitive load can be individualized to the student which benefits the learning process. For example, an adaptive recommendation system could provide personalized suggestions for learning activities to students [5]. Furthermore, simple awareness of WMC supports students in making

better choices for self-regulated learning and teachers may factor in WMC when making interventions for their students.

Traditionally, WMC is measured by asking students to take a specific multitasking test such as operation span task (OSPAN) [6]. OSPAN is considered a stable and reliable test [7] and several online versions of this test have been created, such as WebOSPAN [8]. Although such tests are effective, they have the notable drawbacks of requiring additional time and effort from learners to do the test and the risk of inaccuracies due to factors such as the perceived importance of the test by the students, stress or fatigue [9].

To overcome these drawbacks, automatic approaches have been investigated which analyze students' behavior to identify WMC automatically while students are learning in a learning system. As a basis for such automatic approaches, several studies investigated and found relationships between WMC and other student characteristics as well as their relation to student behavior [e.g., 1, 10, 11]. To the best of our knowledge, only one automatic approach for identifying WMC is proposed so far. DeWMC (Detecting Working Memory Capacity) [12, 13] calculates WMC using six patterns which all contribute equally to the identification of students' WMC.

This paper presents a tool for automatic WMC identification called WMCID-GA. WMCID-GA is based on DeWMC [12] and extends it through the use of a genetic algorithm which optimizes the weights of patterns impacting the WMC calculation in order to improve the precision of identifying WMC.

The remainder of this paper is structured as follows. Section 2 introduces the proposed WMC identification approach. Section 3 describes the evaluation of WMCID-GA and Sect. 4 concludes the paper.

## 2 WMCID-GA

In this section, we start with introducing DeWMC, followed by presenting WMCID-GA (WMC Identifier-Genetic Algorithm) and how its genetic algorithm was built.

DeWMC [12] uses six patterns (five behavior patterns and one pattern related to learning styles based on the Felder-Silverman learning style model [14]) to calculate WMC. The five behavior patterns consider behaviors including linear navigation, constant reverse navigation, performing simultaneous tasks, recalling learned material, and revisiting passed learning objects. Each of the six patterns has been selected based on detailed investigations and evidence from literature that there exists a relation between the respective pattern and WMC [12]. To calculate WMC, DeWMC first extracts student data from a learning system's database and computes the respective patterns considering student behavior and their learning styles. For each pattern, a high or low value is associated to high or low WMC, based on existing studies from literature [12]. Then, for each learning session of a student, a WMC session value is calculated building the average of all pattern values. Subsequently, the overall WMC value is calculated by building a weighted average over all WMC session values, considering the amount of available behavior data per learning session as a weight.

WMCID-GA is based on DeWMC. It uses the same patterns and a similar concept to calculate WMC from these patterns, with the only difference that WMCID-GA is

using a weight for each pattern when building the WMC session value (instead of assuming that all patterns contribute equally to the WMC session value). To find the optimal weight for each pattern, WMCID-GA uses a genetic algorithm (GA) [15, 16] which is an optimization algorithm that utilizes concepts from evolutionary biology to solve optimization problems.

A GA represents solutions as genomes, where each genome consists of a set of numbers representing genes. To find the optimal weights of patterns, each genome consists of six genes (each for one pattern) where each gene has a range of values (representing the weight of the respective pattern) from 0.01 to 1.0 in increments of 0.01. A value of 0 is excluded since according to literature [12], each of these patterns has at least a small contribution to the WMC identification. To calculate the fitness/quality of a genome, the error between the actual WMC and the calculated WMC for each student in a given dataset is calculated and the average error over all students is used as fitness value. The calculated WMC is computed from the six patterns, as described above, using the genome's gene values as pattern weights. The GA starts by initializing the population ( $P$ ) with random values for each genome as no information is available on the potential quality for any weight value. In each generation,  $P/2$  genome pairs are selected for crossover using the roulette wheel technique and uniform crossover is used where each gene has a chance of being swapped equal to the crossover weight ( $C$ ). Then, uniform mutation is used on each new offspring where each gene has a chance of being mutated equal to the mutation weight ( $M$ ). After crossover and mutation, the new genomes are merged into the population and the genomes with the lowest fitness are culled until the population is size  $P$  again. Once the new population is built, a new generation starts. To promote finding the optimal solution, the generation number of the best solution ( $G_{best}$ ) is recorded and the GA stops only after another  $G_{best}$  generations passed without finding a new best solution. To prevent early termination, a minimum of 10,000 generations must pass before WMCID-GA can terminate.

### 3 Evaluation

In this section, the evaluation of WMCID-GA is reported, starting with presenting the dataset and describing the evaluation design and performance metrics. Subsequently, the optimization of parameters and overfitting reduction strategies are explained, followed by a discussion of the results.

To evaluate WMCID-GA, data from 63 undergraduate students on the five behavior patterns and the learning style pattern (identified by the Index of Learning Styles questionnaire [17]), and WMC (identified by WebOSPAN [8]) was used.

The evaluation consists of three parts. First, to find the optimal values for the parameters of the GA, an iterative experimental process was used. Second, an experimental process was also used to test overfitting reduction strategies and find optimal parameters for those strategies. Third, the optimal GA parameters and the optimal overfitting reduction strategies were then used to run WMCID-GA and get final results. In order to ensure generalizability to any datasets, 10 fold cross validation was used for each part of the evaluation.



To evaluate the performance of WMCID-GA in each part of the evaluation, three metrics were used: ACC measures the difference between a student’s actual WMC and the WMC identified by WMCID-GA. An ACC value is computed for each student and an average ACC is built, which provides details on the overall accuracy of WMCID-GA. LACC is the lowest ACC value in the assessment set and measures the worst case scenario for an individual student. %Match measures the percentage of students who were identified with reasonable accuracy. A threshold for reasonable accuracy of  $ACC > 0.7$  was calculated by considering the range of WMC values in the dataset and assuming that ACC has to be at least higher than half of this range.

In the first part of the evaluation, the GA parameters are optimized in the following order: population size ( $P$ ), crossover weight ( $C$ ) and mutation weight ( $M$ ). For each parameter, suitable parameter ranges or principles were investigated based on existing literature [15, 16], resulting in a set of possible values for each parameter. For the first parameter, WMCID-GA was executed iteratively for each value in the set while using a mid-range value for the remaining parameters. The parameter value which produces the best result is considered the optimal choice and used for all subsequent executions. This process is repeated for each parameter with the resulting optimal parameter settings shown in Table 1.

**Table 1.** Optimal parameter settings

Population	Crossover weight	Mutation weight
25	0.80	0.001

**Table 2.** Optimal overfitting reduction settings

Stratification	FEP	$min_{gen}$
On	On	25

With genetic algorithms, overfitting is a potential problem. This problem was addressed in the second part of the evaluation where the benefit of using two overfitting reduction techniques, stratification [18] and future error prediction (FEP) [19], was assessed through experimentation. For FEP, the optimal setting of an early termination parameter ( $min_{gen}$ ) was also investigated. Table 2 shows the optimal overfitting reduction settings.

In the third part of the evaluation, the optimal parameter and overfitting reduction settings were used to obtain a final result. The results for the three performance metrics are shown in Table 3, together with the respective results from DeWMC.

Comparing the results shows that WMCID-GA has outperformed DeWMC in every metric; thereby, showing that optimizing the pattern weights improves the overall accuracy of WMC identification as well as provides solutions that are fairer for each single student. By conducting a closer examination of the results for each individual student, it could be seen that WMCID-GA improved identification accuracy (ACC) for every individual. Additionally, students with WMC between 0.4 and 0.7 (60.3 % of students in the dataset) are identified better (average  $ACC = 0.898$ ) than students below 0.4 (average  $ACC = 0.820$ ) and above 0.7 (average  $ACC = 0.762$ ). These results still compare favorably to the corresponding results for DeWMC with an average ACC of 0.818 and 0.684 respectively. Most likely, this is caused by the GA not

**Table 3.** Result comparison between WMCID and DeWMC (top result bolded)

Approach	ACC	LACC	%Match
WMCID-GA	<b>0.851</b>	<b>0.694</b>	<b>0.893</b>
DeWMC [12, 13]	0.809	0.442	0.809

**Table 4.** Minimum, maximum, and average weights and percentage of activated learning sessions per pattern

Pattern	Min	Max	Average	Activated
Linear navigation	3	13	7	89.98 %
Constant reverse navigation	50	99	82	78.62 %
Performing simultaneous tasks	81	100	97	8.25 %
Recalling learned material	10	33	22	58.86 %
Revisiting passed learning objects	36	84	62	60.19 %
Learning styles	2	17	10	100.00 %

having enough data for students with very high and very low WMC, as 28.6 % of students have a WMC higher than 0.7 and 11.1 % have a WMC below 0.4. Therefore, a larger sample size could help improve the results of WMCID-GA even further.

For each pattern, the minimum, maximum and average weights across all folds are shown in Table 4. Additionally, Table 4 shows the percentage of learning sessions in which a pattern was activated. These results indicate that constant reverse navigation, performing simultaneous tasks and revisiting passed learning objects are more predictive of WMC than other patterns, however, further investigations have to be done with respect to performing simultaneous tasks as such behavior was only found in 8.25 % of the learning sessions.

## 4 Conclusions

This paper has introduced WMCID-GA, an approach for identifying students' working memory capacity (WMC) from their behavior in learning systems. WMCID-GA extends the rule-based approach DeWMC by optimizing the weights of patterns through the use of a genetic algorithm. An evaluation with data from 63 students shows that WMCID-GA is outperforming DeWMC in all investigated metrics and therefore, can provide more accurate WMC results for more students. The results also indicate that different patterns have different impact on the WMC identification.

By improving the precision of WMC identification and making it possible to identify WMC automatically while students learn, learning environments can be personalized, providing students with individualized recommendations for learning activities that help balancing the cognitive load to their WMC. By optimizing the cognitive load, students can have better learning outcomes and may require less time to learn [3, 4]. Furthermore, more accurate WMC information can help students make better choices for self-regulated learning by taking their WMC into account while teachers may make better individualized suggestions to their students to help them learn.

Future work will deal with investigating other optimization algorithms and hybrid algorithms for the given problem, to overcome some of the weaknesses of GAs.

**Acknowledgement.** The authors acknowledge the support of this research by Alberta Innovates Technology Futures, Alberta Innovation and Advanced Education, Athabasca University and NSERC. This work was also supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS – UEFISCDI, project number PN-II-RU-TE-2014-4-2604.

## References

1. Graf, S., Liu, T.-C., Chen, N.-S., Kinshuk, Yang, S.J.H.: Learning styles and cognitive traits – their relationship and its benefits in web-based educational systems. *Comput. Hum. Behav.* **25**(6), 1280–1289 (2009)
2. Miller, G.A.: The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychol. Rev.* **63**(2), 81 (1956)
3. Kirschner, P.A.: Cognitive load theory: implications of cognitive load theory on the design of learning. *Learn. Instr.* **12**(1), 1–10 (2002)
4. Teigen, K.H.: Yerkes-Dodson: a law for all seasons. *Theory Psychol.* **4**(4), 525–547 (1994)
5. Chang, T.-W., Kurcz, J., El-Bishouty, M.M., Graf, S., Kinshuk: Adaptive recommendations to students based on working memory capacity. In: *Proceedings of the International Conference on Advanced Learning Technologies*, Athens, Greece, pp 57–61, July 2014. IEEE (2014)
6. Turner, M.L., Engle, R.W.: Is working memory capacity task dependent? *J. Mem. Lang.* **28** (2), 127–154 (1989)
7. Klein, K., Fiss, W.H.: The reliability and stability of the turner and engle working memory task. *Behav. Res. Meth. Instr. Comput.* **31**(3), 429–432 (1999)
8. Lin, T.: Cognitive trait model for adaptive learning environments. Dissertation, Massey University, Palmerston North, New Zealand (2007)
9. Gohar, A., Adams, A., Gertner, E., Sackett-Lundeen, L., Heitz, R., Engle, R., Haus, E., Bijwadia, J.: Working memory capacity is decreased in sleep-deprived internal medicine residents. *J. Clin. Sleep Med.* **5**(3), 191 (2009)
10. Ford, N., Chen, S.Y.: Individual differences, hypermedia navigation, and learning: an empirical study. *J. Educ. Multimedia Hypermedia* **9**(4), 281–311 (2000)
11. Graf, S., Lin, T., Kinshuk, : The relationship between learning styles and cognitive traits – getting additional information for improving student modelling. *Comput. Hum. Behav.* **24** (2), 122–137 (2008)
12. Chang, T.-W., El-Bishouty, M.M., Graf, S., Kinshuk: An approach for detecting students’ working memory capacity from their behavior in learning systems. In: *Proceedings of the International Conference on Advanced Learning Technologies*, Beijing, China, pp 82–86, July 2013. IEEE (2013)
13. Chang, T.-W., El-Bishouty, M.M., Kinshuk, Graf, S.: Identifying students’ working memory capacity in learning systems. Technical report (2016)
14. Felder, R.M., Silverman, L.K.: Learning and teaching styles in engineering education. *Eng. Educ.* **78**(7), 674–681 (1988)
15. Grefenstette, J.J.: Optimization of control parameters for genetic algorithms. *IEEE Trans. Syst. Man Cybern.* **16**(1), 122–128 (1986)
16. Srinivas, M., Patnaik, L.M.: Genetic algorithms: a survey. *Computer* **27**(6), 17–26 (1994)

17. Felder, R.M., Solomon, B.A.: Index of learning styles North Carolina State University (1998). <http://www.engr.ncsu.edu/learningstyles/ilsweb.html>. Accessed 1 Jan 2016
18. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Mellish, C.S. (ed.) Proceedings of the 14th International Joint Conference on Artificial Intelligence, vol. 2, pp 1137–1145, August 1995. Morgan Kaufmann Publishers Inc. (1995)
19. Mitchell, T.: Machine learning, vol. 45. McGraw Hill, Burr Ridge (1997)

# Stratified Learning for Reducing Training Set Size

Peter Hastings<sup>1</sup>(✉), Simon Hughes<sup>1</sup>, Dylan Blaum<sup>2</sup>, Patricia Wallace<sup>2</sup>,  
and M. Anne Britt<sup>2</sup>

<sup>1</sup> DePaul University, Chicago, IL, USA  
`peterh@cdm.depaul.edu`

<sup>2</sup> Northern Illinois University, Dekalb, IL, USA

**Abstract.** Educational standards put a renewed focus on strengthening students' abilities to construct scientific explanations and engage in scientific arguments. Evaluating student explanatory writing is extremely time-intensive, so we are developing techniques to automatically analyze the causal structure in student essays so that effective feedback may be provided. These techniques rely on a significant training corpus of annotated essays. Because one of our long-term goals is to make it easier to establish this approach in new subject domains, we are keenly interested in the question of how much training data is enough to support this. This paper describes our analysis of that question, and looks at one mechanism for reducing that data requirement which uses student scores on a related multiple choice test.

## 1 Introduction

The Next Generation Science Standards (NGSS) provide detailed expectations about engaging students in the practices of constructing scientific explanations and engaging in arguments from evidence about important everyday phenomena using complex literacy and modeling skills [1]. Explaining how or why phenomena occur is a key goal of scientific research [1, 13]. However, most students have trouble with explanation and argumentation, particularly in science [6, 7, 9, 12]. In constructing an explanation, students provide an assertion that states how one or more factors lead to the to-be-explained phenomenon through one or more intermediate processes or mechanisms [3, 11, 13]. Insufficient domain knowledge prevents readers from making the connections required for creating a coherent representation of a scientific explanation [3, 10, 13].

The high level goal of Project READi is to provide a deeper understanding of how students *read* texts. An important method for assessing that skill is analyzing what they write. In this paper, we are focusing on explanatory essays that students write after reading several short documents. Our long-term goals

---

P. Hastings—The assessment project described in this article is funded, in part, by the Institute for Education Sciences, U.S. Department of Education (Grant R305F100007). The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Department of Education.

for this research are to be able to automatically provide analyses of student explanations, and to be able to extend these techniques to other domains. We are approaching the first goal, but the machine learning mechanism underlying our system is trained on over 1000 essays that have been meticulously annotated by human coders. To achieve the second goal, we need to understand just how much training data we need, and if there are more efficient mechanisms for training. Here we describe one technique.

## 2 Evaluation Context

To describe students' overall skill in constructing causal explanations from reading multiple documents of a variety of types (e.g., descriptive texts, graphs and maps), 10th-grade students in science classes were asked to read a set of documents and write an essay explaining the causes of a scientific phenomenon [2]. Students wrote their essays with the documents available and were given several hints to make sure they understood the task. Then, while they still had the texts, students were given 9 multiple choice questions to assess learning of the causal model with a low-production (high recognition) measure of learning. 1011 students received the coral bleaching assessment ("explain how and why coral bleaching rates vary at different times"). Human coders annotated the essays for concepts mentioned (e.g., increased coral stress) and the connectedness of their causal chains against our causal model (e.g., increased coral stress causes ejection of algae which causes coral bleaching — see [7] for a causal model of coral bleaching). Inter-rater reliability between two human scorers was high (average  $\kappa = 0.85$ ).

For a subset of the essays (440 students; 59.5 % female and 33.6 % Hispanic, 25.7 % African American, 20.0 % White, 4.5 % Asian) we analyzed their essay quality into four categories [7] based on the completeness and coherence of their explanations. On average, students had difficulty in constructing an explanation from multiple documents with only 30.9 % of the essays including an explanation with at least one intervening factor (highest quality). 25.7 % of the essays included no target concepts whatsoever (lowest quality), 15.7 % included at least one concept but it was not connected to the outcome, and 27.7 % made at least one connection to the outcome but with no intervening elements.

The high production essay task and the low production multiple choice measures did significantly converge on assessing student learning. First, there was a significant effect of essay quality category on multiple choice percent correct ( $F = 45.12$ ,  $MSE = 2.48$ ,  $p < .001$ ). The average percents correct on the multiple choice test for the four quality groups were 32 %, 47 %, 52 % and 67 %, respectively. Post hoc SNK found that those in the lowest essay category learned less than the middle two groups (which did not differ from each other) which both learned less than the highest quality group. Second, there was a significant correlation between the number of unique core concepts that students mentioned in their causal chains (claims) and their accuracy on the multiple choice measure (Pearson correlation = .43,  $p < .001$ ).

## 3 Stratified Learning

### 3.1 Identifying Concepts and Structure

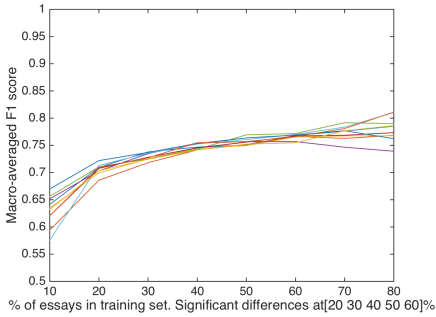
In earlier work, we experimented with a number of different machine learning techniques to detect the core concepts and claims in student essays from several different domains, including history and science [7,8, for example]. We compared the efficacy of a set covering algorithm using frequent multi-word expressions with that of Latent Semantic Analysis (LSA) at detecting how effectively students were using information from different sources when constructing evidence-based history essays [8, for example]. In more recent work, we have focused on detecting causality in scientific explanatory essays. To address this problem, we decomposed the problem of causality detection into two simpler problems, a word-tagging problem and a sentence classification problem. The word tagging problem involves predicting which concept code or codes, if any, are associated with each word in the essay. The sentence classification problem then involves taking the predicted concepts for each word in a sentence, aggregating these predictions across the sentence and then determining what causal relations exist between these identified concepts.

The word tagging problem requires the algorithm to predict varying numbers of concepts per word. This is called a Multi-Label Classification problem (MLC) and presents a challenge as most machine learning algorithms are designed to predict a single class at a time. To solve this problem, we use a problem transformation method called *Binary Relevance* (BR), in which you train a separate binary classifier for each concept code to be predicted. With BR, we moved a fixed-size sliding window of 7 words across the text, using the words within the window and their relative positions as separate features for the classifier [5].

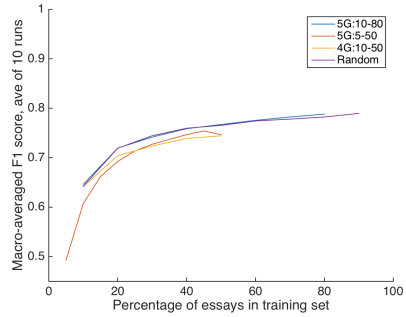
To solve the sentence classification problem, we use stacked generalization [15], feeding the predictions from the word tagging models as features into a ‘meta-classifier’. For each sentence, we create a separate feature for each concept code that was predicted for at least one word in the sentence, and also for each unique pair of predicted concepts. Additionally, we compute the minimum and maximum probabilities predicted by each classifier for each concept over all words in the sentence. Using these features, we then train a second ensemble of logistic regression classifiers to detect whether each sentence has a causal relation, and if so between which concepts.

### 3.2 Meta-Evaluation

As described in the introduction, this paper has two important goals: to determine just how much training data is needed to achieve a level of accuracy which is “acceptable” for providing relevant feedback to students, and to determine if there is a more effective training protocol which will enable us to reduce the required amount of training data. The training protocol we explore in this paper was inspired by stratified sampling from the world of statistics [14], so we call it Stratified Learning. One potential problem with machine learning approaches



**Fig. 1.** Averaged F1 scores



**Fig. 2.** Coarse-grained F1 scores

occurs when the concepts to be learned have skewed distributions. Stratified learning attempts to avoid this by taking advantage of prior knowledge about the data to be learned. In this case, when we are learning about student essays, we already have their scores on the multiple choice tests, and as mentioned above, we know that there is a moderate correlation between these scores and their essay quality. We hypothesize that by taking advantage of this prior knowledge, and by ensuring a balanced training set, the accuracy of the learned model would be greater than one trained with an equivalent number of essays from an imbalanced set.

With stratified learning, we start with a given, relatively small amount of the training set, and increase the training set size by that same percent, while ensuring that we get a (roughly) even number of items from each stratum or group. In the simple case, we used 5 groups<sup>1</sup> based on the multiple choice scores, and started with 10 % of the 1011 essays (in equal groups) in the training data. So there were approximately 22 essays from students with test scores of 0 or 1, 22 essays with scores of 2 or 3, and so on. Figure 1 shows the F1 scores for this technique for 10 different runs starting with 10 % and increasing up to having 80 % of the data in the training set. The F1 score for each essay was computed by averaging the accuracy of predictions for all the concept codes and causal connections within each sentence of the essay. Then the Macro-averaged F1 score was calculated based on all the remaining essays in the test set. The divergence that is evident at the 80 % level is presumed to be due to the relative scarcity of some groups in the test set at that point. (Note that the Y axis starts here at 0.5 in order to make the distinctions more obvious.)

The most obvious observations from this simulation are that there is consistent increase between the iterations, and that the largest jumps are on the left.

<sup>1</sup> The choice of group size is significant. As mentioned above, the distribution of multiple choice scores was fairly normal, and the least frequent score, 0, was assigned to 31 students. In order to maintain balanced representation of groups in the training set, some aggregation is necessary otherwise we could only test on a maximum of 31 items from each group. If the aggregation was too broad, however, it would decrease any benefit of balance in the training set.



As shown in the figure’s subtitle, most of the differences between iterations are statistically significant. In terms of the minimum accuracy that is required to provide meaningful feedback to students, we generally find  $F1 = 0.7$  to be a useful threshold. The machine learning technique was almost successful at achieving this level with only 10% of the essays (around 100), and it quickly reaches this level when going up to around 200 essays.

We also tested the approach using smaller increments of 5% (labeled “5G:5–50” in Fig. 2), and with 4 groups instead of 5, separated by multiple choice scores of 0–2, 3–4, 5–6, and 7–9 (labeled “4G:10–50”). This tested coarser granularity of group size, while maintaining roughly equal distribution. For this simulation, we took care to avoid the problem that was evident on the right side of Fig. 1, namely increased variance due to greatly diminished size of one or more groups in the *test* set. For this simulation, when we created the initial (balanced) training set at the beginning, we also created a separate, “held out” test set which would not be used as a source for additional training items in later iterations.

Figure 2 shows the results of these simulations, this time averaged over the 10 runs for clarity. This image makes it very clear that there is a robust increase in the accuracy going from 5 to 10 to 20% of the corpus. This makes us confident of one of the answers which we were after: 100–200 annotated essays should be sufficient to achieve acceptable levels of classification accuracy.

Figure 2 shows the performance of one additional method: random selection of equivalent increments of essay numbers from the corpus to add to the training set. In other words, this method does not use any balancing at all. Unfortunately for the Stratified Learning approach, the random approach is obviously every bit as effective, without the overhead of matching the scores on the multiple choice test. This provides an answer to the second question. If we are looking for a quicker route to reaching better performance, stratified learning is not it — at least in the case of our stacked learning context, as we will further discuss below.

## 4 Conclusions and Future Directions

In this paper, we have addressed two questions related to machine learning approaches for identifying structure in student explanatory essays: how much training data is required, and is training efficiency improved by maintaining a balanced training set. The exploratory goal showed us that a relatively modest training set size of 100–200 annotated essays should be sufficient to achieve adequate classification accuracy with our stacked machine learning mechanism. Our hypothesis about the benefits of stratified learning was not supported, however. There are several possible reasons for this. One is that although the correlation between multiple choice scores and essay quality is moderate, it is not especially high. Alternatively, there may be enough continuity between the lower- and higher-frequency groups that the random sample approach is not significantly disadvantaged relative to the stratified approach.

We have recently begun exploring another sampling mechanism called active learning [4], which shares some similarity with the last “imbalanced” technique

we described. With this approach, the system is trained on some subset of items, then attempts to classify the rest. Some of the items that it is least (or most, or some combination) confident in predicting are then added to the training set, and the process repeats. Early simulations show that this technique may actually increase learning efficiency over random selection.

## References

1. Achieve, Inc: Next Generation Science Standards: The common core standards for english language arts and literacy in history/social studies and science and technical subjects. Council of Chief State School Officers (2013)
2. Britt, M.A., Wallace, P., Blaum, D., Ko, M., Goldman, S.R.: Project READI science design team: multiple representations in science learning and assessment. In: Multiple Representations and Multimedia: Student Learning and Instruction. Symposium Conducted at the Annual Meeting of the AERA, Chicago, April 2015
3. Britt, M.A., Richter, T., Rouet, J.F.: Scientific literacy: the role of goal-directed reading and evaluation in understanding scientific information. *Educ. Psychol.* **49**(2), 104–122 (2014). doi:[10.1080/00461520.2014.916217](https://doi.org/10.1080/00461520.2014.916217)
4. Cohn, D., Atlas, L., Ladner, R.: Improving generalization with active learning. *Mach. Learn.* **15**(2), 201–221 (1994). doi:[10.1007/BF00993277](https://doi.org/10.1007/BF00993277)
5. Dietterich, T.G.: Machine learning for sequential data: a review. In: Caelli, T.M., Amin, A., Duin, R.P.W., Kamel, M.S., de Ridder, D. (eds.) *SPR 2002 and SSPR 2002*. LNCS, vol. 2396, p. 15. Springer, Heidelberg (2002)
6. Duschl, R., Osborne, J.: Supporting and promoting argumentation discourse in science education. *Stud. Sci. Educ.* **38**, 39–72 (2002)
7. Hughes, S., Hastings, P., Britt, M.A., Wallace, P., Blaum, D.: Machine learning for holistic evaluation of scientific essays. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M.F. (eds.) *AIED 2015*. LNCS, vol. 9112, pp. 165–175. Springer, Heidelberg (2015)
8. Hughes, S., Hastings, P., Magliano, J., Goldman, S., Lawless, K.: Automated approaches for detecting integration in student essays. In: Cerri, S.A., Clancey, W.J., Papadourakis, G., Panourgia, K. (eds.) *ITS 2012*. LNCS, vol. 7315, pp. 274–279. Springer, Heidelberg (2012)
9. Kelly, G.J., Druker, S., Chen, C.: Students' reasoning about electricity: combining performance assessments with argumentation analysis. *Int. J. Sci. Educ.* **20**(7), 849–871 (1998)
10. Meyer, B.J., Freedle, R.O.: Effects of discourse type on recall. *Am. Educ. Res. J.* **22**(1), 121–143 (1984)
11. Millis, K.K., Morgan, D., Graesser, A.C.: The influence of knowledge-based inferences on the reading time of expository text. *Psychol. Learn. Motiv.* **25**, 197–212 (1990)
12. Osborne, J., Erduran, S., Simon, S.: Enhancing the quality of argumentation in science classrooms. *J. Res. Sci. Teach.* **41**(10), 994–1020 (2004)
13. Osborne, J., Patterson, A.: Scientific argument and explanation: a necessary distinction? *Sci. Educ.* **95**, 627–638 (2011)
14. Shahrokh Esfahani, M., Dougherty, E.R.: Effect of separate sampling on classification accuracy. *Bioinformatics* **30**(2), 242–250 (2014). <http://bioinformatics.oxfordjournals.org/content/30/2/242.abstract>
15. Wolpert, D.H.: Stacked generalization. *Neural Netw.* **5**(2), 241–259 (1992)

# Combining Worked Examples and Problem Solving in a Data-Driven Logic Tutor

Zhongxiu Liu<sup>(✉)</sup>, Behrooz Mostafavi<sup>(✉)</sup>, and Tiffany Barnes

Department of Computer Science, North Carolina State University,  
Raleigh, NC 27695, USA  
{zliu24,bzmostaf,tmbarnes}@ncsu.edu

**Abstract.** Previous research has shown that worked examples can increase learning efficiency during computer-aided instruction, especially when alternatively offered with problem solving opportunities. In this study, we investigate whether these results are consistent in a complex, open-ended problem solving domain, where students are presented with randomly ordered sets of worked examples and required problem solving. Our results show that worked examples benefits students early in tutoring sessions, but are comparable to hint-based systems for scaffolding domain concepts. Later in tutoring sessions, worked examples are less beneficial, and can decrease performance for lower-proficiency students.

**Keywords:** Worked examples · Data-driven tutor · Problem solving

## 1 Introduction and Related Work

In this study, we focus on the pedagogical strategy of either presenting a logic proof problem to a student for completion (problem solving – PS), or providing a completed solution of the same problem for review (worked example – WE). Pedagogical strategies [9] are system-level policies that decide which action to take when multiple actions are available.

Worked examples are pedagogically beneficial, especially for inexperienced learners [3]. Interleaving worked examples and problems solving has been found to help students solve problems faster and more accurately on transfer post tests compared to blocked problem solving before worked examples [8]. Prior research has shown that Worked examples are more efficient and require less time on task than tutored problem solving. For example, McLaren and colleagues [4] found that replacing some tutored problem solving with isomorphic worked examples does not increase the learning effect but had significantly higher learning efficiency than problem-solving alone. However, in a more recent survey of the literature, Najar et al. concluded that the research is still inconclusive on when worked examples should be given; how they should be scaffolded; and how they should be designed [7]. Perhaps this is why most existing systems choose problem solving. In our prior work with the Deep Thought logic tutor, we showed that the addition of our data-driven worked examples reduced the time spent in

tutor by 27%, increased the amount of tutor completed by 14%, and increased retention in the system by 35% over students who were given problem solving opportunities alone [6].

In this paper we evaluate the impact of worked examples and problem solving opportunities on student performance in the Deep Thought tutor. We hypothesize that worked examples will reduce problem-solving time and decrease hint usage, but will not have an impact on the length of problem solutions or the percent of rule applications that are correct in first level problem solving. For later levels, we investigate the impact of the number of worked examples and their ordering with problem solving on overall performance in the tutor, as measured by time, number of hints, length of problem solutions, and rule application accuracy. This work will serve as the basis for future research on how and when we should use worked examples in data-driven problem solving environments.

## 1.1 The Deep Thought Tutor

**Problem Levels and Proficiency Track:** Deep Thought (DT) is a data-driven ITS for graphically constructing propositional logic proofs. DT presents proof problems consisting of logical premises and a conclusion to be derived using logic axioms. DT is divided into 6 strictly ordered levels of logic proof problems, each split into a high track with a few complex problems, and a low track with more simple problems. Level 1 is a single track, where students in the control group solve (S) three problems and the WE group solves 2–3 problems and receives 1–2 worked examples (E). At the end of each level, DT uses our data-driven knowledge tracing (DKT) system to assign students to the high or low track in the next level. This feature has been shown to reduce tutor dropout over versions of DT without problem selection or hints [5]. To ensure a fair comparison in this paper, we controlled for track placement in our analyses; this was not necessary in Level 1 where all students solve isomorphic problems.

**Data-driven PS Hint and WE:** DT utilizes data-driven hint generation via the Hint Factory, using prior student solutions to a problem to match problem states with new users, and giving hints that will guide students from their current state to the solution state [1]. The Hint Factory for DT leverages Interaction Networks constructed using prior student work to build a problem-specific domain model [2]. To create a worked example for a particular problem, we selected the shortest student solution in the Interaction Network that contained all the targeted logic rules for that problem. We then plug in information about the steps in the student solution into an annotation template, and present the WE step-by-step. Before deployment, experts checked our data-driven worked examples, to ensure their quality and correctness.

## 2 Methods

DT was used as in an undergraduate computer science class in Fall 2015. Course credit was awarded according to the number of levels completed. Students were

randomly split into two groups; the control group ( $n=24$ ) solved all problems, while the worked example (WE) group ( $n=51$ ) viewed 1–2 of those problems as worked examples. For the WE group, the number and order of worked examples was chosen randomly in each level so that students received 1–2 worked examples (E) and solved (S) the final problem. Low track sequences are: EESS, ESES, SEES, ESSS, SESS, SSES; and high track sequences are: EES, ESS, SES.

To study our hypothesis that worked examples improve learning efficiency, we compared the performance of the control and WE groups on Level 1. In Level 1, the control group solved three problems and the WE group was randomly assigned to view 1–2 examples and solve 2–3 problems. To study the impact of the number of worked examples, we compared performance on problems solved by track and number of worked examples across Levels 2–6. We investigated the impact of order of practice by studying student performance on the last problem in each level for high track levels. Low-track levels were not large enough to compare the impact of order.

Measures include: time, rule-application accuracy, solution length difference, and the number of hints requested. Rule application accuracy is the percentage of correct rule applications out of all applications a student attempted. Solution length difference is the number of steps a student used over the shortest recorded proof for the given problem. This is a good measure of student problem-solving ability comparable across levels, as shorter solutions usually indicate more expert-like knowledge. Comparisons between groups were made using the Kruskal-Wallis test for one-way analysis of variance, since normality assumptions were not met.

### 3 Results and Discussion

In this section, we first present descriptive statistics for the WE group to demonstrate that they read the worked examples and solved the planned number of problems in each level. We then present three studies: comparison of the control and WE groups in level one, the impact of the number of worked examples on problem-solving in levels 2–6, and the impact of ordering for high-track WE levels. Times that were 3 or more standard deviations from the mean were considered outliers and were not included in the analysis. This resulted in a cap of 25 min for each solved problem, and 6 min for each example, excluding 174 of 1936 problem instances (8.98%). We note that these cutoffs are necessary because students use DT through a web browser, and long times may indicate intense work or idle time that are not separatable in our data.

#### 3.1 Time and Practice Type for WE Group

Students with worked examples received an average of 7.5 ( $SD = 1.09$ ) worked examples, which accounts for 38.2% ( $SD = 5.25\%$ ) of the problems they encountered. WE students spent a mean of 10.02% ( $SD = 15.17\%$ ) of tutoring time on worked examples, and 5–10 sec on each step. We conclude that students are actually reading the worked examples, especially in earlier levels.

### 3.2 Comparison of Groups in Level 1

We then compared performance of the groups over all problems solved on Level 1. Hints were available for all 3 problems for the control group, but were not available for the 4th problem solved by the WE group. We found no significant difference between groups in learning time, hint usage, rule application accuracy, or solution length. Data for the WE group indicate that learning may have been more efficient for some students, but the variance in time, hint usage, and solution length difference was too high for this effect to be significant.

**Table 1.** Measurements for Level 1 in DT, by WE and control groups.

Level 1	Worked example group			Control group		
	Mean	Median	Std dev	Mean	Median	Std dev
<b>Hints/problem</b>	15.7	2.5	33.04	13.2	5.33	20.82
<b>Total time (min)</b>	33.18	26.64	23.32	35.28	29.1	22.83
<b>Time/problem (min)</b>	5.88	5.45	3.79	7.78	8.27	3.85
<b>Avg PS step time (sec)</b>	8.78	7.11	6.14	7.60	6.87	3.11
<b>% correct applications</b>	48.09	47.22	19.67	49.27	49.46	19.03
<b>Solution length difference</b>	2.93	1.33	2.32	3.17	2.0	2.15

From our data, the median time spent on worked examples by the WE group is under 1 min, for an average of 1.49 worked examples and 2.72 problem solved. This means that WE group solved almost three problems and viewed a worked example in about the same time that the control group solved three problems. This result confirms prior research that worked examples do not increase learning, but does not replicate the learning efficiency result. This may be also due to the availability of hints in DT. With the mean number of hints at 13 and 15, students in both groups clearly used bottom-out hints to generate step-by-step examples.

### 3.3 Number of Worked Examples

We further investigated the number of examples. In levels 2–6, the WE and control groups were not directly comparable, since our DKT assigned most WE students to the high track and control to the low track. This was because it assigned equal credit for actions in worked examples (viewing) and in problem solving (applying). Therefore, we aggregated data across levels 2–6 into groups by number of worked examples (0, 1, 2) and track. Table 2 reports central measures across all solved problems.

High-track levels with two worked examples have significantly shorter solutions than those with 0 or 1 examples. On the other hand, high-track levels with no worked examples had a higher proportion of correct rule applications. This,

**Table 2.** Measurements for high and low tracks in DT, by the number of worked examples encountered per level. \* indicates  $p < 0.05$ , † indicates  $0.05 \leq p < 0.1$ .

	High level track			Low level track		
# Worked examples	0	1	2	0	1	2
<i>n</i>	49	186	57	80	35	16
<i>Hints/problem</i>						
Mean	3.7	2.22	NA	2.22*	4.96*	4.18*
Median	0	0	NA	0.5	2.5	1.75
Std dev	10.7	8.36	NA	3.92	6.97	7.65
<i>Solution length difference</i>						
Mean	4.79*	5.08*	4.15*	4.3†	5.49	5.4†
Median	4.5	5	2	3.67	4.33	5
Std dev	3.17	3.62	3.84	3.02	3.32	2.78
<i>% Correct applications</i>						
Mean	0.76*	0.7*	0.7	0.7*	0.54*	0.54*
Median	0.81	0.7	0.71	0.7	0.55	0.54
Std dev	0.18	0.17	0.25	0.18	0.14	0.13
<i>Average problem solving time</i>						
Mean	386.78	376.30	338.21	244.71	242.89	299.51
Median	380.4	294.25	248.96	173.88	194.76	204.54
Std dev	266.60	252.77	290.55	181.63	213.01	217.45

along with short solutions, suggests that students with 0 or 2 examples may have more quickly learned a small set of rules to apply efficiently, while students with a single worked example may try applying more rules.

In the high-track, worked examples reduced dependence on hints. Surprisingly, low-track levels with no worked examples have significantly fewer hints and higher correct rule applications than low-track levels with worked examples. In this study, low track are those who have not demonstrated proficiency in problem solving, even with skill overestimation in the WE group. Therefore, we conclude that for low-proficiency students, worked examples increase both hint usage and the length of solutions. It may be that for DT, worked examples decrease self-regulation for low proficiency students solving simpler problems.

### 3.4 Worked Example Ordering

We hypothesized that interleaved PS practice and WE, with the PS occurring first, would result in better final problem performance as measured by time or length. In high-track Levels 2–6, we aggregated data based on the ordering of worked examples (E) and problem solving (S), where the possible orderings are EES, ESS, and SES. We studied only the high-track levels given the higher

ordering and fewer students in low track. We found no significant difference for any performance measurements. This result corresponds with previous research that worked examples work as well as problem solving for learning.

## 4 Conclusions

To summarize, we found that the impact of worked examples may be complex and individual in environments for open-ended complex problem solving. The results of our Level 1 controlled study show no significant differences in problem solving time, solution lengths, accuracy of rule applications, or hint usage per problem. However, the hint usage was high, showing that some students used bottom-out hints as worked examples. Worked examples seem valuable for students early on, but hints can provide some of the same scaffolding while encouraging students to self-regulate their learning.

To study the impact of the number of worked examples on learning, We aggregated data across levels 2–6 by high and low track. For the high track, having 0 or 2 worked examples improved solution length; high-track levels with no examples had higher hint usage. These results suggest that the lack of worked examples encouraged some students to choose when to see a hint. High track levels with one example had longer solutions and lower rule application correctness than no examples. Our low track represents true low-proficiency students, and in this track we found that worked examples had a negative impact: increasing hint usage and solution lengths, and decreasing rule accuracy. Together, these results suggest that worked examples detract from learning after Level 1 for both high and low tracks. We also investigated the impact of the order of examples and problem solving. We did not detect any significant differences in time, solution length, or accuracy based on the ordering of worked examples for our high track, and had insufficient data to compare the orderings in the low track. This result is consistent with prior research on worked examples.

## References

1. Barnes, T., Stamper, J.: Toward automatic hint generation for logic proof tutoring using historical student data. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 373–382. Springer, Heidelberg (2008)
2. Eagle, M.J., Hicks, A.G., Peddycord III, B., Barnes, T.: Exploring networks of problem-solving interactions. In: Learning Analytics and Knowledge (LAK 2015), pp. 21–30 (2015)
3. Kalyuga, S., Chandler, P., Tuovinen, J., Sweller, J.: When problem solving is superior to studying worked examples. *J. Educ. Psychol.* **93**(3), 579 (2001)
4. McLaren, B.M., Lim, S.-J., Koedinger, K.R.: When and how often should worked examples be given to students? In: Proceedings of the 30th Conference on Cognitive Science Society, pp. 2176–2181. Cognitive Science Society (2008)
5. Mostafavi, B., Liu, Z., Barnes, T.: Data-driven proficiency profiling. In: Educational Data Mining (EDM 2015), pp. 249–252 (2015)



6. Mostafavi, B., Zhou, G., Lynch, C., Chi, M., Barnes, T.: Data-driven worked examples improve retention and completion in a logic tutor. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M.F. (eds.) AIED 2015. LNCS, vol. 9112, pp. 726–729. Springer, Heidelberg (2015)
7. Najar, A., Mitrovic, A.: Should we use examples in intelligent tutors? In: Proceedings of Computers in Education, pp. 5–7 (2012)
8. Trafton, J.G., Reiser, B.J.: Studying examples, solving problems: contributions to skill acquisition (1993)
9. VanLehn, K.: The behavior of tutoring systems. *Int. J. Artif. Intell. Educ.* **16**(2), 227–265 (2006)

# NDLtutor: An Automated Conversational Agent to Facilitate Metacognitive Skills in Fully-Negotiated OLMs

Raja M. Suleman<sup>(✉)</sup>, Riichiro Mizoguchi, and Mitsuru Ikeda

School of Knowledge Science, Japan Advanced Institute of Science and Technology,  
Nomi, Ishikawa, Japan  
{suleman,mizo,iked}@jaist.ac.jp

**Abstract.** In this paper we discuss the findings related to our research on the paradigm of Negotiation-Driven Learning (NDL). Fully-negotiated OLMs have employed different negotiation mechanisms to support learner learning and reflection. In NDL research we are trying to combine and extend the best practices of previous OLMs to enhance the role of negotiations and promote cognitive and metacognitive learning in the context of fully-negotiated OLMs. This paper describes the findings of our research and introduces the NDLtutor, which is the realization of the NDL paradigm.

**Keywords:** Metacognition · Negotiation-Driven Learning · Interest-Based Negotiation · Affect · Behavior · Motivation · Natural-language dialogue · Self-reflection

## 1 Introduction

We have been working on the paradigm of Negotiation-Driven Learning (NDL) to enhance the role of negotiation in fully-negotiated Open Learner Models (OLMs) [1]. In fully-negotiated OLMs, learners have the ability to change their belief base which can be different from the belief of the system about their knowledge level [2, 3]. Such differences (conflicts) serve as the basis of a dialogue between the learner and the system where both of the parties collaboratively construct and maintain the Learner Model (LM). This strategy has shown to promote learning gains as well as enhance metacognitive skills in the learner [2]. OLMs have deployed different negotiation mechanisms to discuss the LM with the learner [2, 3, 5, 6]. In our research on NDL, we are trying to maximize the utility of this negotiation mechanism by adopting and extending previous best practices into a single system.

NDLtutor is a concretization of the NDL paradigm [1], where we seek to elevate the role of negotiation as a tool to promote learning gains both in the sphere of cognitive and metacognitive skills. NDL follows the notion that learning is maximized by spontaneous participation. When a learner is challenged by the system about the change they made in their belief base, they are inherently driven to defend this change. It is basic human psychology that humans become stronger advocates of their beliefs once they

are challenged, and are intrinsically motivated to defend their belief [4]. NDL aims at exploiting this opportunity created by the occurrence of a conflict to involve an intrinsically motivated learner in a deep learning dialogue which not only discusses the domain knowledge but also encourages them to assess the discussion to promote self-reflection. To this effect, NDLtutor is being developed to advance the current state of the art of dialogue capabilities that provide the learner with the tools and support to interact with the system in a naturalistic environment.

In this paper we introduce the NDLtutor, report the work we have done previously and discuss the results of our work in the light of future directions. The rest of the paper is organized as follows; the next section provides the background of our study. Next we present Negotiation-Driven Learning along with the outline of the architecture and implementation of NDLtutor. The next section provides the result and discussion about the evaluation followed by the concluding remarks.

## 2 Related Work

Early fully-negotiated OLMs explored different forms of the negotiation methods (menu selection and conceptual graphs) to provide the learner with the opportunity to interact with the system [3, 5]. However, it was noted that the negotiation methods used by these OLMs were not very flexible or naturalistic. To overcome this chatbots were used to provide a more naturalistic interface for negotiation [6].

Automated conversational agents have been shown to successfully engage learners and promote learning gains [7]. One of the main reasons that human tutors are more effective is hypothesized to be their use of natural language dialogue. Allowing the learner to interact with the system in natural language requires that the system is able to understand the learner's input. To deal with the complexities of natural language, different Natural Language Processing (NLP) techniques have been employed by ITSs with varying success [12].

Another important factor in the success of human tutors is their ability to interact with the learner according to their mental state [8, 9]. If a learner is in some sub-optimal state, they need to be supported to an optimal state for increased learning.

Metacognition has been recognized as a trait of effective learners and therefore much work has been done in the field of OLMs to continuously promote these skills in learners [10]. In current OLMs self-reflection is mostly implicitly implied by the externalization of the LM and the changes the learner makes to it.

## 3 Negotiation-Driven Learning

As mentioned earlier, this paper provides new insights to our previous work where we introduced our paradigm of Negotiation-Driven Learning (NDL) [1]. NDL aims to maximize learning participation by providing adequate support to the learners that allows them to interact with the system in a natural language environment. The basic philosophy of NDL is to engage a learner according to their mental state and to ensure that they remain in an optimal learning state. From previous work on modeling of affect

and motivation, we have selected 6 states, 3 affective states (*Confusion, Frustration, Engagement*) and 3 behavioral states (*Confidence, Interest, Motivation*) to be used in NDL through a comprehensive Wizard of Oz experiment [1].

### 3.1 The NDLtutor

NDLtutor provides a natural language interface to the learner to interact with the system. NDLtutor is different from its counterparts [3, 5, 6] in that it uses the *approximate* affective and motivational states of a learner to control the flow of the dialogue. To accomplish this we employed Interest-Based Negotiations (IBN) [11] as its negotiation strategy. Figure 1 shows the architecture of the NDLtutor.

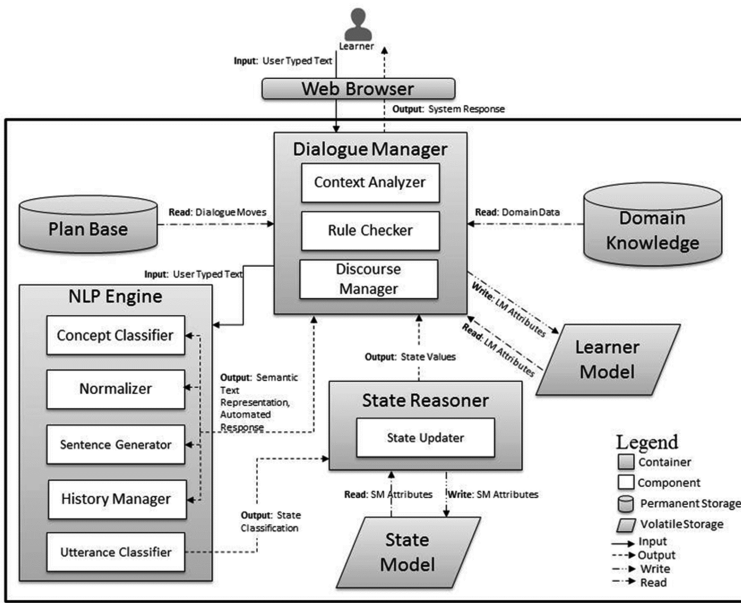


Fig. 1. NDL architecture adopted from [1]

### 3.2 Dialogue Design

One of the biggest challenges in the design of the NDLtutor was the dialogue management capability of the system. To design a dialogue management system, we needed to understand the dynamics of the possible learner interactions with our system. We conducted a Wizard of Oz study (WoZ) to create a basic classification of the learner input. Complete details of this experiment can be found in our previous work [1]. The data collected from the experiment allowed us to generate three main libraries; *User\_Utterance\_Library, System\_Utterance\_Library* and *Rules\_Library*.

### 3.3 Implementation

The backend of the NDLtutor has been implemented using PHP and MySQL whereas the frontend (user interface) has been designed using HTML5 and jQuery. The backend database consists of:

- *Domain Knowledge*: The domain knowledge is stored as plain text which is divided into topics and sub-topics. Each topic has 2 sets of questions; (1) *Multiple-Choice Questions* (MCQs) to assess the learner's performance (2) *Domain Discussion Questions* (DDQs) that are used to discuss the topic with the learner during the conflict resolution phase.
- *State Model*: The state model is stored as a list of attributes (states).
- *Learner Model*: The learner model is an overlay of learner's knowledge upon the domain.
- *Plan Base*: Consists of the dialogue moves that have been identified to work during a specific scenario to improve the system's response time.
- *The Reflection Log*: The database also stores the learner's responses during the reflection phase and this act as a self-assessment log for the learner to review at any time.

## 4 Evaluation of NDLtutor's Performance

We have planned a number of experiments to test the feasibility and applicability of our system. For the purpose of this paper, the evaluation was focused on investigating the following:

- The dialogue management capabilities of NDLtutor. (Quality of dialogues, Meaningful dialogues, Utility of the affective and behavioral selected states, Appropriate Feedback)
- Whether the inclusion of a reflective dialogue phase was beneficial for the learners?

### 4.1 Method

The participants for this evaluation were 20 students from the undergraduate Software Engineering program at Bahria University Islamabad, Pakistan. A pre-experiment test was conducted to generate an ad-doc LM for each participant. The average interaction time was 15.6 min. Post-experiment survey and interviews were conducted to get user feedback on the system.

### 4.2 Results and Discussion

As mentioned earlier, for this cycle of evaluation, we focused on the user's perceptions of the dialogue management capabilities of our system as well as the inclusion of a reflective dialogue as a means to promote self-reflection.

*The dialogue management capabilities of the NDLtutor* – The use of selected affective and behavioral states allowed the dialogues to progress smoothly, which suggests that the state selection together with the state-based dialogue management was appropriate to control the flow of dialogues. There were a total of 257 user utterances recorded. Out of these, 114 were domain-dependent, while the remaining 143 were domain-independent utterances. From the 143 domain-independent utterances, 129 (90.2 %) user utterances were successfully matched with the User\_Utterance\_Library while 14 (9.7 %) domain-independent user utterances could not be classified by our scheme.

*Whether the inclusion of a reflective dialogue phase was beneficial for the learners* – the answer to this question was retrieved from 3 sources:

- *The interaction logs of the reflection phase:* the interaction logs were analyzed to see learner’s responses during the reflection phase. The students were able to identify their knowledge gaps by comparing their initial answers to their final answers. This comparison allowed them to reflect upon their initial understanding and how it evolved during the course of the discussion.
- *Post-experiment survey:* the survey results in Table 1 showed that majority of the students were receptive of the reflective dialogue phase introduced in NDLtutor.

**Table 1.** Post-experiment survey

	<Strongly agree.....Strongly disagree>					Mean
	(5)	(4)	(3)	(2)	(1)	
Do you think discussing a topic with the chatbot was a good way of justifying your proficiency in that topic?	16	3	1	0	0	4.75
Do you think discussing a topic with the chatbot helped you improve your understanding?	13	4	2	1	0	4.45
Was the chatbot able to correctly understand what you wanted to say?	16	2	0	1	1	4.55
Were the system’s reactions to your inputs valid?	14	4	1	0	1	4.5
Did the chatbot make the negotiation process easy?	14	2	3	1	0	4.45
Did the use of off-topic discussion/ small talk make dialogue feel realistic/natural?	4	7	6	2	1	3.61
Did you find the reflection dialogue beneficial?	16	1	3	0	0	4.65
Would you be interested to use a similar system in the future as a study resource?	18	1	1	0	0	4.85

- *Post-experiment interviews:* students who were engaged in the reflective dialogue with the system were later interviewed to get their complete opinion about the system.

The students were very receptive of the reflection phase and found it to be very helpful in reviewing the dialogue and supporting self-reflection.

The analysis of the reflection phase showed that the participants did engage in self-reflection, however the role of the NDLtutor in this phase still needs further consideration. An observation made during the analysis of logs was that the less knowledgeable participants *gamed* the system by using the answers provided by the NDLtutor during domain discussion phase to generate their summarized answer. This allowed them to achieve high answer and concept coverage scores for their final answers. Such observations need to be further investigated and will be part of the future work.

## 5 Concluding Remarks

This paper presents the work we have done on our research on Negotiation-Driven Learning. By combining previous best practices, our work so far has produced very promising results. We understand the main reason for having such high rate of acceptance from the participants was partly because the survey was only focused on the dialogue management capabilities of the system. There are other major perspectives such as learning gains, managing high performing students etc. that this evaluation did not take into account. Including these perspectives will definitely affect the outcome of the evaluation study.

**Acknowledgments.** This research is partially supported by JSPS KAKENHI Grant Number 26240033 and by the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan.

## References

1. Suleman, R.M., Mizoguchi, R., Ikeda, M.: Negotiation-driven learning. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M. (eds.) AIED 2015. LNCS, vol. 9112, pp. 470–479. Springer, Heidelberg (2015)
2. Bull, S., Vatrappu, R.: Negotiated learner models for today. In: ICCE (2012)
3. Bull, S., Pain, H.: ‘Did I say what I think I said, and do you agree with me?’ Inspecting and questioning the student model. In: Greer, J. (ed.) AIED95, AACE, Charlottesville VA, pp. 501–508 (1995)
4. Gal, D., Rucker, D.D.: When in doubt, shout! Paradoxical influences of doubt on proselytizing. *Psychol. Sci.* (2010)
5. Dimitrova, V.: STyLE-OLM: interactive open learner modelling. *Int. J. Artif. Intell. Educ.* **13**, 35–78 (2003)
6. Kerly, A., Ellis, R., Bull, S.: CALMsystem: a conversational agent for learner modelling. *Knowl.-Based Syst.* **21**(3), 238–246 (2008)
7. Graesser, A.C., Wiemer-Hastings, K., Wiemer-Hastings, P., Kreuz, R., The Tutoring Research Group: AUTOTUTOR: a simulation of a human tutor. *J. Cogn. Syst. Res.* **1**(1), 35–51 (1999)
8. Du Boulay, B., et al.: Towards systems that care: a conceptual framework based on motivation, metacognition and affect. *Int. J. Artif. Intell. Educ.* **20**(3), 197–229 (2010)

9. Lehman, B., Matthews, M., D'Mello, S.K., Person, N.K.: What are you feeling? Investigating student affective states during expert human tutoring sessions. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 50–59. Springer, Heidelberg (2008)
10. Bull, S., Kay, J.: Metacognition and open learner models. In: The 3rd Workshop on Metacognition and Self-Regulated Learning in Educational Technologies, ITS 2008, pp. 7–20 (2008)
11. Fisher, R., Ury, W.: Getting to Yes: Negotiating Agreement Without Giving In. Penguin Books, New York (1983)
12. Boonthum, C., Levinstein, I.B., McNamara, D.S., Magliano, J., Millis, K.K.: NLP Techniques in Intelligent Tutoring Systems. IGI Global, Hershey (2009)



# Concept Maps Similarity Measures for Educational Applications

Carla Limongelli<sup>1</sup>, Matteo Lombardi<sup>2</sup>, Alessandro Marani<sup>2</sup>,  
Filippo Sciarro<sup>1</sup>(✉), and Marco Temperini<sup>3</sup>

<sup>1</sup> Engineering Department, Roma Tre University,  
Via della Vasca Navale, 79, 00146 Roma, Italy  
{limongel,sciarro}@ing.uniroma3.it

<sup>2</sup> School of Information and Communication Technology,  
Griffith University, 170 Kessels Road, Nathan, QLD 4111, Australia  
{matteo.lombardi,alessandro.marani}@griffithuni.edu.au

<sup>3</sup> Department of Computer, Control and Management Engineering,  
Sapienza University, Via Ariosto, 25, 00184 Roma, Italy  
marte@dis.uniroma1.it

**Abstract.** Concept maps represent a significant tool in education, used to plan and guide learning activities and to help teachers in some endeavors such as analyzing and refining their teaching strategies, retrieving suitable learning material, and supporting the provision of adaptive guidance in adaptive learning environments. Here we propose seven measures of similarity among concept maps, representing course modules. They deal with both structural and didactic aspects of the maps, to find out educational similarities among their associated course modules. The performance of the proposed similarity measures are analyzed and evaluated by means of some significant case studies.

**Keywords:** Concept map similarity · Learning · Ontology

## 1 Introduction

A Concept Map (CM) is a well established means for organizing concepts and the relationships among them in a easy and useful visual way. It is used in various fields, such as knowledge management, information systems development [3,4,6], collaborative work [9,10] and industrial fields [5]. A CM can be managed as either an ontology or a graph. In the literature the problem of computing the similarity among ontologies has been already addressed and many approaches have been suggested [11]. On the other hand, very few works consider the particular case of educational CMs. [2] proposes some ways to suggest the user for additional concepts and learning material during the creation of her CM. In [8] is addressed the matching of elements or parts among CMs, based on a similarity flooding algorithm, with the aim to support comparisons and merging of maps. This paper focuses on educational CMs, taking into account both structural aspects of the associated graphs and some didactic aspects, such as the

prerequisite relationships and the commonality of concepts, which are of capital importance to state the educational similarity between two CMs. An evaluation of the suggested measures is conducted to check the following research question: *Given two CMs, do the proposed similarity measures capture the didactic aspects of concepts commonality and prerequisite relationships?*

Section 2 presents the proposed measures, and Sect. 3 reports a first evaluation of the measures, concluding in Sect. 4 with foreseen future works.

## 2 Similarity Measures

In the following we present some different and independent ways for comparing two CMs. Let  $CM$  be a CM represented by a Direct Acyclic Graph (DAG), where nodes and edges represent, respectively, concepts and “prerequisite” relationships among concepts. We define the set of common nodes  $CN$  between two CMs  $CM_1$  and  $CM_2$  as follows:  $CN = \{CM_1 \cap CM_2\}$ . The *distance between nodes*,  $\delta(c_1, c_2, CM)$ , given a concept map  $CM$  and two nodes  $(c_1, c_2) \in CM$ , is defined as *the length of the shortest path from  $c_1$  to  $c_2$*  (or  $\infty$ , if there is no path). Moreover, the *Predecessor* of a node  $c$  in a concept map  $CM$  is defined as:  $Preds(c, CM) = \{\forall c_i \in CM \text{ such that there exists a path from } c_i \text{ to } c\}$ .

**Overlapping Degree (OD).** This measure analyses if there is a significant number of  $CN$  (note that  $|CN|$  is the *cardinality* of the set) (1) and how such nodes are placed in the maps (2). The following formula expresses how significant is the set of common nodes:

$$a = \frac{|CN|}{\min(|CM_1|, |CM_2|)} \in [0 \dots 1] \tag{1}$$

Then, adjacency matrices of the common nodes are built for both maps. The elements of these matrices are  $\delta(c_i, c_j, CM)$  for each pair of nodes  $c_i, c_j \in CN$ . The formula (2) computes the cosine similarity of the vectors of the two matrices, allowing to determine the similarity of the arrangement of the  $CN$  in the two maps.

$$b = \text{CosineSimilarity}(\overrightarrow{Adj}_{(CN, CM_1)}, \overrightarrow{Adj}_{(CN, CM_2)}) \tag{2}$$

where  $\overrightarrow{Adj}_{(CN, CM_i)}$  is the vector obtained from the linearization of the adjacency matrix of the common nodes in  $CM_i$ . The criteria pursued in (1) and (2) are then unified in (3):

$$OD = \frac{a + b}{2} \cdot \alpha, \text{ with } \alpha = \frac{2|CM_1 \cap CM_2|}{|CM_1| + |CM_2|} \tag{3}$$

Basically, the higher is the OD, the more similar and important is the arrangement of the common nodes in the maps. In other words, the same common nodes could be placed as a common subgraph of the two maps (higher similarity) or just be differently scattered in the maps (lower).

**Prerequisites Constraints Measure (PCM).** This measure determines the shared predecessors  $Preds$  of  $CN$  in the two maps. Given a concept  $k \in CN$ , let  $P_1$  and  $P_2$  be respectively  $Preds(k, CM_1) \cup k$  and  $Preds(k, CM_2) \cup k$ . The PCM is the sum of the following three elements:

$$a_k = \frac{|P_1 \cap P_2|}{|P_1 \cup P_2|}, \quad b_k = \frac{|CN \cap (P_1 \cup P_2)|}{|P_1 \cup P_2|}, \quad c_k = \frac{\min\{|P_1|, |P_2|\}}{\max\{|P_1|, |P_2|\}}$$

$a_k$  is the ratio of common predecessors on the total number of predecessors.  
 $b_k$  is the ratio of the number of predecessors in  $CN$  (they may not be common predecessors) on the total number of predecessors.  
 $c_k$  says the similarity of the amount of knowledge required by  $k$  in the two maps.

Given the three aforementioned elements, PCM is stated as follows:

$$PCM = \frac{1}{|CN|} \sum_{\forall k \in CN} \frac{a_k + b_k + c_k}{3} \tag{4}$$

In summary, this measure analyses the required knowledge for the  $CN$  shared in the two maps.

**Topological Similarity Measure (TSM).** TSM combines the purely structural measure given in (3) with the semantic information given in (4). Concepts might be differently scattered in the maps, so considering only their co-occurrence in the maps might be not enough. On the other hand, the structural information provided by OD can be an improvement to the PCM, so the following definition (5) tries to express a level of integration between the two previous measures:

$$TSM = \frac{OD + PCM}{3} \cdot \alpha \tag{5}$$

where  $\alpha$  is given in (3).

**Flux-Based Similarity Measure (FBSM).** By *flux* we mean a property of a node of the CM representing how much information is passing through it. The higher the *flux* of a node, the more “important” is the associated concept in the map. FBSM computes the similarity of importance of concepts in the two maps, expressed by the accumulated *flux*  $\varphi(c, CM)$  of the associated map nodes. The computation of FBSM is based on the spread activation technique [1]. In particular, let  $|CM_1| < |CM_2|$  and let  $c \in CM_1$ , when  $c$  is *activated* it receives *flux* equal to 1 in  $CM_1$ . If  $c \in CM_2$ ,  $c$  is activated in the second map too. In general, when a node receives *flux*, it retains at most an amount  $T$  ( $= 0.3$  in our case) that is added to its total flux:  $\varphi(c, CM) = \varphi(c, CM) + T$ . If there is any exceeding flux (which is  $flux - T$ ), such *flux* is spread to the child nodes evenly. So, a concept may receive *flux* from its own activation or from the predecessors. FBSM computes the sum of flux differences of the concepts in  $CM_1$  in the two maps. If  $c \notin CM_2$ , then  $\varphi(c, CM_2)$  is equal to 0.

$$FBSM = 1 - \frac{\sum_{c \in CM_1} \text{abs}(\varphi(c, CM_1) - \varphi(c, CM_2))}{|CM_1|} \tag{6}$$

**Flux-Based Similarity Measure on Common Nodes (FBSM-CN).** In this case, the same spread activation algorithm of measure (6) is used, but only the flux on  $CN$  is considered. This measure results in high scores if common nodes are similarly distributed in the two maps. Given the two vectors of the flux on  $c_1, \dots, c_i \in CN$  in  $CM_1$  and  $CM_2$ ,  $\vec{V}_1 = \langle \varphi(c_1, CM_1), \dots, \varphi(c_i, CM_1) \rangle$ , and  $\vec{V}_2 = \langle \varphi(c_1, CM_2), \dots, \varphi(c_i, CM_2) \rangle$  respectively, FBSM-CN is computed as follows:

$$\text{FBSM-CN} = \text{CosineSimilarity}(\vec{V}_1, \vec{V}_2) \quad (7)$$

**Comprehensive Flux-Based Similarity Measure (C-FBSM).** This measure combines the two previous flux-based measures given in (6) and (7):

$$\text{C-FBSM} = \frac{\text{FBSM} + \text{FBSM-CN}}{2} \quad (8)$$

**Comprehensive Similarity Measure (C-SM).** This measure is a linear combination of the Topological Similarity Measure (5) and the Flux-Based one (8):

$$\text{C-SM} = \text{TSM} \cdot (1 - \beta) + \text{C-FBSM} \cdot \beta \quad (9)$$

where

$$\beta = \left( \frac{\sum_{c \in CM_1} \text{outgoingArcs}(C, CM_1)}{|CM_1| - |\text{sinks}(CM_1)|} + \frac{\sum_{c \in CM_2} \text{outgoingArcs}(C, CM_2)}{|CM_2| - |\text{sinks}(CM_2)|} \right) \cdot \frac{1}{2 \cdot N}$$

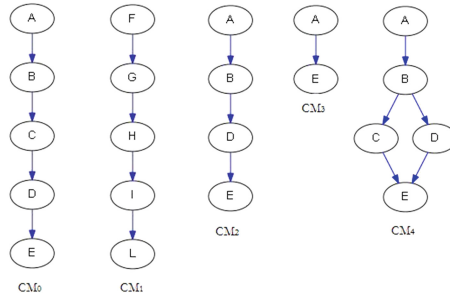
here,  $N \in [7..10]$  is a parameter of the algorithm, and *sinks* denotes the nodes having no successors. In practice,  $\beta$  is expected to express the significance of the flux-based measures according to the structure of the concept maps: the more the concept maps are linear or sequential, the less flux-based measures are expressive.

### 3 Evaluation

This section presents an evaluation of the similarity measures presented in Sect. 2 and Fig. 1 shows the sample of CMs used for this goal. The sample is composed by a set of five CMs which includes the seed ontology  $CM_0$  and its progressive variations;  $CM_0$  will be compared Vs. all the others, including itself. The rationale is to show the behavior of the proposed measures for different variations of the seed ontology  $CM_0$ , as suggested by ontology matching literature [11].

Here we discuss the five comparison cases, whose results are reported in Table 1:

**Evaluation I:**  $CM_0$  Vs.  $CM_0$ . This is the comparison between two identical maps, so all the similarity measures must be equal to 1, (cfr. Table 1).



**Fig. 1.** The sample of CMs.  $CM_0$  represents the seed CM.

**Table 1.** Results of Evaluations I–V using the similarity measures presented in Sect. 2.

Eval.	CMs	OD	PCM	TSM	FBSM	FBSM-CN	C-FBSM	C-SM
I	$(CM_0, CM_0)$	<b>1.0</b>	<b>1.0</b>	<b>1.0</b>	<b>1.0</b>	<b>1.0</b>	<b>1.0</b>	<b>1.0</b>
II	$(CM_0, CM_2)$	0.889	0.888	0.853	0.975	0.998	0.986	0.899
III	$(CM_0, CM_3)$	0.571	0.700	0.514	0.750	0.886	0.818	0.615
IV	$(CM_0, CM_4)$	0.976	0.967	0.973	0.680	0.958	0.819	0.890
V	$(CM_0, CM_1)$	<b>0.0</b>	<b>0.0</b>	<b>0.0</b>	<b>0.0</b>	<b>0.0</b>	<b>0.0</b>	<b>0.0</b>

**Evaluation II:**  $CM_0$  Vs.  $CM_2$ . This is the case where two CMs differ for a concept only, namely concept  $C$ . Not surprisingly, all the measures report a lower similarity than the previous case (refer to Table 1) but with different trends. FBSM-CN falls very slightly from 1 to 0.998, whereas TSM is the most sensible falling to 0.853. The other measures are in between.

**Evaluation III:**  $CM_0$  Vs.  $CM_3$ . As expected, the similarity measures still decrease because  $CM_3$  is a very small subset of  $CM_0$ ; it consists of only the source concept  $A$  and the target concept  $E$  of  $CM_0$ . All the similarity measures capture such situation, especially the flux based measures with the highest similarity values. This happens because  $A$  has the same amount of flux and  $E$  is a sink in both maps.

**Evaluation IV:**  $CM_0$  Vs.  $CM_4$ . This is the case where the FBSM-CN similarity presents the highest value with respect to the other measures: almost 1. This is because the Flux-Based measure captures a similar knowledge dissemination on concepts  $C$  and  $D$  in both maps; all the other measures increase report a more didactic similarity.

**Evaluation V:**  $CM_0$  Vs.  $CM_1$ . The two CMs are formed by all different concepts. Consequently, all the measures return a similarity score equal to 0.

In all the evaluation cases, we notice that the similarity measures were able to capture both topological and educational aspects (common concepts and prerequisites relationships) shared by a pair of CMs.

## 4 Conclusions

In this paper we addressed the problem of measuring the similarity among educational CMs. Seven similarity measures have been presented and evaluated in order to test their capability to capture both topological and educational differences between two concept maps. The evaluation shows that the research question is strengthened: all the measures are able to capture both topological and educational aspects. As a future work we plan to strengthen the evaluations of all the measures with a larger set of CMs involving teachers to assess their validity. Finally, the proposed measures would significantly benefit of tools for domain-based retrieval of synonyms, like SynFinder [7] or Word2Vec<sup>1</sup> for a more appropriate identification of common nodes of two CMs.

## References

1. Crestani, F.: Application of spreading activation techniques in information retrieval. *Artif. Intell. Rev.* **11**(6), 453–482 (1997)
2. Cañas, A., Leake, D.B., Maguitman, A.: Assessing conceptual similarity to support concept mapping. In: Fifteenth International Florida Artificial Intelligence Research Conference (2002)
3. Freeman, L.A.: The effects of concept maps on requirements elicitation and system models during information systems development. In: First International Conference on Concept Mapping, Pamplona, Spain, pp. 257–264 (2004)
4. Limongelli, C., Lombardi, M., Marani, A., Sciarrone, F.: A teacher model to speed up the process of building courses. In: Kurosu, M. (ed.) *HCII/HCI 2013, Part II. LNCS*, vol. 8005, pp. 434–443. Springer, Heidelberg (2013)
5. Limongelli, C., Miola, A., Sciarrone, F., Temperini, M.: Supporting teachers to retrieve and select learning objects for personalized courses in the moodle<sub>ls</sub> environment. In: IEEE International Conference on Advanced Learning Technologies, pp. 518–520. IEEE Computer Society (2012)
6. Limongelli, C., Lombardi, M., Marani, A., Sciarrone, F., Temperini, M.: A recommendation module to help teachers build courses through the moodle learning management system. *New Rev. Hypermedia Multimedia* **22**, 58–82 (2015)
7. Lombardi, M., Marani, A.: SynFinder: a system for domain-based detection of synonyms using WordNet and the web of data. In: Sidorov, G., Galicia-Haro, S.N. (eds.) *MICAI 2015. LNCS*, vol. 9413, pp. 15–28. Springer, Heidelberg (2015). doi:10.1007/978-3-319-27060-9\_2
8. Marshall, B., Chen, H., Madhusudan, T.: Matching knowledge elements in concept maps using a similarity flooding algorithm. *Decis. Support Syst.* **42**(3), 1290–1306 (2006)

<sup>1</sup> <http://deeplearning4j.org/word2vec>.

9. De Medio, C., Gasparetti, F., Limongelli, C., Sciarrone, F., Temperini, M.: Automatic extraction of prerequisites among learning objects using wikipedia-based content analysis. In: Micarelli, A., Stamper, J., Panourgia, K. (eds.) ITS 2016. LNCS, vol. 9684, pp. 375–381. Springer, Heidelberg (2016)
10. Novak, J.D., Canas, A.J.: The theory underlying concept maps and how to construct and use them. In: IADIS International Conference on e-Learning 2007, pp. 443–450 (2007)
11. Otero-Cerdeira, L., Rodríguez-Martínez, F.J., Gómez-Rodríguez, A.: Ontology matching: a literature review. *Expert Syst. Appl.* **42**(2), 949–971 (2015)

# Can Adaptive Pedagogical Agents' Prompting Strategies Improve Students' Learning and Self-Regulation?

François Bouchet<sup>1</sup>(✉), Jason M. Harley<sup>2</sup>, and Roger Azevedo<sup>3</sup>

<sup>1</sup> Sorbonne Universités, UPMC Univ Paris 06, CNRS,  
LIP6 UMR 7606, Paris, France  
francois.bouchet@lip6.fr

<sup>2</sup> Department of Educational Psychology,  
University of Alberta, Edmonton, Canada  
jharley1@ualberta.ca

<sup>3</sup> Department of Psychology, North Carolina State University, Raleigh, USA  
razeved@ncsu.edu

**Abstract.** This study examines whether an ITS that fosters the use of metacognitive strategies can benefit from variations in its prompts based on learners' self-regulatory behaviors. We use log files and questionnaire data from 116 participants who interacted with MetaTutor, an advanced multi-agent learning environment that helps learners to develop their self-regulated learning (SRL) skills, in 3 conditions: one without adaptive prompting (NP), one with fading prompts based on learners' deployment SRL processes (FP), and one where prompts can also increase if learners fail to deploy SRL processes adequately (FQP). Results indicated that an initially more frequent but progressively fading prompting strategy is beneficial to learners' deployment of SRL processes once the scaffolding is faded, and has no negative impact on learners' perception of the system's usefulness. We also found that increasing the frequency of prompting was not sufficient to have a positive impact on the use of SRL processes, when compared to FP. These results provide insights on parameters relevant to prompting adaptation strategies to ensure transfer of metacognitive skills beyond the learning session.

**Keywords:** Adaptivity · Pedagogical agents · Self-regulated learning · Metacognition · User perception

## 1 Introduction

Designing intelligent tutoring systems (ITSs) that dynamically adapt to learners' emerging understanding of content and to their use of metacognitive processes has been a major objective for the past decade [1]. Specifically, intelligent systems should provide learners with individualized instruction, feedback and scaffolding during their learning session [2], in a way that fosters the transfer of metacognitive skills beyond that session [3]. It is even more challenging in non-linear open-ended learning environments (OELEs) where no optimal way to navigate through the learning material



exists and where learners' goals may vary [4, 5]. Many critical questions remain unanswered: how often should learners be prompted to perform actions known to foster effective learning? Should prompts vary over time? How can instances where scaffolding should fade be detected?

In this study, we investigated the effect of adaptive prompting on undergraduates' learning and their use of self-regulated learning (SRL) strategies in an OELE with embedded pedagogical agents (PAs). Specifically, we examined how adapting PA prompting impacted learners': (1) use of SRL processes, (2) learning gains, and (3) perception of the system's usefulness. Our associated hypotheses were that: (1) learners should deploy more SRL processes overall, particularly once the scaffolding fades; (2) more efficient SRL should lead to higher learning gains with adaptive prompts; (3) system adaptivity should have a positive effect on learners' evaluation, but the more frequent initial prompting could have a negative effect by making the learners feel overwhelmed.

## 2 Method

### 2.1 Participants and Experimental Conditions

One hundred and sixteen undergraduate students ( $N = 116$ , 17–31 years old,  $M = 20.9$  years,  $SD = 2.4$ ; 64.6 % female; 62.9 % Caucasian) from two North American Universities, studying different majors and with various levels of prior knowledge participated in this study. Each participant received \$50 upon completion of the study and was randomly assigned to one of three experimental conditions: (1) *non-adaptive prompt* (NP –  $n = 29$ ), (2) *frequency-based adaptive prompt* (FP –  $n = 29$ ) and (3) *frequency and quality-based adaptive prompt* (FQP –  $n = 58$ ). Participants from adaptive conditions FP and FQP were grouped in some analyses, leading to two samples of identical sizes.

In the NP condition, learners received a moderate but constant amount of prompts from the PAs (on average, 1 per 10 min) to engage in various SRL processes. In the FP condition, learners received more prompts at the beginning of the session (on average, 3.5 per 10 min), but the probability of prompts being triggered decreased after each new prompt and after each self-initiated enactment of an SRL process. In the FQP condition, the same prompt decreasing rules as in FP apply, but the probability of prompts could also increase if: (1) the learner did not comply with a PA's prompt, or (2) a learner's metacognitive judgment was inaccurate (e.g., marked a page as relevant to their active sub-goal when it was not; cf. Table 1 for the list of conditions of success).

### 2.2 The Testbed System, Experimental Procedure and Data Used

**System overview.** MetaTutor [6] is an intelligent, hypermedia learning environment in which four embedded PAs help the student learn by prompting them to engage in SRL processes (cf. Table 1). A table of contents gives access to 38 pages (with text and images) on the human circulatory system. The overall learning goal is always visible,

**Table 1.** Condition of successes associated to the different type of SRL prompts.

Type	Type of PA's prompt	Condition of success
Monitoring	Judgment of Learning (JOL)	Accurate evaluation of what has been learnt
	Feeling of Knowing (FOK)	Accurate evaluation of what is already known
	Content Evaluation (CE)	Accurate evaluation of the relevance of the content relative to the active sub-goal
	Management of Progress Toward Goal (MPTG)	Learner validates their sub-goal in the next 45s
Strategy	Summarization (SUMM)	If learner delays, must be performed later on
	Coordination of Information Sources (COIS)	Image is opened in the next 45s
	Draw image already opened	Digital notepad in the next 45s
	Draw image not opened yet	Learner accepts to open the image

as well as two progress bars associated to the sub-goals chosen at the beginning of the session. A timer displays the time remaining in the learning session. One of the four PAs is always visible. Each PA has a specific role: *Pam the Planner* helps the student to plan their learning sub-goals, *Mary the Monitor* helps in monitoring the learning, *Sam the Strategizer* assists with the deployment of learning strategies and *Gavin the Guide* introduces the system and its questionnaires. The frequency and circumstances under which PAs' prompts are triggered depends on parameters such as the time spent on a page or the relevance of the page to students' current sub-goal. Below the PA, a palette of buttons allows students to self-initiate SRL processes (cf. Table 1), leading to a set of steps very similar to when the prompt comes from a PA: an invitation to perform the process followed by a feedback on its validity (e.g. agreeing the page is relevant to the current learning sub-goal).

**Experimental procedure.** The experiment involved two different sessions separated by one hour to three days. During the first one (30 to 40 min. long), participants filled and signed a consent form and completed several computer-based self-report questionnaires, a demographics survey and a pre-test on the circulatory system. During the second session (90 min. long), participants used MetaTutor to learn about the circulatory system. Participants had exactly 60 min to interact with the content during which they could initiate SRL processes or do so after a PA's prompt. MetaTutor was paused when participants were watching a video, taking a survey, and during an optional 5 min break half-way through the session. At the end of the session, participants were given a post-test and filled a questionnaire, the Agent Response Inventory (ARI) [7], which included questions on their perception of the quality of PAs' prompts. All participants completed their sessions individually on a desktop computer.

**Data coding and scoring.** Six variables were extracted from the pre-test and post-test questionnaires (two equivalent 25-item multiple choice tests on the human circulatory system), the ARI questionnaire, as well as from the system log files (cf. Table 2).

**Table 2.** List of the six variables used for analyses.

Variable name	Description
PropLearnGain	Proportional learning gains (between 0 and 1) using the standard formula: (posttest-pretest)/(1-pretest), for questions relevant to the 2 initial sub-goals and treating negative values as 0
UserAllProc_ [Session first30  last30]	Ratio (per period of 10 min) of all SRL processes initiated by the user during: the whole learning session/the first 30 min of the session/the last 30 min of the session
FBQuality[Mary Sam]	Learner's evaluation of the quality of the PA's feedback (1 to 7)

### 3 Results

In all of the following statistical analyses, an outlier screening was performed beforehand and outlying scores were replaced by the next most extreme score.

#### 3.1 Effects of Adaptive Prompting on the Use of SRL Processes

**Effect on learner-initiated SRL, overall.** A one-way ANOVA with prompt condition as the 3-level independent variable and UserAllProc\_Session as the dependent variable revealed a significant main effect of condition on learners' self-initiated SRL behaviors,  $F(2,113) = 10.17, p < .001, \eta_p^2 = 0.15$ . The application of a more stringent alpha ( $p < .01$ ) and the general robustness of ANOVAs to violations of assumptions supports the legitimacy of this finding and rendered a transformation unnecessary, despite equality of variances not being met (Levene's test). Follow-up post hoc comparisons using a Bonferroni correction revealed that the quantity of SRL behaviors that learners self-initiated were significantly different between the NP ( $M = 1.00; SD = 0.89$ ) and FP ( $M = 2.04; SD = 1.57$ ), and NP and FQP ( $M = 2.02; SD = 1.42$ ) conditions, but not between FP and FQP conditions.

**Effect on learner-initiated SRL, over time.** A repeated measures ANOVA with prompt condition as the 3-level independent variable, time as an independent 2-level within-subjects variable (first and last 30 min) and learners' self-initiated SRL processes as the dependent variable (i.e. UserAllProc\_first30 and UserAllProc\_last30) revealed a significant main effect of time on learners' self-initiated behaviors  $F(1,113) = 43.95, p < .001, \eta_p^2 = 0.27$  as well as a significant interaction effect of time and condition on learners' self-initiated behaviors  $F(2,113) = 6.65, p < .001, \eta_p^2 = 0.11$ ; both results remained significant after the application of a stricter alpha (related to results of Box's Test of Equality of Covariance Matrices). A significant main effect of condition on learners' use of SRL behaviors was found,  $F(2,113) = 7.61, p < .001, \eta_p^2 = 0.12$  (even with a more stringent alpha). An examination of Table 3 reveals that participants consistently engaged in more self-initiated SRL behavior during the second thirty minutes than the first, the most striking changes occurring in FP and FQP.

**Table 3.** Learner-initiated SRL processes by time and condition.

Variable	NP ( <i>n</i> = 58)		FP ( <i>n</i> = 29)		FQP ( <i>n</i> = 29)		All ( <i>n</i> = 116)	
	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>
	UserAllProc_first30	0.86	0.91	1.16	1.09	1.62	1.17	1.14
UserAllProc_last30	1.09	1.13	2.12	1.74	2.35	1.84	1.66	1.60

### 3.2 Effects of Adaptive Prompting on Learning Gains

Table 4 reveals no difference on average between conditions NP and FP&FQP, counter to our hypothesis that adaptive prompting would help with learning. However, when learning gains from NP and FP are compared, it appears that learners in the FP condition had a small benefit over those in the NP, and that FQP did not help.

### 3.3 Effects of Adaptive Prompting on Perceived System's Usefulness

Two one-way ANOVAs with prompt condition as the 3-level independent variable and FBQualitySam (resp. FBQualityMary) as the dependent variable failed to reveal a significant main effect of condition on learners' self-initiated satisfaction regarding the PAs. Descriptive statistics revealed that participants were most satisfied with Sam in the NP condition ( $M = 3.77$ ,  $SD = 1.63$ ) in comparison to Sam in the FP ( $M = 3.13$ ,  $SD = 1.77$ ) and FQP condition ( $M = 3.31$ ,  $SD = 1.79$ ). In contrast, participants were least satisfied with Mary in the FQP condition ( $M = 4.41$ ,  $SD = 1.74$ ) in comparison to Mary in the NP ( $M = 5.00$ ,  $SD = 1.95$ ) and FP condition ( $M = 4.95$ ,  $SD = 1.66$ ).

## 4 General Discussion

**Adaptive prompting helps learners to self-initiate SRL processes.** Learners in (pooled) condition FP&FQP deployed more SRL processes than those in condition NP, as they received more frequent prompting from the system. The number of learner-initiated processes increased over time despite the decrease of agent-initiated prompts, which can be interpreted as a residual and impactful effect of prompting. Our hypothesis was therefore verified. However, taking into account the quality of SRL processes to reduce PAs' prompts did not help: it may be because inefficient self-regulated learners need more than mere (potentially frustrating) reminders to self-regulate.

**Adaptive prompting may not directly help to improve learning.** We observed no significant differences in learning between conditions NP and FP&FQP, but the expected trend was there when comparing NP and FP. Therefore, it appears that the adaptiveness in FP was going in the right direction, contrary to the one in FQP. Hence our hypothesis was not supported, which could be partially explained by the fact learners might not have been left without scaffolding for long enough for a difference to appear.

**Table 4.** Learning-related variables in the 3 conditions considered.

Variable	NP		FP		FQP		FP&FQP	
	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>
Pre-test score (/1)	0.69	0.24	0.73	0.19	0.71	0.21	0.72	0.20
Post-test score (/1)	0.83	0.15	0.85	0.12	0.79	0.19	0.82	0.16
PropLearningGains	0.400	0.378	0.448	0.375	0.339	0.299	0.394	0.341

**Initially frequent but fading prompting doesn't degrade perceived system's usefulness.** We observed that PAs in FP and FQP were not perceived as less helpful than in NP, despite more frequent prompting at the beginning of the session, which could have been detrimental to learners' willingness to follow PAs' recommendations. Conversely, learners who appreciated PAs' interventions could have found them less useful overall as they were less present towards the end.

**Limitations and future work.** Although this study benefited from a significantly larger sample size than [8], a larger sample size (with as many participants in FP as in NP) may have led to more significant results. The limited duration of the learning session (1 h) might also have prevented observing internalization and integration of the use of SRL processes by learners once agents' scaffolding was fully gone [9]. Another limitation is the lack of evaluation of the importance of the progressiveness in the scaffolding reduction: another condition with frequent prompting for half a session and no prompting for the second half would be necessary to do so. Finally, we have seen that the adaptation exclusively in terms of frequency of prompting might have been detrimental to learners in condition FQP, and that the quality of the feedback should also be adjusted—confirming its importance [10]. The next steps are to test this approach on other systems, on longer periods of time and to have a finer-grained adaptation.

## References

1. Azevedo, R., Aleven, V. (eds.): *International Handbook of Metacognition and Learning Technologies*. Springer, New York (2013)
2. Bannert, M., Mengelkamp, C.: Scaffolding hypermedia learning through metacognitive prompts. In: Azevedo, R., Aleven, V. (eds.) *International Handbook of Metacognition and Learning Technologies*, pp. 171–186. Springer, New York (2013)
3. Roll, I., Aleven, V., McLaren, B.M., Koedinger, K.R.: Improving students' help-seeking skills using metacognitive feedback in an intelligent tutoring system. *Learn. Instr.* **21**, 267–280 (2011)
4. Land, S.M.: Cognitive requirements for learning with open-ended learning environments. *Educ. Technol. Res. Dev.* **48**, 61–78 (2000)
5. Kinnebrew, J.S., Gauch, B.C., Segedy, J.R., Biswas, G.: Studying student use of self-regulated learning tools in an open-ended learning environment. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M. (eds.) *AIED 2015. LNCS*, vol. 9112, pp. 185–194. Springer, Heidelberg (2015)

6. Azevedo, R., et al.: The effectiveness of pedagogical agents' prompting and feedback in facilitating co-adapted learning with MetaTutor. In: Cerri, S.A., Clancey, W.J., Papadourakis, G., Panourgia, K. (eds.) ITS 2012. LNCS, vol. 7315, pp. 212–221. Springer, Heidelberg (2012)
7. Harley, J.M., Carter, C.K., Papaioannou, N., Bouchet, F., Landis, R.S., Azevedo, R., Karabachian, L.: Examining the potential of personality traits and trait emotions to create emotionally-adaptive intelligent tutoring systems. *User Model. User-Adapt. Interact.* **26**, 1–43 (2016)
8. Bouchet, F., Harley, J.M., Azevedo, R.: Impact of different pedagogical agents' adaptive self-regulated prompting strategies on learning with MetaTutor. In: Lane, H., Yacef, K., Mostow, J., Pavlik, P. (eds.) AIED 2013. LNCS, vol. 7926, pp. 815–819. Springer, Heidelberg (2013)
9. Azevedo, R.: Issues in dealing with sequential and temporal characteristics of self- and socially-regulated learning. *Metacogn. Learn.* **9**, 217–228 (2014)
10. VanLehn, K., Graesser, A.C., Jackson, G.T., Jordan, P., Olney, A., Rosé, C.P.: When are tutorial dialogues more effective than reading? *Cogn. Sci.* **31**, 3–62 (2007)

# Automatic Extraction of Prerequisites Among Learning Objects Using Wikipedia-Based Content Analysis

Carlo De Medio<sup>1,2</sup>, Fabio Gasparetti<sup>1(✉)</sup>, Carla Limongelli<sup>1</sup>,  
Filippo Sciarrone<sup>1,2</sup>, and Marco Temperini<sup>2</sup>

<sup>1</sup> Engineering Department, Roma Tre University,  
Via della Vasca Navale, 79, 00146 Roma, Italy  
carlo.demedio@hotmail.it, {gaspare,limongel,sciarro}@ing.uniroma3.it

<sup>2</sup> Department of Computer, Control and Management Engineering,  
Sapienza University, Via Ariosto, 25, 00184 Roma, Italy  
marte@dis.uniroma1.it

**Abstract.** Identifying the pre-requisite relationships among learning objects is a crucial step for faculty and instructional designers when they try to adapt them for delivery in their general education distance courses. We propose a general-purpose content-based approach for facilitating this step by means of semantic analysis techniques: the learning objects are associated to Wikipedia pages (topics), and their dependency is obtained using the classification of those topics supported by Wikipedia Miner.

## 1 Introduction

Collecting educational materials to configure courses is a challenging activity for the teacher. Learning resources are often not to be treated as a mere additive on the activities proposed to students, yet the new resources have to undergo some pedagogical adaptation.

One of the most relevant skills, required while assembling LOs in a course, is in ensuring that pedagogical aspects of the course are preserved by the sequencing of the LOs. One of such aspects is the relationship of *dependence* between two LOs, which must not be betrayed in any instance of the course. In other words, being  $LO_i$  and  $LO_j$  two LOs in the course, with  $LO_i$  known as a prerequisite of  $LO_j$ , it must be assured that the delivery of  $LO_i$  precedes  $LO_j$  in every admissible sequencing of the course's LOs managed by the LMS. Having automated suggestions on how certain LOs should be necessarily sequenced, in order to preserve dependency relationships, can then be of great help for the instructor, as it can ease a part of the selection and sequencing task, and allow the instructor to focus on less automatable aspects.

## 2 Related Work

An approach for the identification of prerequisite relationships among “knowledge components” is to be found in [12], where causal discovery is used on

components represented as latent (unmeasured) variables. To validate the approach simulated data are used, representing a dataset of student-skills measures. Young *et al.* [13] propose to analyze large-scale assessment to determine the dependency relationships between knowledge units. Given sufficient user data, the authors prove that prerequisites for each instructional unit can be identified. On the contrary, the methodology cannot be applied to new curriculum, that is, units to which student performances have not been extensively evaluated. Recently, Sciarrone *et al.* [8,9] proposed an early attempt to exploit Wikipedia as a source of learning materials. Analyzing the links present in the Wikipedia pages, they build courses based on the Grasha teaching styles and on a social didactic approach. In [2,4] a preliminary attempt for sequencing learning materials has been introduced. An interesting case-based reasoning approach, following a self-directed learning paradigm in assisting users to build sequences of elements out of user-defined libraries, is proposed in [3].

### 3 A Feature-Based Approach for Comparing LOs

Annotation, or tagging, is about attaching names, attributes, categories, comments or descriptions to a text document [1]. It provides additional information (metadata) about an existing piece of data. Among popular annotation tools is Wikipedia Miner [11]. Several hypotheses about the existence of a statistical significant relationships between selected content extracted from two text-based LOs and the potential prerequisite relationships between them have been proposed and validated in [2]. On the basis of these working hypotheses, we propose a feature-based and domain-independent classification approach that automatically identifies those prerequisite relationships without any user effort.

A sketch of the whole process is as follows.

- given the learning objects  $LO_i$  and  $LO_j$ , prospectively retrieved by online repositories or crawled from the web [10], the text content is extracted and analyzed, respectively.
- for each LO the annotation step is operated by the Wikipedia Miner Toolkit, so to pair the LOs with one or more references to Wikipedia pages. Each page belongs to one or more categories  $C_{LO}$  in the Wikipedia Taxonomy (WT). The WT is a ... information; in it Wikipedia pages are ...browsing them, without having to fetch the whole pages.  
The WT is a classification of wiki contents into categories of information: in it Wikipedia pages are enriched with metatags that are updated and perfected by the Wikipedia community. A graph of the categories allows browsing them, without having to fetch the whole pages.
- for each LO, the set of annotations is used to relate the LO to a set of topics; after this step the page is in effect represented by a set of Wikipedia pages, that we call  $T_{LO}$ .



- then we apply certain criteria of evaluation to the sets  $T_{LO_i}$  and  $T_{LO_j}$  representing the topics of  $LO_i$  and  $LO_j$ , respectively.
- we infer the existence of dependency relationships on the basis of a set of features defined according with general observations on the Wikipedia content.

The dependency relation of prerequisite is expressed as  $LO_i \rightarrow LO_j$  meaning that  $LO_i$  is a prerequisite for  $LO_j$ . We introduce the recognition of the opposite relationship, represented by  $LO_i \leftarrow LO_j$  meaning that  $LO_i$  is a prerequisite for  $LO_j$ .

The definition of the features that characterize a LO (in the perspective of the prerequisite relation) is based on the following observations.

1. Typically, a more general topic contains much longer discussion/description than a more specific one, and stating that a topic is more general than another can reflect on the generality/specificity of the respectively represented LOs.
2. If a topic makes reference to other topics, probably the former is more broad and, therefore, general of each one of the referenced set.
3. Topics dealing with multiple concepts should be considered more general than topics containing fewer concepts. The occurrence of concepts can be determined by the nouns occurring in the topic extracted by a Part-of-speech tagger.
4. Considering the number of words in the first paragraph of  $T_1$  and  $T_2$  (the first paragraph is the “description” of the topic), if the former is much greater than the latter, then a relation  $LO_j, LO_i \rightarrow LO_j$  could be inferred.

Basing on these observations we have devised a set of features characterizing relevant aspects of the LOs associated to the topics.

### 3.1 Features of a LO

Given two learning objects  $LO_i$  and  $LO_j$ , the features can be formalized as follows:

1.  $avgLen(LO_i)$ : the average length of the text of the Wikipedia topics associated to  $LO_i$  defined in terms of words obtained by a text tokenization process.
2.  $avgLen(LO_j)$ : similar to  $avgLen(LO_i)$  but evaluated on  $LO_j$ .
3.  $fsl(LO_i)$ : the number of link in the first section of the Wikipedia topics associated with  $LO_i$ .
4.  $fsl(LO_j)$ : similar to  $fsl(LO_i)$  but evaluated on  $LO_j$ .
5.  $avgNL(LO_i)$ : the average number of links in the topics associated to  $LO_i$ .
6.  $avgNL(LO_j)$ : similar to  $avgNL(LO_i)$  but evaluated on  $LO_j$ .
7.  $nouns(LO_i)$ : the number of distinct nouns in  $LO_i$  extracted by a part-of-speech tagger.
8.  $nounsIntersect(LO_i, LO_j)$ : The intersection of the two sets of nouns extracted from  $LO_i$  and  $LO_j$ , respectively.
9.  $avgFsLen(LO_i)$ : the average length of the text of the Wikipedia topics associated to  $LO_i$  defined in terms of words obtained by the tokenization process limited to the first section of the topics.

10.  $avgFsLen(LO_j)$ : similar to  $avgFsLen(LO_i)$  but evaluated on  $LO_j$ .
11.  $intersec(LO_i, LO_j)$ : the intersection between the set of nouns used in links to other topics in the topics associated to  $LO_i$ , and the nouns extracted from  $LO_j$ .

All the features are represented by elements in real or integer domains.

## 4 Experimental Results

In this section, we conducted an experimental evaluation using the Weka (Waikato Environment for Knowledge Analysis) toolkit [5]. Weka is a comprehensive suite of Java class libraries that perform many advanced ML and data mining algorithms.

The test set includes a total of 5 course materials with various levels of difficulty, conveying different random topics (see Table 1), e.g., scientific, archaeological, cinematography and art. For each topic domain, experts manually identified the expected dependencies among LOs with a ratio between the former and the latter varying in the [1.14, 2.27] interval.

The evaluation is performed on the entire pool of LOs making no distinction between courses. The expected dependencies are the relationships between prerequisite and successor concepts represented by LOs. Each LO is represented by a text file containing the entire text of the lesson; the prototype is implemented so as to accept both `html` pages and text documents, automatically retrieved by the network or stored in the local filesystem. Standard lexical analysis is performed in order to filter out `html` formatting elements [6] and tokenize the input stream into tokens.

Two of the most popular ML approaches have been considered in this evaluation: J48 decision tree [5] and JRip propositional rule learner [7].

Due to the size limit of the evaluation dataset, the risk of overfitting the training data, making them somewhat poor predictors is almost non-existent for both of the ML approaches. Decision trees have the advantages to be less sensitive to outliers and nonlinear relationships between parameters.

In the experiments reported here, each approach is validated following a  $k$ -fold cross-validation. A randomly selected portion (one-ten, in this case) of the training data is set aside for validation prior to training. After training on the remaining data, the number of matches and correct predictions over the

**Table 1.** Stats about the test courses.

Course topic	Number of LOs	Expected dependences
Italian Neorealist Cinema	11	16
Programming Languages (Java)	18	41
Lucus Feroniae (guided tour)	7	8
Futurism in art	4	5
Basic Mathematics	4	5

validation set is evaluated. In order to get as much out of the training data as possible, this procedure *training and validation* is repeated 10 times ( $k = 10$ ), once for each of 10 partitions of the training data.

In the classification task, the following measures can be defined:

- $tp$ : the number of identified dependencies that are also expected in the test set;
- $fp$ : the number of dependencies returned by the classifier but missing in the test set;
- $fn$ : the number of expected dependencies that the classifier misses to identify.

and, consequently, the performances can be evaluated with the standard measures of Precision (**Pr**), Recall (**Re**), **F1**-measure and the area under the ROC curve (**AUC**).

The input pattern consisting of the identified attributes' values for  $LO_1$  and  $LO_2$  is classified into one of the following three target classes:

- $c_1$ : set of all pairs  $(LO_i, LO_j)$  for which there is the prerequisite relation  $LO_i \rightarrow LO_j$ ;
- $c_2$ : set of all pairs  $(LO_i, LO_j)$  for which there is the prerequisite relation  $LO_i \leftarrow LO_j$ ;
- $c_3$ : set of all pairs  $(LO_i, LO_j)$  for which there is not any prerequisite relation.

Table 2 shows the obtained performances of the two ML-based classifiers considering also the evaluation for each single target class. The average precision reaches 0.828, proving that the hypothesis of a classifier trained on features extracted from two LOs has the chance to correctly identify prerequisites among them.

At first glance, the precision, recall and F1-measure averages are significantly higher for the J48 classifier, whereas the AUC values are comparable. Basically, while both of the classification models are valid, different performances exist varying the ratio between false positives and true positives, that is, the discrimination threshold.

There is a high variability on all the four measures across the three target classes. As for the precision, J48 obtains higher accuracy for  $c_1$ , JRip on  $c_2$  and  $c_3$ , by contrast. The two classifiers behave quite different on the considered data set, in spite of the k-fold cross validation.

**Table 2.** Obtained Precision, Recall, F1 measure and ROC values for the two considered ML approaches.

	J48				JRip			
	$c_1$	$c_2$	$c_3$	<i>avg</i>	$c_1$	$c_2$	$c_3$	<i>avg</i>
<b>Pr</b>	<b>0.818</b>	0.607	0.95	<b>0.828</b>	0.538	<b>0.727</b>	<b>0.818</b>	0.735
<b>Re</b>	<b>0.621</b>	<b>0.81</b>	0.95	<b>0.811</b>	0.389	0.593	<b>1</b>	0.756
<b>F1</b>	<b>0.706</b>	<b>0.694</b>	<b>0.95</b>	<b>0.812</b>	0.452	0.653	0.9	0.736
<b>AUC</b>	0.722	0.814	<b>0.954</b>	<b>0.846</b>	<b>0.748</b>	<b>0.826</b>	0.889	0.842

Deeper investigation and larger datasets are required for finding out the parameters and ML-based approaches that guarantee good performances across the three classes. Regretfully, there is a scarce availability of public courses with concept maps and prerequisite dependencies.

## 5 Conclusions

Experimental results presented in this article have reinforced the appropriateness of an approach based on the data, so, a ML approach that provides precious indications that strengthen our working hypothesis. Obviously, since this approach is *data driven*, the provided information may be domain dependent.

The amount of inference performed by the classifiers is much greater than standard approaches based on a set of manually defined rules over a predefined set of topics. No hints, predefined taxonomies or similar concepts for each considered domain are provided by a teacher. But of course the chance to reuse the same trained model over different courses and topics lead to less of course sequencing activity burden for instructors, which are able to focus their attention of other tasks, such as assessments and grading strategies or personalized feedbacks to students.

In order to produce results as *independent domain* as possible we aim at exploring alternative approaches of ML and to substantiate the validity of our work hypotheses also theoretically.

## References

1. Biancalana, C., Gasparetti, F., Micarelli, A., Sansonetti, G.: Social semantic query expansion. *ACM Trans. Intell. Syst. Technol.* **4**(4), 60:1–60:43 (2013)
2. Gasparetti, F., Limongelli, C., Sciarrone, F.: Exploiting wikipedia for discovering prerequisite relationships among learning objects. In: 2015 International Conference on Information Technology Based Higher Education and Training (ITHET), pp. 1–6, June 2015
3. Gasparetti, F., Micarelli, A., Sciarrone, F.: A web-based training system for business letter writing. *Knowl. Based Syst.* **22**(4), 287–291 (2009)
4. Gasparetti, F., Limongelli, C., Sciarrone, F.: A content-based approach for supporting teachers in discovering dependency relationships between instructional units in distance learning environments. In: Stephanidis, C. (ed.) *HCI 2015 Posters*. CCIS, vol. 529, pp. 241–246. Springer, Heidelberg (2015)
5. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. *SIGKDD Explor. Newsl.* **11**(1), 10–18 (2009)
6. Kohlschütter, C., Fankhauser, P., Nejd, W.: Boilerplate detection using shallow text features. In: *Proceedings of the Third ACM International Conference on Web Search and Data Mining, WSDM 2010*, pp. 441–450. ACM, New York (2010)
7. Leon, F., Aignatoaiei, B., Zaharia, M.: Performance analysis of algorithms for protein structure classification. In: 20th International Workshop on Database and Expert Systems Application, DEXA 2009, pp. 203–207, August 2009

8. Limongelli, C., Gasparetti, F., Sciarrone, F.: Wiki course builder: a system for retrieving and sequencing didactic materials from wikipedia. In: 2015 International Conference on Information Technology Based Higher Education and Training (ITHET), pp. 1–6, June 2015
9. Limongelli, C., Lombardi, M., Marani, A., Sciarrone, F., Temperini, M.: Concept maps similarity measures for educational applications. In: Micarelli, A., Stamper, J., Panourgia, K. (eds.) ITS 2016. LNCS, vol. 9684, pp. 361–367. Springer, Heidelberg (2016)
10. Micarelli, A., Gasparetti, F.: Adaptive focused crawling. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 231–262. Springer, Heidelberg (2007)
11. Milne, D., Witten, I.H.: An open-source toolkit for mining wikipedia. *Artif. Intell.* **194**, 222–239 (2013)
12. Scheines, R., Silver, E., Goldin, I.: Discovering prerequisite relationships among knowledge components. In: Stamper, J., Pardos, Z., Mavrikis, M., McLaren, B. (eds.) Proceedings of the 7th International Conference on Educational Data Mining, pp. 355–356. ELRA, May 2014
13. Vuong, A., Nixon, T., Towle, B.: A method for finding prerequisites within a curriculum. In: Pechenizkiy, M., Calders, T., Conati, C., Ventura, S., Romero, C., J. Stamper, J. (eds.) The 4th International Conference on Educational Data Mining (EDM 2011), pp. 211–216 (2011)

# Using Electroencephalogram to Track Learner's Reasoning in Serious Games

Ramla Ghali<sup>(✉)</sup>, Claude Frasson, and Sébastien Ouellet

Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, Montréal,  
QC H3C 3J7, Canada  
{ghaliram, frasson}@iro.umontreal.ca, sebouel@gmail.com

**Abstract.** In this paper we present a serious game, Lewispace, where we focus on measuring and using Electroencephalograms in order to detect how the learner reasons in the game. We track learner's reasoning according to different regions of the brain. Four standard lobes were taken into consideration: frontal, parietal, occipital and temporal. Each lobe was measured for each participant. We also studied the lobes measures distribution for all the participants. We found that some regions are more related to learner's vision and reflexion during the game and this could be an indice that the learner follows the correct reasoning process. Primary results show that our game enhance learners' performance. Moreover, the learners use almost occipital lobe to visualize the task presented in the game and the frontal lobe for the reasoning process.

**Keywords:** Serious Games · Electroencephalogram · Reasoning · Brain lobes · Performance

## 1 Introduction

Tracking learner's reasoning in Intelligent Tutoring System (ITS) or Serious Games (SG) is a very challenging task [1, 2]. Furthermore, it is very difficult to detect if the learner's follows a correct reasoning process or not. If the learner reasons correctly that means that he is progressing while interacting with an ITS or a SG. So, he understands the presented content and can achieve the game or the lesson until the end. However, if he has problems in this interaction, it means that he is stacked and he needs may be more help for the presented content. In this case, we think that it is necessary to detect this incorrect reasoning immediately and react adequately according to each case. One way to detect learner's problems of reasoning and/or misunderstanding of an educational content is to use some electrophysiological measures while interacting with Computer Based Environments. From these measures, we can cite as examples EEG: electroencephalogram [3], eye tracking [4, 5], emotions [5, 6] and the states of workload and engagement [7–9].

In this paper, we present a new SG called LewiSpace, which is dedicated to college students for learning a chemistry lesson, and in particular how to construct molecules' Lewis diagrams based on some given rules and instructions. More precisely, we focus in this study on data collected from EEG using Emotiv EPOC headset and their

distribution according the main areas of the brain. We defined mainly four different areas for the brain: frontal, parietal, temporal and occipital.

In this paper, we make the two following hypothesis:

- (1) There is a significant score improvement while interacting with our game. However, learner's performance depends on the difficulty of each mission of our game,
- (2) We can use EEG brain regions to detect how the learner reasons in our game. We suppose also that reasoning in our game follows some common process for all the participants.

## 2 Related Work

To measure learners behaviors, emotions or interactions in ITS or SG, the existing works used mainly some non-intrusive measures such as emotions [6, 7, 12] and eye tracking data [4, 5]. To measure or predict emotions, Ochs et al. [6] had presented the Emotion Recognition Agent (ERA) to exploit the relations between emotions and colors. D'Mello and colleagues [7] integrated non-intrusive affect-sensing technique with Auto Tutor System to classify emotions using facial expressions, gross body movements, and conversational cues. In [12], Elliot and Pekrun developed a model to automatically predict and adapt learners' emotions. Emotions were measured with a self-assessment questionnaire. Moreover, some other works used eye tracking data to detect or predict emotions. For instance, D'Mello and colleagues [4] have used eye tracking data to automatically detect emotions of boredom and disengagement among learners in interactions with a tutoring system. Recently, Jaques and colleagues [5] used also gaze data features in order to predict two main emotions: boredom and curiosity.

Some other works are focused on the use of the electroencephalogram (EEG). These works are interested mainly to measure some mental states to deduce if the learner is in the correct way of learning. EEG could be used to extract some mental states. Above these states we distinguish the states of engagement and workload. Engagement is related to the level of mental vigilance and alertness during the task (high or low states of vigilance). For instance, highly challenging or difficult tasks involve more engagement. Mental workload can also be seen as the mental vigilance and cognitive load in a particular task. For example, Berka and her colleagues used the indexes of workload and engagement within a learning environment to analyze the students' behaviors while acquiring skills during a problem solving session [8]. Pope (1995) developed an EEG index to measure engagement from EEG inputs [13]. Recently, focusing on these two measures, Chaouachi and his team [9] developed a system, Mentor. This system uses some rules in order to maintain students in a positive mental state while learning, and reacts each time on selecting the appropriate next activity to present to the learner.

Instead of using Power Spectral Densities and mental indexes to study EEG, we could say that EEG in medicine could be studied according to different regions or areas due to the difference of its functionalities in each area. In the literature, the brain could be divided into **four** main areas named **lobes**. The four lobes are: (1) Frontal lobe: this area is located in the front of the head and controls several elements such as reasoning, problem solving, behavior, attention, judgement, etc., (2) Parietal lobe: this lobe is

located in the cerebral hemisphere. It focuses on comprehension, (3) Temporal lobe: it controls visual and auditory memories, and (4) Occipital lobe: this lobe is located in the back of the head and responsible for vision [14, 15]. In the following, we will study the possibility of using these lobes to see how learners reason in our game.

### 3 A Brief Description of LewiSpace Game

LewiSpace is a game intended to teach diagrams of Lewis for college students. For a detailed description, the reader is referred to [10, 11].

Our game is a puzzle-game designed using Unity 4.5 (a 3D environment) integrating EEG and Eyetracking sensors data using the Emotiv SDK v2.0 LITE and the Tobii SDK 3.0. In this game, the learner appears as an astronaut exploring a planet's surface. The astronaut falls into a cavern and for surviving he has to accomplish five missions elaborated in an ascending order of difficulty (see Table 1).

**Table 1.** Missions distribution in LewiSpace game

Missions	Molecules to construct
Mission 1	Produce water (H <sub>2</sub> O)
Mission 2	Produce methane gas (CH <sub>4</sub> )
Mission 3	Produce a sulfuric acid (H <sub>2</sub> SO <sub>4</sub> )
Mission 4	Craft a refrigerant (C <sub>2</sub> F <sub>3</sub> Cl)
Mission 5	Refuel the fuel tank with ethanol (C <sub>2</sub> H <sub>6</sub> O)

During all the game, the learner has access to the standard periodic table of atoms and a list of basic rules in order to understand the lesson.

### 4 Experimental Study and Our Approach

In the experiment, 40 students from Montréal University with no prior knowledge were selected to play our game *LewiSpace*. At the first step, the participant was invited to answer a pretest (3 Lewis diagrams of 3 molecules to construct: CO<sub>2</sub>, CCL<sub>2</sub>F<sub>2</sub>, CH<sub>4</sub>). In the second step, the participant was invited to play with *LewiSpace*. EEG data inputs were collected during the game using Emotiv EPOC headset, which is communicating to the computer through Wi-Fi. It only requires a saline solution for conduction. EEG is sampled at a rate of 128 Hz, and 14 channels (AF3, F7, F3, FC5, T7, P7, O1, O2, T8, F4, F8, AF8) could be measured using this device through TestBench according to the 10–20 electrodes placement international system.. In the last step, the participant did a post-test which is at the same difficulty of the pre-test.

After conducting the experiment, we built our approach as follows: first, we filter noise artifacts from EEG using two filters: a low-pass filter and a high-pass filter. Second, we calculated Fast Fourier Transformation from EEG for the 14 channels. Then, we distributed these channels according to lobe regions. We used mainly four lobes' regions, each one being identified by several channels as described below according to Emotiv



Headset classification: Frontal: AF3, F7, F3, FC5, FC6, F4, F8 and AF4; Parietal: P7, P8; Occipital: O1, O2 and Temporal: T7, T8.

After that, we calculated for each participant the mean of FFT EEG inputs according to each region. Then, we calculated for each region the overall average of all the participants. So, the **maximum value** of each region gives us an indication that this region is the most used in our game. We summarize this method in the figure below (Fig. 1).

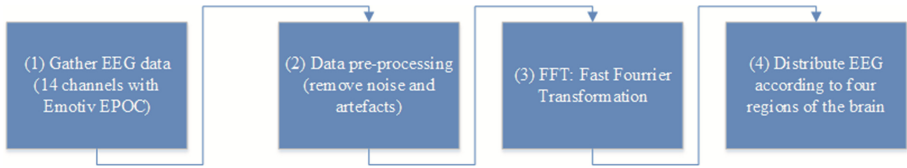


Fig. 1. Our proposed method for EEG regions distribution

## 5 Results and Discussion

### 5.1 Descriptive Results of LewiSpace Game

In this part, we will discuss mainly of learners’ performance in the different missions of our game as well as the improvement achieved while interacting with our game.

For that aim, we define a learner’s score (S) for each mission according to the duration to complete the learner’s session and the number of failures for each mission. This later is defined as follows:

$$S = T + \alpha * nF \tag{1}$$

Where T is the total duration to complete the whole learner’s session,  $\alpha$  is the average duration for each sequence (trial) for all the participants and nF is the number of failures to complete each learner’s mission.

Using this equation, the score is defined in seconds. In order to be clearer, we normalize this score (SN) in order to range between 0 and 100. We use thus this formula:

$$SN = 100 * \left( 1 - \left( \frac{S - m}{M} \right) \right) \tag{2}$$

Where m and M are respectively the minimum and the maximum score collected from all the participants.

In the following figure, we present the distribution of the normalized scores between the participants according to the five missions. From this Fig. 2 below, we noticed that mission 1 and mission 2 have the biggest average which is not surprising for us because these two missions are the **easiest** ones. Whereas, mission 4 and mission 3 have the lowest ones. So we can conclude that these two missions are almost **harder** than mission 5 for people that success to finish our game. We conclude then that **learners’ performance depend on the mission difficulty**.

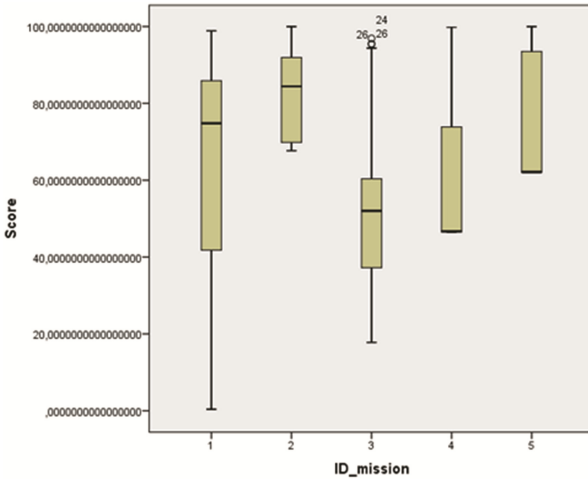


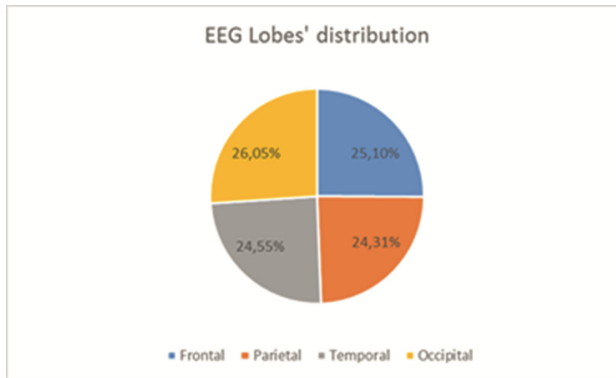
Fig. 2. Distribution of scores according to the game’s missions.

Finally, in order to study the score improvement while interacting with our game, we calculated the effect size as well as the Cohen’s D. We obtained an effect size  $r$  of 0.6222 which is relatively high and a Cohen’s D of 1.59. This last result shows clearly that our game contributes to **improve learning process and enhance learners’ performance** despite its difficulty.

### 5.2 EEG Distribution and Lobes Regions

In this part, we will discuss about EEG channels’ distribution according to four lobes for all the participants using our proposed approach above. We noticed that only 33 participants were considered in this part because of some technical errors while recording EEG. For each participant, we calculated the mean of each region: frontal, parietal, temporal and occipital. Then, to detect the tendency of reasoning in the game, we calculated the overall mean of FFT EEG channels of all participants per lobes (chart 2) (Fig. 3).

From this chart, we can see clearly that occipital lobe followed by frontal lobe are the most used during the game. We can explain this result by the fact that the learner tries to visualize the presented material (Vision lobe: 26.05 %) and therefore he proceeds at **thinking and reasoning** to find an answer and progress in the game (percentage of 25.1 %). These two functions are the main characteristics of frontal lobe. Whereas the parietal lobe is the least used. This could be explained by the fact that it is possible that some students had some comprehension problems.



**Fig. 3.** EEG lobes' distribution in LewiSpace game (Color figure online)

## 6 Conclusion

In this paper we have presented some descriptive results from LewiSpace game. From these results, we noticed that our game enhances learning performance. We showed also that learner's performance depends on mission's difficulty. In this game, we noticed that almost learners use mainly two fundamental lobes: occipital to visualize the information and frontal to concentrate, think and solve problems. We reach a percentage of 25.10 % for occipital lobe and a percentage of 26.05 % for frontal one.

As future work, we plan to build predictable models for learner's need of help according to our findings. Lobes distribution with weights ponderation for each lobe will be used to feed ML techniques.

**Acknowledgements.** We thank the LEADS project and NSERC for funding this work.

## References

1. Treur, J., Wetter, T.: Formal Specification of Complex Reasoning Systems. Ellis Horwood, New York (1993)
2. Selman, B., Kautz, H., McAllester, D.: Ten challenges in propositional reasoning and search. In: IJCAI (1997)
3. Heraz, A., Daouda, T., Frasson, C.: Decision tree for tracking learner's emotional state predicted from his electrical brain activity. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 822–824. Springer, Heidelberg (2008)
4. D'Mello, S., Olney, A., Williams, C., Hays, P.: Gaze tutor: a gaze-reactive intelligent tutoring system. *Int. J. Hum.-Comput. Stud.* **70**(5), 377–398 (2012)
5. Jaques, N., Conati, C., Harley, J.M., Azevedo, R.: Predicting affect from gaze data during interaction with an intelligent tutoring system. In: Trausan-Matu, S., Boyer, K.E., Crosby, M., Panourgia, K. (eds.) ITS 2014. LNCS, vol. 8474, pp. 29–38. Springer, Heidelberg (2014)
6. Ochs, M., Chaffar, S., Abdel Razek, M., Frasson, C.: Using machine-learning techniques to recognize emotions for on-line learning. In: The 18th International FLAIRS Conference, 15–17 May. AAAI Press, Clearwater (2005)

7. D'Mello, S.K., Craig, S.D., Gholson, B., Franklin, S., Picard, R.W., Graesser, A.C.: Integrating affect sensors in an intelligent tutoring system. In: *Proceedings of Affective Interactions: The Computer in the Affective Loop Workshop at International Conference on IUI*, pp. 7–13 (2005)
8. Berka, C., Levendowski, D.J., Ramsey, C.K., Davis, G., Lumicao, M.N., Stanney, K., Reeves, L., Regli, S.H., Tremoulet, P.D., Stibler, K.: Evaluation of an EEG workload model in an Aegis simulation environment. In: *Defense and Security International Society Optics and Photonics*, pp. 90–99 (2005)
9. Chaouachi, M., Jraidi, I., Frasson, C.: Adapting to learners' mental states with a real-time physiologically controlled tutoring system. In: *UMAP 2015*, 29 June–3 July (2015)
10. Ghali, R., Frasson, C., Ouellet, S.: Towards Real Time Detection of Learners' Need of Help in Serious Games. In: *Flairs 2016*, Key Largo, Florida, 16–18 May (2016, to appear). Short paper
11. Ghali, R., Ouellet, S., Frasson, C.: LewiSpace: an exploratory study with a machine learning model in an educational game. *J. Educ. Training Stud.* **3**(6) (2015)
12. Elliot, A.J., Pekrun, R.: Emotion in the hierarchical model of approach-avoidance achievement motivation. *Emot. Educ.* (2007)
13. Pope, A.T., Bogart, E.H., Bartolome, D.S.: Biocybernetic system evaluates indices of operator engagement in automated task. *Biol. Psychol.* **40**(1), 187–195 (1995)
14. Teplan, M.: Fundamental of EEG measurement. *Measur. Sci. Rev.* **2**, 1–11 (2002)
15. Vuillemier, P.: How brains beware: neural mechanisms of emotional attention. *Trends Cogn. Sci.* **9**(12), 585–594 (2005)

# Behavior and Learning of Students Using Worked-Out Examples in a Tutoring System

Nick Green<sup>1</sup>(✉), Barbara Di Eugenio<sup>1</sup>, Rachel Harsley<sup>1</sup>, Davide Fossati<sup>2</sup>,  
and Omar AlZoubi<sup>3</sup>

<sup>1</sup> Computer Science, University of Illinois at Chicago, Chicago, IL, USA  
{ngreen21,bdieugen,rhars12}@uic.edu

<sup>2</sup> Computer Science, Emory University, Atlanta, GA, USA  
davide@fossati.us

<sup>3</sup> Computer Science, Carnegie Mellon University in Qatar, Doha, Qatar  
oalz5092@uni.sydney.edu.au

**Abstract.** *Worked-out examples* have been shown to increase learning gains over problem solving alone. These increases are even greater in novices and those who are learning algorithmic topics, such as those in Computer Science. We have integrated this strategy into our Intelligent Tutoring System and evaluated it on undergraduate students learning the linked list data structure. Although promising, we have identified behavioral differences between high and low gainers - spending less time on an example, and prematurely quitting them led to greater learning.

**Keywords:** Intelligent Tutoring Systems · Worked-out examples

## 1 Introduction

Computer Science (CS) fundamentals are difficult to grasp and require a new way of thinking. Unfortunately, this has an adverse affect on continued enrollment with high levels of attrition in CS courses. High quality, easily accessible educational resources may aid in retention. Intelligent Tutoring Systems (ITS) is an option to provide such a resource.

We have developed an ITS, ChiQat-Tutor (ChiQat), specifically for teaching CS concepts [3]. The ITS teaches fundamentals such as linked lists, binary search trees, and recursion. All lessons are built on a common framework that allows easy integration of new lessons, teaching strategies, and utilities. Each of these lessons support standard problem solving, some with various types of feedback.

We enriched ChiQat with a new teaching strategy; worked-out examples (WOE). WOE provides a step-by-step example of solving a problem from beginning to end. Our prior research touched briefly on student usage of WOE [3]. Participants were from the same student population, i.e. all students were enrolled in the same class - a compulsory second year class in computer programming for CS students. Via visualization of behavior, we observed the following trend: higher gainers tend to quit an example before completion more than lower gainers, and high gainers use the example feature earlier than their counterparts.

Here, we build on this foundational work and look more in depth into the behavioral differences shown by these visualizations. We analyze the data collected for all students who used the WOE feature at any point during the first problem of the linked list tutorial.

## 2 Related Work

Worked-out examples, sometimes referred to as ‘worked examples’, have gained traction since the 1980s. They give students a step-by-step solution to a problem, where they contain: problem formulation, solution steps, and final solution. First coined by Sweller and Cooper [8], WOE were found to be an effective teaching strategy when compared to traditional problem solving during the early stages of development. The advantages of WOE can be explained via *cognitive load theory* [7], in that people require working memory resources to learn. Examples may aid in reducing such resource requirements, thus enhance learning.

Worked-out examples are not a silver bullet, and can be detrimental to learning [2]. Firstly, working memory is not the same for all students - [9] showed that working memory resources decrease with age, thus suggesting WOE are more effective than problem solving for mature learners. Age aside, learning difficulties such as dyslexia [4] may also contribute to deficiencies in working memory. Therefore, younger, cognitively gifted students may not benefit from WOE. Also, there is the possibility *expertise reversal* [5], whereby a student (typically advanced) may perform worse after intervention. McLaren et al. [6] suggests that WOE may not enhance learning, but may increase learning efficiency.

Our prior work [3] includes results of how students used worked-out examples. Visualizations from two experimental groups during the first problem of the linked list lesson, using log data (user actions and their timestamps), suggested that high gainers would use the WOE feature differently from low gainers. More specifically, higher gaining students would tend to quit the WOE prematurely, possibly restarting it later. Low gainers would complete the whole WOE more often. Our conclusion - WOE usage behavior may be an indicator of student learning gains. Here, we explore WOE usage behavior in greater detail.

## 3 Linked List Lesson and Examples

Our experiments focus on the use of worked-out examples in the linked list lesson of ChiQat. Linked lists are a fundamental data structure in computer science, and are typically taught early in a CS undergraduate degree. They are a one-dimensional structure composed of *nodes* that contains a data value, and a pointer to another node. Nodes are linked in sequence to form a list that can be traversed and manipulated. Linked lists have the advantage over an array of offering easier and more efficient ways of manipulating the list, e.g. inserting.

Our linked list lesson aids students in constructing, manipulating, and searching a list. Students are given a problem (out of a total of seven), that gives an initial list and a goal that must be achieved. Such a goal would be to add a node

**Table 1.** Linked list WOE from human tutoring dialogue.

We want to put this new node in after b, and we have to nd b
OK, that's not the b so what you want to do is advance p
And the way you do that is you give p a new value
You give it the next value in this node it points to right now
So we say p equals this next. so it moves p from here to here
That's not the b, so we say p is the next and it moves this over here
So that's how you advance across the list

with value '2' to the end of the list. The student uses Java or C++ commands to manipulate or query the list. The operations available are those traditionally used to manipulate a linked list in these programming languages.

Each problem has an associated on-demand example via the 'Example' button. On execution, the example will graphically play out the three components of a WOE. Our examples were modeled from previously collected human-human tutorial dialogues, which included the topic of linked lists (example in Table 1). One step of the example is given at a time, which includes text from the tutor and the graphical representation of the linked list. The student steps through the example by clicking on the 'OK' button, which will update the interface, until the example concludes. Examples are based on acyclic-directed graphs, where each node in the graph plays out a step, and transitions link these steps together.

Each of the seven problems have an associated WOE that exercises the learning concept of that problem, e.g. for a problem focused on inserting a node would have a different example on inserting a node. Students are given complete control over the usage of examples, allowing them to start, stop, and step forward through one at will. An WOE would typically have around 11–14 steps.

## 4 Experiment Setup

Two sets of experiments were run over a total of eight lab sessions. The two experiments were conducted on different cohorts of the same class. Both cohorts were the same in terms of major, with no distinguishing features between them. Part of this class included material on the linked list data structure. Over the lab sessions, students individually participated in our research by completing a 12 min pre-test, activity, and 12 min post-test. As part of the activity, students were given access to the linked list lesson, and they could use the system for approximately 40 min in any way they wished. There were a total of 55 students who chose to participate in this study. Participants had their usage of the system recorded, such as actions performed and feedback they received from the system, in the form of log data. All tests were anonymized and randomized before being graded by two or three graders.

## 5 Behavioral Analysis

Preliminary visualizations of the log files suggested patterns of WOE usage [3]. Thus, we extracted several features for individual students from the existing log data to see if they correlate with any potential learning gain. The features are:

- **First WOE Completed:** 0 if first viewing is quit by the user, else 1.
- **Completed WOEs:** Number of WOEs that played to completion.
- **Incomplete WOEs:** Number of WOEs that were terminated by the user.
- **Total WOEs:** Total number of WOEs played (Complete + incomplete).
- **Proportion Completed WOEs:** Completed WOEs/total WOEs.
- **Total Duration:** Total duration of all WOEs in seconds.
- **Average WOE Length:** Average length of played WOEs in seconds.
- **Average WOE Steps:** Average number of steps made in a WOE.
- **Standard Deviation Steps Used:** Std dev of steps used over all WOEs.
- **Standard Deviation Duration:** Std dev of all WOE durations in seconds.
- **All Complete WOEs:** 1 if all WOEs played were completed, else 0.
- **All Incomplete WOEs:** 1 if all WOEs played were incomplete, else 0.

In addition to these WOE features we also used the most explanatory feature from our previously collected student-human tutor tutorial dialogues, that being pre-score [1]. These features were calculated and organized from the experiment logs for the first problem. Only students that used a WOE for the first problem were included in this analysis (42 students).

We built logistic regression models for all combinations of these features, correlating them with standard learning gain. Table 2 show the top three regression results (with respect to adjusted- $R^2$ ), with our baseline of only pre-score. WOE features do appear to improve models that correlate with learning. The most significant features are ones related to the type of WOE termination - a positive correlation has been established for not completing WOEs, and negative for completing them. This further reinforces our prior results of students who quit WOEs achieve higher learning gains. Further models were created using normalized learning gain, however, the best model had a far lower adjusted- $R^2$  at just 0.205. The most explanatory features in these models mirrored much of those in the standard learning gain models, however, the total number of WOEs used was also shown to be a significant feature at  $p < 0.05$ .

Since some of these features correlated with learning, we looked into the differences between high and low gaining students. Students were ordered by learning gain and split on the median, giving two unbiased groups - 21 high, and 21 low gainers. This was done for standard and normalized learning gain. Table 3 shows the mean WOE features for students within each of these sets. Statistically significant values, using a paired t-test, are labeled with \* ( $p < 0.05$ ), and values trending toward significance ( $p < 0.1$ ) are indicated in bold.

The difference between groups tend to be more pronounced when using normalized gain, where all statistically significant features were reported. Low gainers would tend to complete all played WOEs more often than high gainers, and



**Table 2.** The most explanatory models with WOE features.

	Predictor	$\beta$	$R^2$	$P$
Model 1	Pre-score	-.518	.212	< .01
Model 2	IncompleteWOEs	.07	.364	= .165
	ProportionCompletedWOEs	.59		< .1
	TotalDuration	-.0004		= .127
	AllCompleteWOEs	-.315		< .1
	AllIncompleteWOEs	.357		< .05
	Pre-score	-.586		< .001
Model 3	TotalWOEs	.046	.366	= .105
	AvgWOELength	-.001		= .134
	AllIncompleteWOEs	.175		< .02
	Pre-score	-.555		< .001
Model 4	IncompleteWOEs	.054	.366	= .279
	ProportionCompletedWOEs	.499		< .1
	AvgWOELength	-.001		= .118
	AllCompleteWOEs	-.258		< .1
	AllIncompleteWOEs	.351		< .05
	Pre-score	-.577		< .001

**Table 3.** Mean student WOE based features from experiments (trending toward significance in bold, significance with \*).

	Low	High	Low (norm.)	High (norm.)
First WOE Completed	.381	.238	<b>.427</b>	<b>.191</b>
Completed WOE	.952	.667	1.0	.619
Incomplete WOE	1.0	1.333	1.095	1.238
Total WOE	1.952	2.0	2.095	1.857
Proportion Comp. WOE	.389	.206	<b>.405</b>	<b>.19</b>
Total Duration	123.052	71.295	<b>133.04</b>	<b>61.303</b>
Avg WOE Length	<b>57.19</b>	<b>27.93</b>	<b>58.385</b>	<b>26.735</b>
Avg WOE Steps	7.421	5.794	<b>8.191*</b>	<b>5.024*</b>
Stdev Steps Used	2.22	1.558	2.474	1.304
Stdev Duration	<b>29.25</b>	<b>12.18</b>	<b>30.445*</b>	<b>10.986*</b>
All Complete WOE	.19	.048	.19	.048
All Incomplete WOE	.429	.667	<b>.381*</b>	<b>.714*</b>
Learning Gain	<b>-.079*</b>	<b>.244*</b>	<b>-.135*</b>	<b>.454*</b>

when looking at students who would never complete a WOE, our results indicate that high gainers would be more likely to never complete a WOE (t-test shows significance). Although high gainers perform significantly fewer steps (from t-test on normalized measure) on average per WOE, 5.024 vs 8.191, the actual time spent in a WOE by a high gainer is about half that of a low gainer. This amounts to a high gainer performing one step, on average, every 5.32 seconds, while low gainers average at 7.12 seconds. This indicates that even though high gainers do fewer steps, they also spend less time on those steps.

## 6 Conclusions and Future Work

The mined features do show interesting data points on how worked-out example usage may be correlated with learning gain. Through both regression modeling and comparison of high/low gainers, there is evidence that students may benefit from using WOE's incrementally, and not to study them fully to the end. Whereas it may be advantageous for an ITS to use WOE's, our results suggest that their usage should either be regulated to enforce good behavior, or use these features in student modeling.

In future work, we will use promising features - shorter, faster, WOE execution - to implement additional strategies in ChiQat to promote learning gains. Currently, students have complete control over all aspects of the WOE (start, stop, and step). We will add a dynamic component to WOE's, whereby an example may intervene if a student appears to be using poor learning practices.

**Acknowledgments.** This publication was made possible by NPRP grant 5-939-1-155 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

## References

1. Di Eugenio, B., Chen, L., Green, N., Fossati, D., AlZoubi, O.: Worked out examples in computer science tutoring. In: Lane, H.C., Yacef, K., Mostow, J., Pavlik, P. (eds.) AIED 2013. LNCS, vol. 7926, pp. 852–855. Springer, Heidelberg (2013)
2. Di Eugenio, B., Green, N., AlZoubi, O., Alizadeh, M., Harsley, R., Fossati, D.: Worked-out example in a computer science intelligent tutoring system. In: 16th Annual Conference on Information Technology Education. Chicago, IL., October 2015
3. Green, N., Di Eugenio, B., Harsley, R., Fossati, D., AlZoubi, O., Alizadeh, M.: Student behavior with worked-out examples in a computer science intelligent tutoring system. In: International Conference on Educational Technologies. Florianopolis, Santa Catarina, Brazil, November 2015
4. Jeffries, S., Everatt, J.: Working memory: its role in dyslexia and other specific learning difficulties. *Dyslexia* **10**(3), 196–214 (2004)
5. Kalyuga, S., Ayres, P., Chandler, P., Sweller, J.: The expertise reversal effect. *Educ. Psychol.* **38**(1), 23–31 (2003)

6. McLaren, B.M., van Gog, T., Ganoë, C., Yaron, D., Karabinos, M.: Worked examples are more efficient for learning than high-assistance instructional software. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M.F. (eds.) AIED 2015. LNCS, vol. 9112, pp. 710–713. Springer, Heidelberg (2015)
7. Miller, G.: The magical number seven, plus or minus two. *Psychol. Rev.* **63**, 81–97 (1956)
8. Sweller, J., Cooper, G.A.: The use of worked examples as a substitute for problem solving in learning algebra. *Cogn. Instruction* **2**(1), 59–89 (1985)
9. Van Gerven, P.W.M., Paas, F., Van Merrinboer, J.J.G., Schmidt, H.G.: Cognitive load theory and aging: effects of worked examples on training efficiency. *Learn. Instruction* **12**(1), 87–105 (2002)

# The Frequency of Tutor Behaviors: A Case Study

Vincent Alevén<sup>(✉)</sup> and Jonathan Sewall

Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, USA  
{aleven, sewall}@cs.cmu.edu

**Abstract.** For cross-pollination between ITS authoring tools, it may be useful to know the prevalence of tutoring behaviors crafted with these tools. As a case study, we analyze the problem units of *Mathtutor*, a web-based intelligent tutor for middle-school mathematics built, as an example-tracing tutor, with the Cognitive Tutor Authoring Tools (CTAT). We focus on tutoring behaviors that are relevant to a wide range of tutoring systems, not just example-tracing tutors, including behaviors not found in VanLehn's (2006) taxonomy of tutor behaviors. Our analysis reveals that several tutor behaviors not typically highlighted in the ITS literature were used extensively, sometimes in unanticipated ways. Others were less prevalent than expected. This novel insight into the prevalence of tutor behaviors may provide practical guidance to ITS authoring tool developers. At a theoretical level, it extends VanLehn's taxonomy of tutor behavior, potentially expanding how the field conceptualizes ITS behavior.

**Keywords:** ITS authoring tools · Behavior of tutoring systems · Authoring data

## 1 Introduction

Versatile, robust, easy-to-use, and easy-to-learn tools for authoring ITSs are an important development [1–3] and may be key to making ITS widespread. In developing an ITS authoring tool, a key question is: What tutor behaviors should the tool support? VanLehn's classic taxonomy of tutor behaviors [4] provides one possible answer. This taxonomy was induced by theoretically analyzing six ITSs. On the other hand, ITS authoring tools may provide a unique practical perspective that may not be fully captured in this taxonomy. This may be so especially if the tool has had a long life and seen widespread use; it may gradually have acquired features aimed at supporting a wide range of tutoring behaviors. If many tutors or tutor units have been built with the tool, we can measure the frequency of key tutor behaviors in these tutors. We present a case study, focusing on the Cognitive Tutor Authoring Tools (CTAT) [1], which support an ITS technology called example-tracing tutors. Over the years, many tutors have been built with CTAT and these tools have been honed and extended based on the needs of these projects. It is thus an interesting question which tutor behaviors are prevalent in CTAT-built tutors. We focus on one such tutor, *Mathtutor*, [5], one of a number of web-based ITS for middle-school mathematics (cf. *ASSISTments* [3] and *Wayang Outpost* [6]). A distinguishing characteristic may be that *Mathtutor* supports more complex problem-solving scenarios.

Our investigation focuses on a set of tutor behaviors commonly found in many ITSs and not specific to example-tracing tutors. It includes some behaviors not found in VanLehn's taxonomy [4]. Some of these behaviors were described in our prior publications [1], but we have not previously undertaken a systematic analysis of their use or frequency, nor are we aware of any other projects reported in the ITS literature that did so. Baker et al. created a taxonomy of tutor features to investigate students' gaming behaviors [7], but this taxonomy was too fine-grained for current purposes, nor did it focus on tutor behavior exclusively.

The work contributes to the ITS literature both at a practical and theoretical level. At a practical level, insight into the prevalence of tutoring behaviors may provide guidance for developers of ITS authoring tools. At a theoretical level, our analysis enriches theoretical accounts of tutor behavior by extending VanLehn's (2006) taxonomy of ITS behaviors.

## 2 Overview of *Mathtutor* and CTAT

*Mathtutor* [5] covers five content strands for mathematics in grades 6 through 8: (1) numbers and operations, (2) algebra, (3) data analysis, (4) geometry and (5) ratios and proportional reasoning. It is a re-implementation, as an example-tracing tutor, of a set of Cognitive Tutors for middle-school mathematics created prior to CTAT's inception. *Mathtutor* offers 65 units, each comprising between 8 and 30 problems for students to solve. So far, *Mathtutor* has been used by 2,215 students, who completed a total of 31,918 problems in 1,258 h of work. *Mathtutor* was built by a team of authors that included professional staff, many student interns, and teachers. A goal was to reproduce the tutor behaviors of the original Cognitive Tutors, adhering to a model of tutoring that is encoded in eight Cognitive Tutor principles [8]. This model prescribes making thinking visible by breaking problems into steps and providing step-level guidance such as next-step hints and feedback.

Example-tracing tutors can be built with CTAT through a combination of end-user programming techniques such as drag-and-drop interface building, programming by demonstration, Excel-like formula writing, and template-based problem generation [1]. An author first decides for which problems to provide tutoring and conducts cognitive task analysis to identify solution steps, common major and minor strategy variations, and common errors (although given that *Mathtutor* is a reimplementation of existing tutor units, this information was instead gleaned from the existing units). She then creates a user interface for each of the targeted problem types, which lays out the steps of the problems. Using CTAT's Behavior Recorder, the author creates a "behavior graph" that defines acceptable solution strategies. An author can generalize a behavior graph in a number of ways, so that it can stand for a wider range of problem-solving behavior than literally just what is recorded in the graph. The author also writes hints and feedback messages. At student run time, the tutor uses the graph to interpret student problem-solving behavior and to provide hints and feedback.

### 3 Analysis of Tutoring Behaviors Supported in *Mathtutor*

Our analysis focuses on the following inner-loop (i.e., within-problem, step-level) tutor behaviors: error-specific feedback, multiple solution paths; dynamic interfaces; accepting complex input, notational variants, and minor step dependencies; input substitution; partial ordering of steps; and optional and repeatable steps. Of these, only error-specific feedback is included in VanLehn's taxonomy [4]. We analyzed the 897 behavior graphs that make up the 65 *Mathtutor* units. We ran awk scripts over the behavior graphs, generating a table with information about 33,950 behavior graph links. We then used Excel PivotTables to compute the statistics reported below.

#### 3.1 Error Feedback Messages

First, we investigated the prevalence of error-specific feedback messages. These messages react to specific student errors and explain for example why the error is an error. We found that error-specific feedback messages are present in 38 out of 65 *Mathtutor* units (58 % of units). Across all tutor units, 21 % of links represent errors (as opposed to correct problem-solving steps). Thus, although error-specific feedback messages are used frequently, it is clear that the *Mathtutor* authors made no attempt to systematically cover the majority of errors. If they had, there would be many more error links than correct action links. In *Mathtutor*, students can rely on on-demand hints, rather than error-specific feedback, if they do not understand how to solve a step. Nonetheless, the high prevalence of error-specific feedback suggests that ITS authoring tools should support them.

#### 3.2 Multiple Solution Paths

Next, we investigated to what degree, in the *Mathtutor* units, the tutor is capable of following students with respect to multiple strategies *within a single problem* [9]. Surprisingly, the ability to support multiple strategies within a given problem is not mentioned in many theoretical accounts of intelligent tutors (e.g., [4]), possibly because it is assumed to be present. Not all ITSs however appear to support multiple strategies or solution paths within a problem, so this ability should not be taken for granted. Example-tracing tutors offer two main ways of authoring tutors that can accept multiple solution strategies within a problem. First, an author can create multiple paths in a behavior graph. For example, in a *Mathtutor* unit dealing with proportional reasoning, the tutor recognizes two major strategies, Equivalent Fractions and Cross Multiplication. These major strategy variations are captured as two separate branches in the behavior graph. Second, as discussed below, an author can use formulas, regular expressions, or numeric ranges to capture minor strategy variations.

Approximately 30 % of *Mathtutor* units have behavior graphs with multiple solution paths. This percentage was lower than expected, especially when one considers that multiple paths were often used to capture notational variants rather than genuinely different strategies. It may be that from a pedagogical perspective, accommodating multiple strategies within a single problem is not always high priority or even desirable.

It is often difficult for students to practice a single strategy to mastery, let alone multiple. Also, even when the goal is for students to learn multiple strategies, the tutor may still need to offer single-strategy problems, to make sure all strategies are practiced. Nonetheless, we recommend that ITS and ITS authoring tools be able to accommodate multiple solution strategies [9].

### 3.3 Dynamically Adjusting the Tutor Interface to the State of Problem Solving

Next, we consider dynamic interfaces, that is, interfaces that change at specific points in the problem-solving process. Using CTAT, authors can create dynamic interfaces without programming, by adding links in the behavior graph that capture “tutor-performed actions” (TPAs) [1]. Dynamic interfaces are used in 35 % of *Mathtutor*’s units, for a variety purposes. Often they are used to manage limited screen real estate, when there is not enough space to accommodate all required interface components simultaneously. Another common use of dynamic interfaces is to reveal the steps in tutor problems gradually, as the student progresses through the problem, rather than displaying all the steps from the start, to enforce an orderly problem-solving process.

### 3.4 Variable Steps, Including Dependencies Among Steps

A fourth category of behaviors comprises variable (or non-literal) steps, which an author can create by attaching formulas, regular expressions, and other matchers to behavior graph links. Formulas were used far more extensively than we anticipated, namely, in 54 out of 65 units (83 % of the units). Their most common use in *Mathtutor* is to capture notational variants of student input. For instance, on steps where students enter an arithmetic expression, a formula is needed to deal with the range of equivalent expressions that students enter. In other tutor units, formulas were used to accept notational variations such as “40” and “40 %.” Formulas were also used to express dependencies among steps. For example, in a unit dealing with proportional reasoning, students compared two proportions (e.g., what is a better deal, buying 12 tickets for \$18 or 20 of the same tickets for \$25?) by first choosing a suitable “comparison number” (e.g., a number of tickets, such as 4) and then scaling the proportions to this comparison number. Formulas were used to capture the multiple options for the comparison number and also to capture how later steps depend on that number.

### 3.5 Input Substitution

Input substitution refers to the behavior in which the tutor replaces student input by a different expression of that input, when the input is accepted as correct. A common use is to replace text typed by the student by a spelling-corrected version, or to replace an arithmetic expression by the value to which it evaluates. The latter form of input substitution makes the cell function as a simple calculator, for example in units in which the student masters arithmetic and the instructional objectives focus on other aspects of mathematics. Input substitution is used in 21 of the 65 *Mathtutor* units (32 % of units). In addition to evaluating arithmetic expression, input substitution was used for

formatting student input (e.g., avoiding many decimals, making sure a percent sign is included, and money notation). The prevalence of input substitution in *Mathtutor* suggests that this functionality is important in a real-world ITS.

### 3.6 Partial Ordering of Steps

In creating a tutor, it is often desirable to constrain the order in which students carry out problem steps, although without necessarily restricting students to a single ordering of steps. In some mathematical procedures, the order of steps matters (e.g., order of operations, or processing columns right-to-left in multi-column addition). At other times, the order of steps does not matter mathematically, but it matters for creating an effective tutor, for example because it can be difficult to give good hints for a step when prior steps to which the hint refers have not been completed. In CTAT, an author can set whether overall the tutor should treat a problem as ordered or unordered. In addition, an author can define groups of links and designate them as ordered or unordered. The tutor only accepts steps that conform to the author-specified ordering constraints. Of the 65 *Mathtutor* units, the authors defined ordered or unordered groups in 40 units. Thus, it is clear that authors often want to define a partial ordering of problem steps.

### 3.7 Optional and Repeatable Steps

In tutored problem solving it is often desirable to make steps optional, meaning that they are not required for completing the problem. Similarly, it is useful to make steps repeatable, meaning that they can be, or have to be, done multiple times within a given tutor problem. In CTAT, authors can create optional and repeatable steps by specifying a lower and upper bound on the number of times a link in a behavior graph can be “traversed” as the student solves the given problem. Optional links are used in 15 units, or 23 % of *Mathtutor* units, repeatable links in only 3 of the 65 units. Optional links are used primarily to provide optional scaffolding within a problem (i.e., extra steps with tutor guidance that may be helpful but not necessary for all students). Sometimes, optional links were used for actions that are mathematically correct but not strictly necessary, such as entering leading or trailing zeros for decimal numbers.

## 4 Conclusion

To the best of our knowledge, this paper is the first that reports on the frequency of tutor behaviors in an ITS. We focus on a set of common inner-loop behaviors including some that are not included in VanLehn’s taxonomy [4] and that are rarely if ever mentioned in theoretical accounts of ITSs. A striking finding is the frequent use of formulas (over 80 % of *Mathtutor* units use them) to capture input variations and (less frequently) dependencies among steps. We also found that dynamic interfaces are used frequently, that great attention is paid in *Mathtutor* to being able to accept notational variations in input and to replace student input with a different expression of it (input substitution). On the other hand, flexibility in following students with respect to multiple problem



solution paths was more rare than expected, even if it is still a highly desirable tutor behavior that ITS authoring tools should support.

The tutor behaviors discussed in this paper are not specific to example-tracing tutors; they are likely to cut across many types of tutors. At the same time, it seems likely that the reported prevalence of these behaviors is somewhat specific to mathematics at the middle-school level. Further, the particular frequencies may be somewhat specific to the tutoring paradigm used, based on Cognitive Tutor principles. It is an interesting question how much variability there is among authors in terms of what tutoring behaviors are used. We do not, however, have data to answer that question.

A limitation of the work is that it involves only a single tutoring system and only a single authoring tool, albeit a comprehensive tutoring system that has seen substantial classroom use, and an authoring tool whose range of tutoring behaviors may be wide and shaped substantially by demands from the field. It will be useful to repeat this type of analysis across many tools and tutor-building projects.

Practically, the work might provide guidance to developers of ITS authoring tools. At a theoretical level, the work elaborates the range of inner loop functionality identified by VanLehn [4], advancing our field's conceptualization of tutor behaviors.

## References

1. Alevan, V., McLaren, B.M., Sewall, J., van Velsen, M., et al.: Example-tracing tutors: intelligent tutor development for non-programmers. *Int. J. Artif. Intell. Educ.* **26**, 224–269 (2016)
2. Mitrović, A., Suraweera, P., Martin, B., Zakharov, K., Milik, N., Holland, J.: Authoring constraint-based tutors in aspire. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) *ITS 2006*. LNCS, vol. 4053, pp. 41–50. Springer, Heidelberg (2006)
3. Razzaq, L., Patvarczki, J., Almeida, S.F., Vartak, M., et al.: The assistment builder: supporting the life cycle of tutoring system content creation. *IEEE Trans. Learn. Technol.* **2**, 157–166 (2009)
4. VanLehn, K.: The behavior of tutoring systems. *Int. J. Artif. Intell. Educ.* **16**, 227–265 (2006)
5. Alevan, V., McLaren, B.M., Sewall, J.: Scaling up programming by demonstration for intelligent tutoring systems development: an open-access web site for middle school mathematics learning. *IEEE Trans. Learn. Technol.* **2**, 64–78 (2009)
6. Arroyo, I., Woolf, B.P., Burleson, W., Muldner, K., et al.: A multimedia adaptive tutoring system for mathematics that addresses cognition, metacognition and affect. *Int. J. Artif. Intell. Educ.* **24**, 387–426 (2014)
7. Baker, R.S., de Carvalho, A., Raspat, J., Alevan, V., et al.: Educational software features that encourage and discourage “gaming the system”. In: Dimitrova, V., Mizoguchi, R., du Boulay, B., Graesser, A. (eds.) *Proceedings of the 14th International Conference on AI*, pp. 475–482. IOS Press, Amsterdam (2009)
8. Koedinger, K.R., Corbett, A.T.: Cognitive tutors: technology bringing learning sciences to the classroom. In: Sawyer, R.K. (ed.) *The Cambridge Handbook of the Learning Sciences*, pp. 61–78. Cambridge University Press, New York (2006)
9. Waalkens, M., Alevan, V., Taatgen, N.: Does supporting multiple student strategies lead to greater learning and motivation? investigating a source of complexity in the architecture of intelligent tutoring systems. *Comput. Educ.* **60**, 159–171 (2013)

# Towards an Effective Affective Tutoring Agent in Specialized Education

Aydée Liza Mondragon<sup>(✉)</sup>, Roger Nkambou, and Pierre Poirier

Université de Québec à Montréal (UQAM), Montréal, Canada

aydeelizamondragon@gmail.com

nkambou@gmail.com

pierre.g.poirier@gmail.com

**Abstract.** This research contributes to the advancement of intelligent tutoring systems by proposing an affective intelligent tutoring system in the field of specialized education. The Integrated Specialized Learning Application (ISLA) helps autistic children manage their emotions by analyzing the learning trace and considering the learner's current performance to respond accordingly to it during a mathematical learning situation. We have conducted an experiment to validate the support provided by Jessie based on our accompaniment model. The results showed significant improvement in learning by the test group.

**Keywords:** Autism · Affective intelligent tutoring systems · Specialized education · Personalized education · Model of accompaniment

## 1 Introduction

Autism spectrum disorder (ASD) is a neurological disorder affecting the way in which the brain processes information. It can affect all aspects of a person's development. Autism is characterized by impairments in learning and communication, in social interaction, imaginative ability as well as in repetitive and restricted patterns of behavior [5]. Studies reveal that individuals with learning disabilities pose a 'complex multi-factor' problem in the educational system [11]. In this paper, we present an affective intelligent tutoring system (ISLA) to overcome the problem of one-on-one intervention with the purpose of helping the ASD learner to calibrate his/her emotions in mathematical learning. The paper is divided into six sections. The first section is the introduction. The second section presents a brief literature review on autism, emotions, and learning. Section three describes ISLA's components. In section four, the results are presented. Finally, the conclusion and the limitations are discussed outlining the contribution of this research.

## 2 Autism, Emotions and Learning

Emotions and learning have been broadly recognized as challenging among individuals diagnosed with autism [9]. The socio-cognitive and behavioral problems experienced by individuals with ASD are considered to stem from the difficulty of understanding

others' mental states [2, 7]. During intervention, one important challenge is due to the difficulty of anticipating and recognizing negative behaviors, consequently calibrating the child's affective state for effective intervention and learning which vary from child to child as these individuals may have profound cognitive deficiencies while others may have IQ scores that are equal to or higher than the typical person [5]. This diversity of profiles causes multiple challenges in terms of methodologies and teaching programs directed towards autistic children. This is the reason why we believe that modeling affect is the proper approach for ISLA to teach mathematics to children with autism.

An Intelligent Tutoring System (ITS) is a computer system designed with the objective of providing instant and customized instruction or feedback to students as effective as one-to-one tutoring [3]. Within the domain of intelligent tutoring systems, [1] points out that the companion agent has the potential of providing students of all ages with information that will help the student to become self-regulated, consequently become independent learners. In [1] they examined the effectiveness of pedagogical agents (PAs') with MetaTutor for training students on self-regulated learning (SRL) processes through prompting and feedback that facilitated learning about the human circulatory system. The next part presents the system overview of the Integrated Specialized Learning Application (ISLA).

### 3 System Overview and Pedagogical Model

In ISLA, the pedagogical agent called Jessie is capable of detecting the affective state of an autistic child in mathematical learning. This is displayed in the user's interface and related to the accompaniment model. The interface provides a three-dimensional view that allows personalizing the interaction of the three core models of ISLA. This is from the domain model point of view (by providing tools to manipulate domain objects), the accompaniment model point of view through Jessie (pedagogical agent), and the learner model point of view using an open-learner modeling approach [4]. The accompaniment model of ISLA implements rules that should be followed by Jessie to help an autistic learner manage his/her emotions based on the learning trace and his/her current performance. This component is drawn from the self-regulated learning theory highlighting the essential role that metacognition plays in self-regulation and learning [10].

The ASD learner must finish a task before moving to the next phase in order to increase the chance to master the prerequisites of the activity at hand. When a right answer is provided, positive reinforcement is used by Jessie, with social rewards and feedback in order to encourage and motivate the learner, such as 'Yes, you did it!', or 'Good Job!'. By contrast, when a wrong answer is given, Jessie can say something like this: 'That was close, nice try!' and it invites the ASD learner with prompting to try again. Furthermore, if the learner needed help, hints were provided based on pedagogical scenarios. ISLA makes use of a personalized individual plan (IIP) [8], which provides guidance and key elements about the curriculum, the pedagogy, and the behavior required from the autistic person. The learner model is made of the cognitive profile and the affective profile of the learner. Both profiles are maintained by the system and the specialized educator during learning activity.

The affective profile selected in this study includes the affects of: disengagement, encouragement, frustration, interest, anxiety, happiness, guidance, and anger because they are considered relevant in autism intervention practices [6].

## 4 The Methodology

The research population consisted of twelve participants diagnosed with high functioning autism spectrum disorders (ASDs), i.e. boys and girls aged from 6 to 12 years old, with the consent of their parents and under the supervision of a specialized educator. Each learning session lasted one hour, in which, a one-on-one structured intervention in mathematical learning was provided. The participants recruited in this study came from private clinics, specializing in autism, as well as from centers for rehabilitation and specialized education related to autism, all located in Montreal, Canada.

We would like to mention that a preliminary study (a fully Wizard of OZ experiment with a specialized educator playing ‘Jessie’) was previously carried out. The results of the preliminary experiment revealed that the performance of the ASD learners, in a mathematical situation with the use of a pedagogical agent providing real-time support, had a positive impact on these participants’ performance. For the main experiment dealing with the prototype, we have developed an interactive game in mathematical learning for the two groups interacting with ISLA. Two versions of the system were created to measure the performance and affective state of each ASD participant. The first version of the interactive game was intended for the six participants without the pedagogical agent Jessie. The other version was used for the test group interacting with the pedagogical agent Jessie.

**Table 1.** Level of competency criteria

Level of competency	Score Quiz	# of questions
1) Beginner	$\leq 50\%$	5
2) Intermediate	$>50$ and $\leq 60\%$	7
3) Advanced	$>60$ and $<85\%$	10
4) Mastery	$>85$ and $\leq 100\%$	<u>8</u>
		30

The measure of performance baseline quiz score was considered according to the level of competency described in Table 1. The quiz was validated by professionals in the field of specialized education related to autism. The raw scores were compiled with Jessie and without Jessie by correcting what the child achieved during the quiz.

## 5 Results

In this research, we have conducted a study using a prototype of ISLA consisting of two experiments that implemented Jessie as a pedagogical agent. The Wizard of Oz experiments involved twelve participants with high functioning autism in which, one

control group of six student interacted without the pedagogical agent Jessie under the supervision of a specialized educator, while the test group of six participants interacted with the pedagogical agent Jessie under the supervision of a specialized educator. The statistical analysis was based on two important hypothesis.

**E1:** The use of a pedagogical agent to provide support and encourage motivation would have a positive impact on the performance of the autistic student in mathematical learning.

**E2:** The use of a pedagogical agent to provide support and encourage motivation would have a positive impact on the affective state of the autistic student in math learning.

## 5.1 Methods

Descriptive statistics summarize all study variables of interest. For categorical variables, we reported counts and percentages whereas for continuous variables we reported medians and inter-quartile range (IQR), because the values did not follow an approximate normal distribution. We compared scores between the group with and without Jessie. All statistical tests of hypothesis were two-sided and performed at the pre-specified level of significance of 5 %. The p-values reported are not adjusted for multiple testing.

**Table 2.** Participants' profile—group without Jessie

Participant #	Diagnosis	Age	Gender
#1	Autism disorder	12	Male
#2	Autism disorder	9	Female
#3	Autism disorder	9	Male
#4	Autism disorder	8	Male
#5	Autism disorder	11	Male
#6	Autism disorder	7	Male

**Table 3.** Participants' profile—group with Jessie

Participant #	Diagnosis	Age	Gender
#7	Autism disorder	9	Female
#8	Autism disorder	6	Male
#9	Autism disorder	7	Male
#10	Autism disorder	8	Male
#11	Asperger syndrome	10	Female
#12	Autism disorder	12	Male

## 5.2 Results Analysis

Participation of each child in each group was allocated randomly. In the group without Jessie, the age of the children ranged from 7 years old to 12 years old. The participants' profile for the group without Jessie is presented in Table 2. In the group with Jessie, the age of the children ranged from 6 years old to 12 years old. The participants' profile for the group with Jessie is illustrated in Table 3.

**5.3 Comparison of Performance Scores (N = 12)**

Table 4 presents the results related to the relationship between support and performance dealing with the score of each participant for both groups with and without Jessie during the mathematical activity. Since competencies are at the heart of the pedagogical model of ISLA, raw scores were corrected to give the ASD participant his/her level of success according to his/her level of competency in addition.

**Table 4.** Scores performance: with and without Jessie (N = 12)

					<b>Wilcoxon Rank Sum test</b>	
	<b>Version</b>	<b>N</b>	<b>Median</b>	<b>IQR</b>	<b>S</b>	<b>p-value</b>
<b>Raw Scores</b>	With	6	50.0	33.3-63.3	42.0	0.70
	Without	6	41.7	23.3-63.3		
<b>Competency Scores</b>	With	6	86.4	83.3-90.9	46.0	0.31
	Without	6	72.0	58.3-86.4		

The raw scores fluctuated from 7 % as being the lowest score to 100 % as being the maximum score. In this group, the median for the raw scores was 41.7 (IQR 23.3–63.3). In the group with Jessie, where participants benefited from its support, all six children were able to complete the quiz according to their level of competency. The raw scores differed from 10 % as being the lowest score to 67 % as being the maximum score. The results indicated a median of 50.0 (IQR 33.3–63.3). For the competency scores, in the group without Jessie, the scores fluctuated from 40 % to 100 %. In this group, the median for the competency scores was 72.0 (IQR 58.3–86.4). In the group with Jessie, the competency scores differed from 60 % to 92 %. The results indicated a median of 86.4 (IQR 83.3–90.9).

**5.4 Comparison of Affective States (N = 12)**

The results indicated that the participants who benefited from the help of the pedagogical agent Jessie were more encouraged with a median of 27.3 (IQR 21.8–30.2), more interested with a median of 62.9 (IQR 37.6–70.4). They showed less negative behavior such as disengagement with a median of 8.6 (IQR 3.8–17.2). They displayed less frustration with a median of 2.8 (IQR 2.2–3.7) whereas without Jessie the level of frustration was greater with a median of 21.9 (IQR 6.8–47.0). They were less anxious with a median of 3.3 (IQR 1.3–7.2) compared to a median of 10.9 (IQR 8.1–13.4) without Jessie, less angry in comparison to 13.5 (IQR 0–33.9) without Jessie.

Table 5 reveals that the support of Jessie to help the autistic child to calibrate his/her emotions during the mathematical activity had a significant difference for the affects of encouragement between the groups (WRS test, S = 57.0, p = 0.002), frustration (WRS test, S = 19.5, p = 0.05), and guidance (WRS test, S = 51.5, p = 0.04). A one-sided WSR test on the affect of anxiety revealed a significant difference between the groups (WRS test, S = 27.0, p = 0.03), with a distribution with higher values for

**Table 5.** Affective states: with and without Jessie

Affective State	With and Without Jessie (N=12)		With and Without Jessie (N=11)	
	Test-Statistic	P-value	Test-Statistic	P-value
Disengagement	29.0	0.13	42.0	0.03
Feedback	41.0	0.81	31.0	0.93
Encouragement	57.0	0.002	15.0	0.004
Frustration	19.5	0.05	19.5	0.10
Interest	45.0	0.39	21.0	0.13
Anxiety	27.0	0.06	40.0	0.08
Guidance	51.5	0.04	20.0	0.08
Joy	39.0	1.00	31.5	0.84
Calmness	33.0	0.45	33.0	0.45
Anger	29.0	0.09	41.0	0.04

the group with Jessie. Similarly, for the affect of anger, a one-sided WSR test revealed a significant difference between the groups (WSR test,  $S = 29.0$ ,  $p = 0.05$ ). The results showed that when the possible outlier was removed from the group without Jessie ( $N = 11$ ), it had a significant difference for the affects of disengagement (WSR test,  $S = 42.0$ ,  $p = 0.03$ ), encouragement (WSR test,  $S = 15.0$ ,  $p = 0.004$ ), and anger with (WSR test,  $S = 51.0$ ,  $p = 0.04$ ). A one-sided WSR test on the affect of frustration revealed a significant difference between the groups (WSR test,  $S = 19.5$ ,  $p = 0.05$ ), with a distribution with higher values for the group with Jessie. Similarly, for anxiety, a one-sided WSR test showed a significant difference between the groups (WSR test,  $S = 40.0$ ,  $p = 0.04$ ), and for guidance, a one-sided WSR test showed a significant difference between the groups (WSR test,  $S = 20.0$ ,  $p = 0.04$ ).

## 6 Conclusion, Limitations and Future Work

In this research, we have conducted a study using a prototype of ISLA that implemented Jessie as a pedagogical agent based on our accompaniment model. The results revealed that the majority of participants in the test group benefited from the personalization and support provided by the pedagogical agent Jessie, which aimed at helping the autistic student become self-regulated by calibrating his/her emotions and encouraging motivation during the mathematical activity. In this group, all children were able to succeed on the quiz according to his/her level of competency. One limitation of this study is that the groups were heterogeneous for the two experiments with and without Jessie in terms of age which varied from 6 years to 12 years old. Also, the level of competency had a limitation, especially in the group without Jessie, one participant scored 100 % on the quiz. Perhaps with a larger group of participants and more time to experiment, we would have the opportunity to perform the baseline and regroup the children according to their level of competency and age. Future research will be dealing with a full implementation of ISLA by reproducing what has been done according to the prototype experiments. A larger group of participants with autism will be interacting with the pedagogical agent Jessie, in which, the behavior of the pedagogical agent Jessie will be programmed by providing real-time support to help calibrate the affective state of the ASD learner. Children will be grouped according to different criteria like age and competency level. They will be interacting with ISLA until the mastery level is achieved.

## References

1. Azevedo, R.: Theoretical, methodological, and analytical challenges in the research of metacognition and self-regulation: a commentary. *Metacognition Learn.* **4**, 87–95 (2009)
2. Baron-Cohen, S., Leslie, A.M., Frith, U.: Does the autistic child have a theory of mind? *Cognition* **21**(1), 37–46 (1985)
3. Bloom, B.S.: The 2 sigma problem: the search for methods of group instruction as effective as one-to-one tutoring. *Educ. Res.* **13**(6), 4–16 (1984)
4. Bull, S., Kay, J.: Open learner models. In: Nkambou, R., Bourdeau, J., Mizoguchi, R. (eds.) *Advances in Intelligent Tutoring Systems*. SCI, vol. 308, pp. 301–322. Springer, Heidelberg (2010)
5. *Diagnostic and Statistical Manual of Mental Disorders: DSM-IV*. American Psychiatric Association, Washington (2000)
6. Dautenhahn, K., Werry, I.: Towards interactive robots in autism therapy: background, motivation and challenges. *Pragmatics Cogn.* **12**(1), 1–35 (2004)
7. Frith, U.: Interacting minds – biological basis review. *Science* **286**, 1692–1695 (1999)
8. Ministère de l'Éducation, de l'Enseignement supérieur et de la Recherche. [www.education.gouv.qc.ca](http://www.education.gouv.qc.ca)
9. National Research Council: *Educating Children with Autism*. National Academy Press, Washington, DC (2001)
10. Schraw, G.: Promoting general metacognitive awareness. *Instr. Sci.* **26**(1), 113–125 (1998)
11. Woolf, B., Arroyo L., Muldner, K., Bursleson, W., Cooper, D., Dolan, R., Christopherson, R.: The effect of motivational learning companions on low high achieving students and students with learning disabilities. In: *International Conference on Intelligent Tutoring Systems*, Pittsburgh (2010)



# Embedding Intelligent Tutoring Systems in MOOCs and e-Learning Platforms

Vincent Aleven<sup>(✉)</sup>, Jonathan Sewall, Octav Popescu, Michael Ringenberg, Martin van Velsen, and Sandra Demi

Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, USA  
aleven@cs.cmu.edu

**Abstract.** Intelligent tutoring systems (ITS) and MOOCs tend to have complementary pedagogical approaches, but their combination is rarely (if ever) seen. A key obstacle may be technical integration. We present a generalizable case study of extending ITS authoring technology to make tutors easily embeddable into a variety of MOOC/e-learning platforms and run on a range of web-enabled devices. We enhanced the domain-independent Cognitive Tutor Authoring Tools (CTAT) to enable integration of CTAT tutors into multiple environments. A salient lesson learned is that use of widely-used web-based technologies (HTML and JavaScript) may be a major factor in ITS uptake. Also, we found that embedding tutors into existing LMS is challenging, but environment-specific changes can be isolated in a generalizable manner.

**Keywords:** ITS · MOOCs · HTML · Javascript · Cross-platform interoperability

## 1 Introduction

Although intelligent tutoring systems (ITS) are rarely used within MOOCs, they could be a useful addition, considering that the pedagogical approaches used in ITS and MOOCs are largely complementary. MOOCs support many forms of instruction, including video lectures, reading with conceptual questions, discussion boards, and various forms of learning-by-doing with automated or peer feedback. ITS on the other hand offer opportunities for adaptive, guided practice in solving complex problems. However, the embedding of ITS in MOOCs or online courses is rare. A prime challenge is creating a ready, repeatable path from ITS authoring to at-scale deployment in a variety of platforms. Upping the challenge is the plethora of e-learning platforms and web-based devices, as well as the fact that efforts towards standardization have not yielded a single overarching standard and are not geared towards tutors.

To address these challenges, we have been working to make it possible to embed tutors built with the Cognitive Tutor Authoring Tools (CTAT) [1] in a variety of e-learning platforms. We previously integrated CTAT-built tutors into two versions of an edX MOOC [2]. Here we report on more recent work in which we (a) have made CTAT tutors runnable on all the devices that support popular web browsers, by supporting HTML as tutor interface technology, and (b) are extending CTAT so that a CTAT-built tutor can run unchanged on a variety of LMS platforms. We illustrate our solutions with

examples of tutors embedded in MOOCs and online courses. The approach taken and the experience gained may be useful for developers of other ITS authoring tools. As such, this work can help in making ITS more widely used.

## 2 HTML-Based User Interface Components

As one step in our strategy of integrating ITS technology with MOOCs and e-learning platforms, we reimplemented the CTAT front-end technology so that authors can build tutor interfaces in HTML, CSS and JavaScript. This change adds a third interface option for CTAT tutors, in addition to Java and Flash/ActionScript. It moves CTAT to a more popular and free web development environment with many benefits:

- open to inspection by (and hence less suspicious to) content filters and other network security gear common on school networks;
- not proprietary or controlled by a single vendor;
- easily learned, with many free resources available for learning HTML;
- extensible, with many free libraries available for adding features;
- stronger support for accessibility tools, which most often read HTML;
- well-supported by free programming tools (JSHint, Google Closure, etc.).

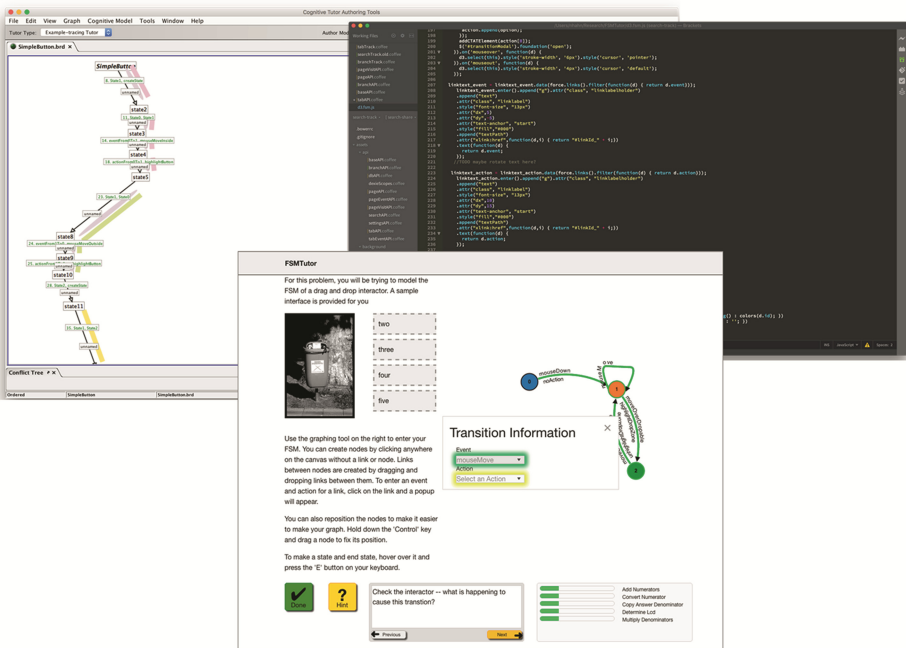


Fig. 1. Tutor for finite state machines using D3 by Nathan Hahn and Pasan Julsakrisakul.

Our reimplementaion of the CTAT tutor interface components adheres to several principles. First, in the given UI framework (here, HTML), we attempt to make tutoring-specific programming follow practices already customary in that framework and allow the author to take full advantage of the framework’s capabilities. Specifically, an author can build a tutor interface with CTAT-enabled interface components using ordinary HTML coding techniques. Further, we permit users to implement their own CTAT-enabled components, as authors did in the examples below. We preserve authors’ ability to display any non-interactive or untutored material.

Second, we separated the visual styling from the interface components themselves. In HTML, all visual styling should be done in CSS. Hence authors can use CTAT’s default CSS style sheet or create their own. Uniform changes (e.g., to the color used for flagging steps as incorrect) can be made easily across all components, and any single interface can be given a different style without internal changes.

Third, as a guiding software-architecture principle, as much as possible, we maintained a strict tool-tutor separation [3], which underlies CTAT and Cognitive Tutors and has many advantages. This principle mandates that the tutor backend (the “tutor”) do all the tutoring (and nothing but tutoring) while the interface (the “tool”) is responsible for interactions with the user but not tutoring. The tool-tutor architecture entails an explicit messaging protocol [3]. In our new HTML tutor interface implementation, we enforce adherence to this protocol by serializing communications between the interface and tutor backend into messages passed over a single software interface.

A number of projects have taken advantage of the new CTAT HTML tutor interfaces, including three prototype tutors and one now used in an online statistics course,

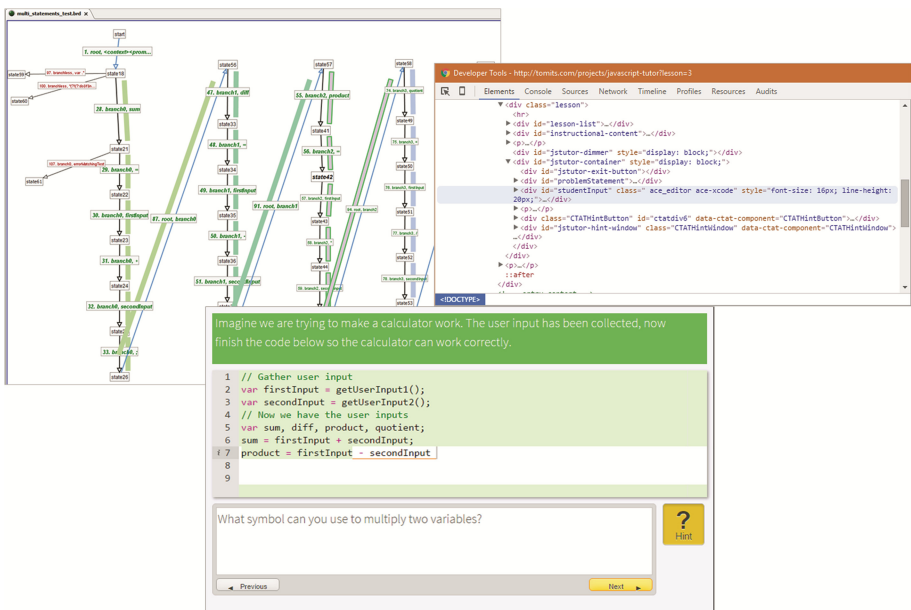


Fig. 2. CTAT tutor to teach JavaScript by Tomit Huynh, with the Ace JavaScript editor.

described below. Interestingly, in two of these prototypes (see Figs. 1 and 2), the authors created their own new tutor-enabled interface components, building on CTAT plus the free, off-the-shelf JavaScript library D3 [4] and the Ace editor [5], respectively. Generally, we have been pleasantly surprised by the enthusiasm that prospective tutor authors have expressed for the HTML version of CTAT interfaces, reflecting, no doubt, the greater popularity of HTML over Flash. The main downside is that at least temporarily, we give up drag-and-drop interface building, meaning we move out of the non-programmer ITS authoring paradigm. We are working to avoid this tradeoff.

### 3 Compatibility with Multiple Deployment Environments

In a second line of work, we pursued interoperability with a variety of MOOCs and LMSs. A key goal was to enable a CTAT tutor to run, without changes to the author's work, in multiple LMS environments. This integration requires:

1. providing means in the LMS to serve or invoke the tutor, via a URL to the tutor's HTML page or a reference to software objects that generate that page;
2. providing access to all runtime files, including images, style sheets, script libraries and data files;
3. providing access to the tutor backend, e.g., its inner loop and outer loop processors, in ITS architectures (such as CTAT) in which they are separate;
4. sharing the permanent student model between the LMS and the tutor and allowing updates;
5. supporting the resumption of a partially-completed problem, so that a student's partial work on a problem can be saved and restored in a later session;
6. supporting instructor review of student work done in the tutor;
7. passing grades and other performance metrics to the LMS (e.g., for use in the LMS's grade book or teacher dashboard).

As we started work on addressing these requirements for CTAT tutors in multiple platforms, we soon saw the value in masking the platforms' differences from the tutor itself, to preserve a modular design. The principal software component of our strategy was therefore to implement, in JavaScript, a layer of "insulating software" between the LMS environment and the CTAT tutor. Its functions were to (a) detect the runtime environment, (b) extract from the environment the runtime information the tutor needed, and (c) provide that runtime information to the tutor via a fixed API.

Over recent years we have achieved a number of different forms of CTAT/LMS integration; the work has revealed pros and cons of each. Our initial attempt at making CTAT tutors deployable across a range of LMSs targeted the Shareable Content Object Reference Model (SCORM) standard [6]. We implemented SCORM 1.2 compatibility so that CTAT tutors could be used in Moodle, Blackboard and other LMSs supporting SCORM. We demonstrated this form of integration in Moodle, but we then found that the LMSs used by many MOOCs do not support SCORM objects.

We next implemented the LTI Tool Provider interface [7], which is supported by many LMSs (e.g., edX, Coursera, Canvas, Blackboard, Moodle, OLI). We demonstrated this

integration by embedding CTAT tutors into the edX courses “Data, Analytics and Learning” and “Big Data in Education,” our first integration of CTAT tutors in MOOCs [2]. The tutors were hosted on our own TutorShop, an LMS that is geared toward tutors and is compatible with CTAT. They were not hosted on edX, for LTI requires external hosting of embedded content. Although this solution worked for us, a key downside is that the tutor must be served from its own host; therefore, the LTI Tool Provider (here, the tutor developer), must maintain server machines scaled to handle however many students might enroll in the course.

Finally, we achieved custom integrations with two additional platforms, edX and the Open Learning Initiative (OLI). In both of these integrations, the tutor is hosted by the LMS itself, addressing a key problem with LTI, but the costs of custom integration are significant: the programming is challenging, and unlike integration to standards, the result covers only one LMS at a time. First, we embedded a CTAT tutor (a reimplementation of an existing non-CTAT tutor) in the Open Learning Initiative’s Probability and Statistics and Statistical Reasoning courses [8] (see Fig. 3). This tutor is the first CTAT tutor with an HTML interface that has seen use in real educational settings. The

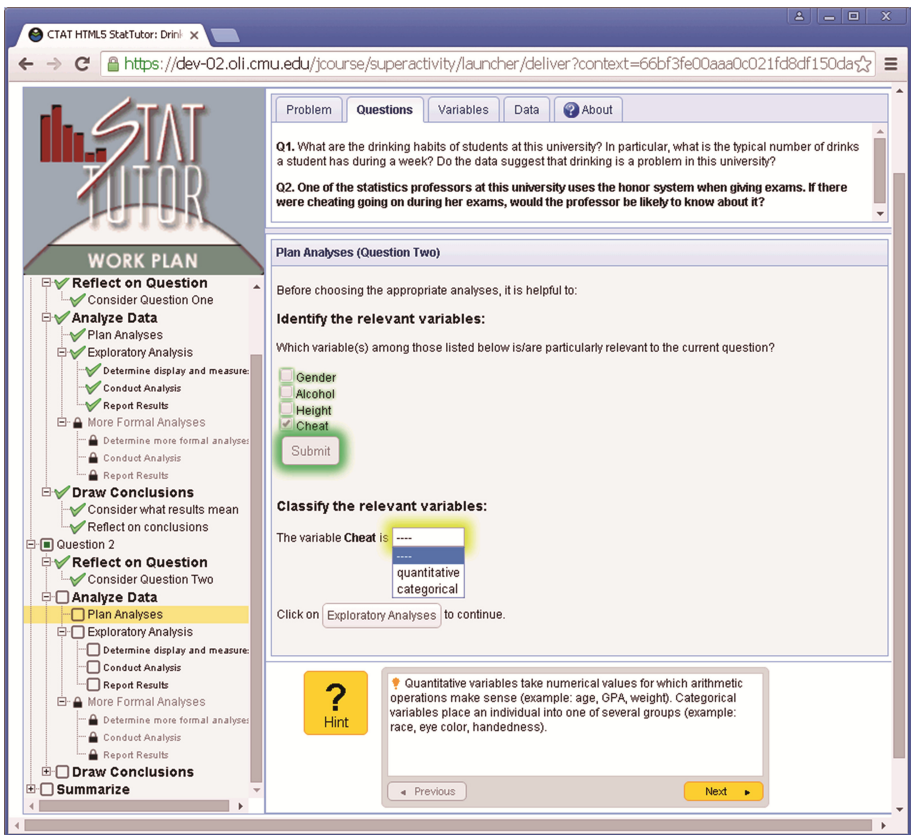


Fig. 3. CTAT StatTutor in OLI Statistics course.

insulation layer was highly useful. We also embedded the revised StatTutor into an edX MOOC via edX's native XBlock interface [9]. The project prompted us to significantly broaden our insulation software, but it also highlighted the need for server-side coding (i.e., extensions to Open edX) so that the dynamically-generated HTML and data could deal with ITS as a new content type for XBlock.

Of the seven integration requirements listed above, most could be met by client-side insulation software plus some server-side code. There is more work to be done (a) to fully share a student model between ITS and MOOC, for example, so that adaptive decisions in the LMS can depend on student performance in the tutor, and vice versa, and (b) to take advantage of the tutor's own adaptive outer-loop capabilities (e.g., individualized problem selection).

## 4 Conclusion

As two parts of our multi-pronged effort towards making ITS widely deployable, we extended CTAT so it supports HTML tutor interfaces, and we integrated CTAT tutors with a variety of MOOC and e-learning platforms. We demonstrated these advances with tutors embedded in MOOCs or online courses in a number of projects.

We see considerable advantages to using HTML for building the front end of web-based tutors: it is free and brings a large community of expertise, tools and libraries. The enthusiasm for building tutors in HTML has been good. A temporary downside is that we give up drag-and-drop interface building, but this downside is likely to disappear. We distill some general principles that may apply in other projects that focus on creating tutor interface technology. First, we try to make ITS interface authoring no different from mainstream HTML authoring, to take advantage of existing tools, libraries, and tutorials. Second, we separate the visual styling from the interface components themselves. Third, we continue to adhere to tool/tutor separation. Although HTML is widely used in e-learning (e.g., <http://elearningindustry.com/the-ultimate-list-of-html5-elearning-authoring-tools>), and may have been used as ITS front end technology, we are not aware of any papers that discuss issues related to the use of HTML for building tutor interfaces.

A second prong in our work is to make CTAT tutors compatible with multiple MOOC platforms and LMSs in a way that does not require platform-specific authoring steps, so that the same tutor can be deployed, without changes, in different environments. A key issue is that no single integration serves a wide range of needs: before we even finished our OLI integration our Statistics users asked for edX. Thus, we were left to pursue multiple integration options. Our technical approach is to insulate the tutor from the details of different environments and (ironically) from different e-learning standards, which turned out to be helpful. Although we demonstrate our approach in CTAT, the seven integration requirements identified above, and the general approach of an insulation layer can be expected to generalize, even if details differ.

The work represents an important and generalizable step toward bringing ITSs into a wide range of deployment environments, which may help spread ITS technology and

promote tutoring at scale. It may open up opportunities to pursue new research questions regarding complementary pedagogies and adaptivity in MOOCs.

## References

1. Alevan, V., McLaren, B.M., Sewall, J., van Velsen, M., et al.: Example-tracing tutors: intelligent tutor development for non-programmers. *Int. J. Artif. Intell. Educ.* **26**, 224–269 (2016)
2. Alevan, V., Sewall, J., Popescu, O., Xhakaj, F., Chand, D., Baker, R., Wang, Y., Siemens, G., Rosé, C., Gasevic, D.: The beginning of a beautiful friendship? Intelligent tutoring systems and MOOCs. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M. (eds.) *AIED 2015. LNCS*, vol. 9112, pp. 525–528. Springer, Heidelberg (2015)
3. Ritter, S., Koedinger, K.R.: An architecture for plug-in tutor agents. *Int. J. Artif. Intell. Educ.* **7**, 315–347 (1996)
4. <http://d3js.org/>
5. <https://ace.c9.io/>
6. Advance d Distributed Learning. <http://adlnet.gov/adl-research/scorm/>
7. IMS Global Learning Consortium. <http://www.imsglobal.org/>
8. <http://oli.cmu.edu/>
9. <http://edx.readthedocs.org/projects/xblock/en/latest/>

# Using Cloze Procedure Questions in Worked Examples in a Programming Tutor

Amruth N. Kumar<sup>(✉)</sup>

Ramapo College of New Jersey,  
505 Ramapo Valley Road, Mahwah, NJ 07430, USA  
amruth@ramapo.edu

**Abstract.** In order to increase the engagement of learners, we incorporated cloze procedure questions into the worked-example-style feedback provided by problem-solving tutors currently used by introductory programming students unsupervised. We conducted a multi-institution controlled study to evaluate the effectiveness of this intervention from fall 2012 through spring 2014. The results of the study were mixed. We found that when students had to answer cloze procedure questions embedded in the feedback, they did spend significantly more time per problem and they learned concepts with significantly fewer practice problems. However, they did not learn significantly more concepts and their change in score from pretest to post-test was not any different on the learned concepts from that of control group. Finally, the increased time on task due to the intervention may benefit different demographic subgroups differently.

**Keywords:** Cloze procedure questions · Worked example · Programming tutor · Evaluation

## 1 Introduction

A typical software tutor provides feedback designed to help students learn from mistakes. But when students use a tutor after hours, unsupervised, on their own time, and as part of a class assignment for which they get completion credit, but not necessarily credit for improvement in learning, (conditions henceforth referred to as *in-natura*), how can we encourage students to read the feedback? How can we ensure that they indeed read the feedback?

One mechanism is to have the students answer questions embedded within the feedback. If students are required to answer these questions before moving on to the next problem, and furthermore, required to answer them correctly, they would have to read the feedback to understand the context before answering the questions. This is the spirit in which we incorporated questions into the worked-example-style feedback [1] provided by the problem-solving tutors we have developed for programming concepts, that are typically used *in-natura* by dozens of schools each semester. The worked-example-style feedback itself has been shown to improve learning in our tutors in prior evaluations [2].



The results of earlier studies where learners were prompted to answer questions embedded in the feedback are inconclusive. One study reported enhanced transfer of solution methods due to the use of embedded questions [3]. However, another study found no difference between complete and incomplete examples [4]. In a more recent study, “active” example walkthrough was found to lead to better learning in some cases [5]. However, a follow-up study by the same authors failed to replicate this result [6]. We conducted a multi-institution, multi-semester study to evaluate whether embedding questions in the feedback, and thereby, increasing the time-spent-on-task actually led to improved learning. We present the results of the study in this paper.

## 2 The Study

**Participants.** We conducted a controlled study over four semesters: fall 2012 - spring 2014. The participants of the study were students of introductory programming from 41 institutions that used the tutor during those four semesters. Each semester, the schools were randomly assigned to control or experimental group. Over the four semesters, a total of 1154 students participated in the control group and 954 students in the experimental group.

**Instrument.** The tutor we used for the study (problets.org) covered concepts of selection statements in C++/Java/C# programming languages. In the tutor, the student is presented a program containing one or more selection statements and asked to identify its output, one output at a time. After the student has submitted the answer, if the answer is incorrect, the tutor presents the same problem as a worked example, complete with step-by-step explanation of the execution of the program that justifies the correct answer. The tutor covers 12 concepts on one-way and two-way selection statements. It is accessible over the web - students can use it on their own time, as often as they please.

For the purposes of this study, we embedded cloze procedure questions within the step-by-step explanation provided as feedback. A cloze procedure question is a question in which words are omitted from a text, and students are asked to fill in the blanks. It is popular in reading comprehension tasks.

In our tutor, each cloze procedure question was presented as question marks within the text of the feedback. In order to answer the question, the student clicked on the question marks, whereupon, the answering options appeared in a drop-down list. If the student selected an incorrect option from the list, the tutor noted that the answer was incorrect and asked the student to try again. The student was allowed as many attempts as necessary to identify the correct answer, but could not proceed to the next cloze procedure question until the current question had been answered correctly. In addition, the student could not proceed to the next problem until the student had answered all the cloze procedure questions embedded in the feedback of the current problem correctly.

For this study, the feedback provided to the student when the student solved a problem incorrectly differed between control and experimental groups:

the feedback presented to experimental group had three embedded cloze procedure questions, whereas that presented to control group had none.

**Protocol.** The tutor administered pretest-practice-post-test protocol as follows:

*Pretest.* During pretest, the tutor presented one problem per concept to prime the student model. If a student solved a problem correctly, no step-by-step explanation was provided to the student, and no more problems were presented to the student on the concept. On the other hand, if the student solved a problem partially, incorrectly, or opted to skip the problem because the student did not know the answer, worked-example-style feedback was presented to the student and additional problems on the concept were scheduled to be presented during practice stage.

*Adaptive practice.* Once a student had solved all the pretest problems, practice problems were presented to the student on only the concepts on which the student had solved problems incorrectly during pretest. For each such concept, the problems were presented, two at a time per concept, in the same order of concepts as on the pretest, until the student had mastered the concept, i.e., had solved at least 60% of the problems on the concept correctly. After each incorrectly solved problem, the student was presented worked-example-style feedback.

*Adaptive post-test.* During this stage, which was interleaved with practice, the student was presented a test problem each on the concepts that the student had mastered during practice.

*Demographics.* Students were given the option to identify their sex, race and major.

The entire protocol was administered back-to-back, entirely over the web. It was limited to 30 min for the control group (a duration considered reasonable for online assignments in introductory courses) and 40 min for the experimental group in order to account for the time needed to answer cloze procedure questions.

A concept was considered to have been learned during this session if the student solved the problem on that concept incorrectly during the pretest, solved enough problems during adaptive practice to master the concept, and proceeded to solve the problem on that concept correctly during post-test.

**Design.** In this study, *treatment* refers to whether or not cloze procedure questions were embedded within the feedback. We considered the following dependent variables:

- Pretest score per problem as a measure of prior preparedness, used to verify that the control and experimental groups were comparable;
- The number of concepts learned as a direct measure of the amount of learning;
- The number of practice problems solved per learned concept as an inverse measure of the pace of learning;
- Pre-post change in score on learned concepts as a direct measure of improvement in learning;

- The time spent per pretest problem to assess the impact of treatment on the pace of solving problems.

We considered the following independent variables other than treatment:

- Sex, as identified by the student: male or female.
- Representation, based on the race identified by the student: Caucasians and Asians were grouped as traditionally represented students in Computer Science, and the rest of the races were grouped as underrepresented students.
- Semester when the student attempted the tutor: Fall 2012-Spring 2014.

**Data Collection and Analysis.** Students could use the tutor as often as they pleased. If a student used the tutor multiple times, we considered only the first attempt when the student had solved all the pretest problems. If the student never solved all the pretest problems, we considered the attempt when the student had solved the most number of pretest problems.

Since worked-example-style feedback and therefore, cloze procedure questions were only presented if the student solved a problem incorrectly, we eliminated all the students who had solved all the pretest problems correctly, as well as all the students who did not solve any practice (and therefore, post-test) problem. Finally, unlike in a prior study [7], we considered only undergraduate students. This left us with 359 students in control group and 253 students in experimental group.

A typical program produces a sequence of outputs. The tutor awarded grade for each problem as: (number of outputs correctly identified in proper sequence - number of incorrect outputs identified)/total number of outputs in the problem. Therefore, the score on each problem was normalized to  $0 \rightarrow 1.0$  regardless of the number of outputs in the program. We analyzed each dependent variable using univariate ANOVA, with treatment, sex, representation and semester as fixed factors.

### 3 Results

**Pretest Score per Problem.** When we analyzed the pretest score per problem, we did not find a significant main effect of treatment [ $F(1,611) = 0.029$ ,  $p = 0.864$ ]: control and experimental groups were comparable in their prior preparation. Unfortunately, we found a significant difference between the sexes [ $F(1,611) = 6.326$ ,  $p = 0.012$ ]: male students scored significantly higher ( $0.829 \pm 0.021$ ) than female students ( $0.785 \pm 0.027$  points per problem). We also found a significant difference between the two types of racial groups [ $F(1,611) = 4.898$ ,  $p = 0.027$ ]: - traditionally represented students scored significantly higher ( $0.826 \pm 0.015$ ) than underrepresented groups ( $0.788 \pm 0.031$  points per problem). The difference between semesters was significant [ $F(1,611) = 3.48$ ,  $p = 0.016$ ] suggesting that students from different semesters were not comparable. The interaction between treatment and semester was also significant

[ $F(3,611) = 3.010, p = 0.03$ ]: students in the experimental group scored higher than those in control group in fall semesters and vice versa in spring semesters.

**Pretest time per problem.** The cloze procedure questions indeed compelled experimental group students to spend significantly more time per pretest problem than control group [ $F(1,611) = 7.003, p = 0.008$ ]: experimental group spent a mean of  $86.31 \pm 7.44$  s versus  $72.225 \pm 7.344$  s for control group. Note that the above are means for the entire pretest, which includes both problems solved correctly (for which neither group received any feedback) and incorrectly (for which both groups received feedback, but experimental group had to answer three embedded cloze procedure questions). Underrepresented students, who had scored less than traditional students on the pretest, also spent more time per pretest problem than traditional students [ $F(1,611) = 5.687, p = 0.017$ ]:  $85.614 \pm 9.385$  s for underrepresented students versus  $72.921 \pm 4.603$  s for traditionally represented students. On the other hand, even though female students had lower pretest scores than male students, they spent marginally *less* time per pretest problem than male students [ $F(1,611) = 3.256, p = 0.072$ ]:  $74.465 \pm 8.249$  s for female students versus  $84.07 \pm 6.421$  s for male students.

**Concepts learned.** When we analyzed the concepts learned, we did not find a significant main effect of treatment [ $F(1,586) = 0.129, p = 0.72$ ]: there was no difference in the learning with versus without embedded cloze procedure questions. So, the treatment did not seem to increase learning overall. Female students learned marginally more concepts ( $1.958 \pm 0.24$ ) than male students ( $1.689 \pm 0.169$ ) [ $F(1,586) = 3.225, p = 0.073$ ], but the interaction between treatment and sex was not significant. Once again, we found significant difference between semesters [ $F(3,586) = 3.056, p = 0.028$ ]: experimental group learned significantly more in spring 2013 ( $1.974 \pm 0.51$  concepts) compared to control group ( $1.523 \pm 0.588$ ) whereas in the other three semesters, there was minimal difference in the concepts learned by control and experimental groups.

**Practice problems solved per concept learned.** There was a significant difference in the number of practice problems solved per learned concept between the two groups [ $F(1,586) = 4.462, p = 0.035$ ]: experimental group students solved  $2.804 \pm 0.232$  problems per learned concept whereas control group students solved  $3.178 \pm 0.259$  problems. So, students learned concepts with fewer practice problems when they had to answer cloze procedure questions, which is a positive result. Once again, there was marginally significant main effect for semester [ $F(1,586) = 2.264, p = 0.08$ ] and the interaction between treatment and semester was significant [ $F(3,586) = 3.580, p = 0.014$ ]. Experimental students learned with significantly fewer problems in spring 2013 ( $2.785 \pm 0.603$  problems per learned concept versus  $4.151 \pm 0.696$  for control group). In Fall 2013, the difference between the two groups was negligible. In Fall 2012, experimental group students learned with fewer problems than control group and vice versa in spring 2014.

**Pre-post change in score for learned concepts.** We analyzed pretest and post-test scores on learned concepts as repeated measure and found significant main effect for treatment [ $F(1,555) = 6.725, p = 0.01$ ] because experimental

group students underperformed control group students on both pretest and post-test. But, no significant interaction was observed between the scores and treatment: control group students improved from 0.084 to 0.956 points per problem from pretest to post-test whereas experimental group students improved from 0.041 to 0.928 points.

**Discussion.** When students had to answer cloze procedure questions embedded in the feedback, they did spend significantly more time per problem and they learned concepts with significantly fewer practice problems. However, they did not learn significantly more concepts and their change in score from pretest to post-test was not any different on the learned concepts from that of control group.

The effectiveness of the treatment varied by semester, which we find to be curious. This applied to both the number of concepts learned by experimental group and the number of practice problems solved per learned concept.

We also found numerous significant interactions among demographic groups, e.g., treatment and semester, representation and sex, and sex and semester on pretest score per problem; representation and sex on the time spent per pretest problem; representation and sex and representation and semester on the number of concepts learned; and treatment and semester on the practice problems solved per learned concept. So, increased time on task may benefit different demographic subgroups differently [8]. We plan to further investigate the differential effects of the intervention on different demographic subgroups in the future.

**Acknowledgments.** Partial support for this work was provided by the National Science Foundation under grant DUE 1432190.

## References

1. Cooper, G., Sweller, J.: Effects of schema acquisition and rule automation on mathematical problem-solving transfer. *J. Educ. Psychol.* **79**, 347–362 (1987)
2. Kumar, A.N.: Explanation of step-by-step execution as feedback for problems on program analysis, and its generation in model-based problem-solving tutors. *Technol. Instruction, Cogn. Learn. (TICL) J.* **4**(1) (2006). Special Issue on Problem Solving Support in Intelligent Tutoring Systems
3. Stark, R.: Learning by worked-out examples. The impact of incomplete solution steps on example elaboration, motivation, and learning outcomes. Huber, Bern, CH (1999) (in German)
4. Paas, F.G.: Training strategies for attaining transfer of problem-solving skill in statistics: a cognitive-load approach. *J. Educ. Psychol.* **84**(4), 429 (1992)
5. Mathan, S.: Recasting the feedback debate: benefits of tutoring error detection and correction skills. Ph.D. dissertation, Carnegie Mellon University, PA (2003)
6. Mathan, S.A., Koedinger, K.R.: An empirical assessment of comprehension fostering features in an intelligent tutoring system. In: Cerri, S.A., Gouardères, G., Paraguaçu, F. (eds.) ITS 2002. LNCS, vol. 2363, pp. 330–343. Springer, Heidelberg (2002)

7. Kumar, A.N.: An evaluation of self-explanation in a programming tutor. In: Trausan-Matu, S., Boyer, K.E., Crosby, M., Panourgia, K. (eds.) ITS 2014. LNCS, vol. 8474, pp. 248–253. Springer, Heidelberg (2014)
8. Kumar, A.N.: Need to consider variations within demographic groups when evaluating educational interventions. In: Proceedings of Innovation and Technology in Computer Science Education (ITiCSE 2009), Paris, France, pp. 176–180, July 2009

# The Effect of Friendship and Tutoring Roles on Reciprocal Peer Tutoring Strategies

Michael A. Madaio<sup>(✉)</sup>, Amy Ogan, and Justine Cassell

Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213, USA  
{madaio, aeo, justine}@cs.cmu.edu

**Abstract.** Intelligent Tutoring Systems that employ a teachable agent or reciprocal tutoring agent are designed to elicit the beneficial effects of tutoring, known as the tutor learning effect. However, untrained tutors do not spontaneously use beneficial tutoring strategies, and in a reciprocal format, it is unclear how the tutor learning effect affects those tutors' future problem-solving. Here, we examine the effect that the relationship between tutor and tutee has on their likelihood to use various tutoring and learning strategies, and the impact those strategies have on tutees' future problem-solving in a reciprocal format. We find that among friends, tutees tend towards more verbalization of their problem-solving, with their tutors adopting a more questioning tutoring style, while among strangers, tutees use more shallow questions, with more procedural instruction from their tutor.

**Keywords:** Tutor learning effect · Reciprocal tutoring · Peer tutoring

## 1 Introduction and Related Work

Teachable Agents (TAs) have been proposed as a scalable way to achieve the benefits of learning-by-teaching that have been seen in human tutoring dyads, where the tutor benefits from the interaction as much or more than the tutee. This is also known as the tutor learning effect [5, 9, 11]. The fixed-role design of current TAs, however, may not elicit the tutor learning effect if the tutor does not have sufficient prior knowledge to tutor, or if they lack the opportunity to apply what they learned while subsequently problem-solving as a tutee. To address this gap, “reciprocal tutoring agents” have been proposed that can both tutor and be tutored by the student [3, 8]. To implement such a system effectively, we must first understand how the use and impact of specific tutoring and learning strategies differs for the tutor and tutee when tutoring in a reciprocal format, as prior research on both TAs and reciprocal tutoring often lacks the fine-grained interaction data necessary to understand the tutor learning effect [5, 7, 10–12, 14].

In this paper, we examine how tutors' and tutees' explanations during the tutoring sessions incorporate knowledge-building (e.g. elaborated explanations of conceptual knowledge) or knowledge-telling (e.g. summarization with little monitoring or elaboration) [11]. We also follow [6] in examining the questions

asked by the tutors and tutees, both deep (e.g. probing their partner for conceptual understanding) and shallow (e.g. asking about procedures or answers). Additionally, prior work suggests that the particular discourse styles of friends provide unique resources for problem-solving and learning [1], indicating that the rapport between friends that allows them to disagree without consequences may account for their ability to foster more mature thinking in one another. If this is true, we want to understand to what extent the relationship between the tutor and tutee affects their use of beneficial tutoring and learning strategies, particularly because TAs (and perhaps reciprocal tutoring agents) rely on the “protégé effect,” which evokes in tutors a feeling of responsibility for their virtual student [2, 4, 8].

This paper expands on prior work by (1) providing a fine-grained, utterance-level analysis of the ways that explanations and questions are used by tutors and tutees of differing relationship statuses. (2) We then shed light on whether and how, in a reciprocal tutoring format, the “tutor learning effect” still holds, to understand whether a tutor’s future problem-solving is more affected by the explanations and questions they use while a tutor, those their tutor uses, or those they use while problem-solving as a tutee.

## 2 Methodology

**Research Questions.** *RQ1:* How frequently do peer tutors and tutees use knowledge-telling, knowledge-building, shallow and deep questions, and metacognitive reflection, and how does that use differ between friend and stranger dyads? Following prior literature we hypothesize that tutors will explain more than tutees, and tutees will question more than tutors [14], and that all participants will use more knowledge-telling than-building and ask more shallow than deep questions [6]. We also hypothesize that dyads of strangers are less likely than friends to use knowledge-building and metacognitive reflection, due to the social risks from explaining incorrectly or reflecting on one’s knowledge in front of a stranger [1, 11].

*RQ2:* Which has more impact on a tutee’s problem-solving strategies: the tutoring strategies they used in the prior period when they were a tutor, the learning strategies they use as a tutee trying to solve those problems, or the tutoring strategies their tutor uses? Perhaps counter-intuitively, our hypothesis, based on the benefits seen from the tutor learning effect, is that the knowledge-building, deep questions, and metacognitive reflection used while tutoring will better predict correctly solved problems in those tutors’ subsequent problem-solving than the strategies their tutors use while teaching them, or the strategies they use while problem-solving [12, 14, 15].

**Dialogue Corpus.** Our corpus comprises interaction data from 10 peer dyads (mean age 13.4,  $SD = 1.1$ ), reciprocally tutoring one another in algebra for 4 weekly hour-long sessions. Each session was split into two tutoring periods, with students switching tutoring roles after each period. Half the dyads were boys,



**Table 1.** Tutoring and learning strategy codes, definitions, and examples

Code	Definition	Example
Knowledge-telling	Stating numbers, variables, procedures, or the answer.	Divide it by 9.
Knowledge-building	Providing elaborated explanations of the idea, concept, or reasoning.	That's because it can be reduced.
Metacognitive Reflection	Verbally reflecting on their or their partner's knowledge.	What I don't understand is what we do with the p.
Shallow Question	Asking for confirmation of an answer, a definition, or an example.	Do I move the numbers first?
Deep Question	Asking about reasoning, concepts, or hypotheticals.	What do you think you would do with this side?

and half girls to mitigate the stereotype threat seen in mixed-gender tutoring dyads [10]. Half the dyads self-defined as friends and half as strangers prior to the tutoring session. Video and audio data were recorded for each session, transcribed, and segmented by clause. Following [6, 11], five annotators coded the corpus for explanations, questions, and reflection used by either the tutor and tutee, as explained in Table 1 (all Krippendorff's  $\alpha > .7$ ). We will refer to these as tutoring strategies when used by the tutor, and learning strategies when used by the tutee. The corpus was also coded for off-task utterances ( $\alpha = .75$ ).

**Learning Outcome Measures.** Each student took a pre-test in the first session with 20 procedural questions, and after the final session, a counterbalanced, isomorphic post-test. The tutees were given 10 problems to solve in each tutoring period, scored as 1 if successfully completed in its entirety, and 0 if not. Because in this analysis we desire to associate strategies used in each tutoring period with an outcome measure for that same period, we used the problem-solving performance in each period as our measure of learning, instead of gains from pre- to post-test.

### 3 Results

**Descriptive Statistics.** The mean percent of problems successfully solved across all sessions was .63 ( $SD = .38$ ), with no significant difference for gender or relationship. Although friends talked more than stranger dyads overall, and friends had more off-task talk than strangers, interestingly, there was no significant difference in their on-task talk. See Table 2 for means and standard deviations of friend and stranger dyads' utterances.

**RQ1: Frequency of Tutoring Strategies.** Because of individual variation in number of utterances (particularly social talk), we analyzed each tutoring and

**Table 2.** Means and Standard Deviations of Utterance Counts, with p value from a t-test of friend and stranger dyads.

	All dyads	Friends	Strangers	Significance
<b>All Utterances</b>	144 (81.7)	178 (82.1)	109 (65.6)	p < .001
<b>Off-task</b>	37 (83)	67 (106)	7.4 (28.9)	p < .001
<b>On-task</b>	127 (65.6)	136 (67.1)	120 (63.6)	Not sig.

learning strategy as a percentage of all on-task utterances, for all dyads, and for each of the four combinations of tutoring role and relationship (See Table 3 for means and standard deviations). As we expected, for all dyads, knowledge-telling was used more frequently than knowledge-building, and shallow questions more than deep questions. However, the variations in those initial results led us to explore interaction effects between gender, relationship, and role for tutoring and learning strategies. We therefore conducted a series of 5 repeated measures ANOVAs. For each of the 5 strategies, we crossed the between-subjects factors of gender (M/F) and relationship (Friend/Stranger) with the within-subject, repeated measures of role (tutor/tutee) and session (1–4) for a  $2 \times 2 \times 2 \times 4$  ANOVA, with Dyad, Role, and Session as error terms. We employed a Bonferroni correction to account for running multiple tests.

The ANOVA for knowledge-building revealed a significant main effect for role ( $F(3,18) = 12.2, p < .05$ ), with tutors using more knowledge-building than tutees, as expected. The ANOVA for knowledge-telling revealed significant interaction effects for role by relationship ( $F(3,18) = 4.6, p < .05$ ), with friend tutees using more knowledge-telling than friend tutors, while stranger tutors used more than stranger tutees. The ANOVA for shallow questions revealed a significant main effect for role ( $F(3,18) = 21.7, p < .01$ ), with tutees asking more shallow questions than tutors, as expected. There was also an interaction effect for role by relationship ( $F(3,18) = 19.8, p < .01$ ), with stranger tutees asking more shallow questions than friend tutees, and friend tutors asking more than stranger tutors. The ANOVA for deep questions and metacognitive reflection revealed no significant main or interaction effects.

**Table 3.** Means and standard deviations of tutoring and learning strategies

	Knowledge-telling	Knowledge-building	Metacognitive reflection	Shallow questions	Deep questions
<b>All Dyads</b>	0.44 (0.18)	0.04 (0.06)	0.04 (0.04)	0.08 (0.08)	0.01 (0.02)
<b>Friend Tutees</b>	0.42 (0.19)	0.02 (0.02)	0.05 (0.05)	0.08 (0.06)	0.01 (0.02)
<b>Stranger Tutees</b>	0.46 (0.20)	0.03 (0.05)	0.03 (0.02)	0.13 (0.11)	0.01 (0.02)
<b>Friend Tutors</b>	0.38 (0.12)	0.03 (0.04)	0.05 (0.04)	0.07 (0.04)	0.01 (0.02)
<b>Stranger Tutors</b>	0.38 (0.13)	0.09 (0.08)	0.03 (0.02)	0.03 (0.03)	0.01 (0.02)

**RQ2: Effect of Tutoring and Learning Strategies on Problem-Solving in Reciprocal Tutoring.** Our hypothesis (from the tutor learning effect) was that the tutoring strategies that participants used while tutoring in period 1 ( $T_1$ ) would be more predictive of their problem-solving in the subsequent period ( $T_2$ ) when they are the tutee, than the strategies their tutor uses to teach them in  $T_2$ . It is therefore necessary to separate the effect of the tutoring strategies that a given participant (e.g.  $P_1$ ) used while tutoring ( $P_1, T_1$ ) on their subsequent problem-solving in  $T_2$ , from the effect of the tutoring strategies that *their* tutor ( $P_2, T_2$ ) used while  $P_1$  was problem-solving. We also wanted to distinguish both of those effects from the effect of the explanations and questions that they ( $P_1$ ) used while problem-solving ( $P_1, T_2$ ).

We thus created three sets of linear mixed effect models. In all models, we set as fixed effects the pre-test percent, gender, and relationship, and set as random effects the dyad and the session. We also included as fixed effects in model (1) the learning strategies used by the tutee ( $P_1, T_2$ ); in model (2) the strategies used by the same participant when they were previously the tutor ( $P_1, T_1$ ); and in model (3) those strategies used by that participant's tutor ( $P_2, T_2$ ).

After running each of the three mixed-effect models, we used pairwise ANOVAs to compare each model's ability to predict the tutee's problem solving. As hypothesized, Model 2 ( $P_1, T_1$ ; the "prior tutoring" model), was more predictive ( $\chi^2(15) = 4.7, p < .001$ ) than Model 3 ( $P_2, T_2$ ; the "current tutor"). Interestingly, Model 1 ( $P_1, T_2$ ; the "current tutee" model) was in fact more predictive ( $\chi^2(12) = 7.4, p < .001$ ) than Model 2 ( $P_1, T_1$ ; the "prior tutoring" model), and it was also more predictive ( $\chi^2(12) = 8.9, p < .001$ ) than Model 3 ( $P_2, T_2$ ; their "current tutor").

To better understand the effect of the individual learning strategies used by the tutee, we examined the coefficients of each of the fixed effects for the most predictive model, the current tutee model (Model 1). As expected, the fixed effect of pre-test was significantly predictive of problem-solving, with a coefficient of .31 ( $p < .01$ ). Unexpectedly, however, shallow questions from the tutee were positively predictive (.29), and deep questions were negatively predictive (-.29), (both at ( $p < .01$ )). Knowledge-telling and -building were both positively predictive of problem-solving, but neither was significant.

## 4 Discussion and Conclusion

We set out to explore the impact that role and relationship have on the use of tutoring and learning strategies, and how variations in that strategy use impacted problem-solving, to identify implications for a teachable agent or reciprocal tutoring system. Although we found that, overall, friends spoke more, and used more off-task utterances than strangers, the amount of on-task talk was equivalent, indicating that friends were supplementing their tutoring talk with social talk, not replacing it. We found that friend tutors asked more questions of their tutees than stranger tutors, indicating a more Socratic questioning style of instruction (e.g., "Two times what equals eight?"). Friend tutees in return used greater amounts of knowledge-telling than their tutors, suggesting that friend

tutors avoided giving direct instruction, while the tutees felt more comfortable verbalizing their problem-solving while working. For instance, a friend tutee said, “okay so that would give you a two... negative two x equals two”, allowing her tutor to provide feedback on a step-level instead of simply evaluating the answer. On the other hand, stranger tutees asked more questions than friend tutees, suggesting that strangers had more of a disposition towards answer- or instruction-seeking than friends. For example, from a stranger tutee, “So do I start with the eight or the two?” In return, stranger tutors used more knowledge-telling than stranger tutees, resulting in more procedural instructions. For example, from a stranger tutor, “Add the eight.” which was subsequently performed by the tutee. This all suggests further interactional benefits to a relationship or friendship between tutor and tutee.

The surprising negative coefficient that we saw for asking deep questions might be because asking conceptual questions is indicative of that tutee’s lack of prior knowledge, or because the tutees received an unsatisfactory response from their tutor to these deep questions. Upon further investigation, the majority of the tutor responses to deep questions were knowledge-telling or a shallow question, instead of the knowledge-building, elaborated response which we would expect to be a beneficial response. More research is needed on how best to provide the structured support needed for untrained peer tutors to provide the elaborated knowledge-building which has been shown to lead to tutor learning, whether that support be from a teachable agent or reciprocal tutoring system. Additionally, our future work will use conceptual items on a repeated pre- and post-test to better understand how tutors’ and tutees’ conceptual knowledge improves from their use of tutoring and learning strategies, in addition to their problem-solving.

For designers of intelligent tutoring systems, whether for a tutoring agent, a teachable agent, or a reciprocal tutoring agent, it is important to understand the consequences that role and relationship have on the specific tutoring and learning strategies used in the tutoring discourse. With a perceived friendship or rapport between tutor and tutee, the tutee may feel more comfortable verbalizing their problem-solving, allowing for more step-level feedback from the tutor, and tutors may feel more comfortable asking questions about their tutee’s problem-solving process instead of simply giving explicit instructions. In this paper, we examined the ways that various types of tutoring and learning strategies are affected by the relationship of the dyad, and their impact on problem-solving. We also offer one approach to untangling the complex interactions between explanations, questions, and problem-solving in a reciprocal tutoring format.

## References

1. Azmitia, M., Montgomery, R.: Friendship, transactive dialogues, and the development of scientific reasoning. *Soc. Dev.* **2**, 202–221 (1993)
2. Biswas, G., Leelawong, K., Schwartz, D., Vye, N., The Teachable Agents Group at Vanderbilt: Learning by teaching: a new agent paradigm for educational software. *Appl. Artif. Intell.* **19**, 363–392 (2005)

3. Chou, C.Y., Chan, T.W.: Reciprocal tutoring: design with cognitive load sharing. *Int. J. Artif. Intell. Educ.*, 1–24 (2015)
4. Chase, C.C.: Teachable Agents and the Protégé Effect: Increasing the Effort Towards Learning. *Psychometrika* (2006)
5. Fantuzzo, J.W., King, J., Heller, L.R.: Effects of reciprocal peer tutoring on mathematics, school adjustment: a component analysis. *J. Educ. Psychol.* **84**, 331–339 (1989)
6. Graesser, C., Person, N.K.: Question asking during tutoring. *Am. Educ. Res. J.* **31**, 104–137 (1994)
7. Leelawong, K., Biswas, G.: Designing learning by teaching agents: the Betty’s brain system. *Int. J. Artif. Intell. Educ.* **18**, 181 (2008)
8. Matsuda, N., Keiser, V., Raizada, R., Stylianides, G., Cohen, W.W., Koedinger, K.: Learning by teaching SimStudent. In: Alevan, V., Kay, J., Mostow, J. (eds.) *ITS 2010, Part II. LNCS*, vol. 6095, pp. 449–449. Springer, Heidelberg (2010)
9. Palinscar, A.S., Brown, A.L.: Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. *Cogn. Instr.* **1**, 117–175 (1984)
10. Robinson, D.R., Schofield, J.W., Steers-Wentzell, K.L.: Peer and cross-age tutoring in math: outcomes and their design implications. *Educ. Psychol. Rev.* **17**, 327–362 (2005)
11. Roscoe, R., Chi, M.: Understanding tutor learning: knowledge-building and knowledge-telling in peer tutors’ explanations and questions. *Rev. Educ. Res.* **77**, 534–574 (2007)
12. Roscoe, R.D., Chi, M.T.H.: Tutor learning: the role of explaining and responding to questions. *Instr. Sci.* **36**, 321–350 (2008)
13. Walker, E., Rummel, N., Koedinger, K.R.: Integrating collaboration and intelligent tutoring data in the evaluation of a reciprocal peer tutoring environment. *Res. Pract. Technol. Enhanced Learn.* **4**(3), 221–251 (2009)
14. Walker, E., Rummel, N., Koedinger, K.R.: Designing automated adaptive support to improve student helping behaviors in a peer tutoring activity. *Int. J. Comput. Support. Collaborative Learn.* **6**, 279–306 (2011)
15. Webb, N.M., Mastergeorge, A.: Promoting effective helping behavior in peer-directed groups. *Int. J. Educ. Res.* **39**, 73–97 (2003)

# CRISTAL: Adapting Workplace Training to the Real World Context with an Intelligent Simulator for Radiology Trainees

Hope Lee<sup>1</sup>, Amali Weerasinghe<sup>1(✉)</sup>, Jayden Barnes<sup>1</sup>, Luke Oakden-Rayner<sup>2</sup>, William Gale<sup>1</sup>, and Gustavo Carneiro<sup>1</sup>

<sup>1</sup> Department of Computer Science, University of Adelaide, Adelaide, SA, Australia  
{amali.weerasinghe, gustavo.carneiro}@adelaide.edu.au

<sup>2</sup> Department of Radiology, Royal Adelaide Hospital, Adelaide, SA, Australia  
lukeoakdenrayner@health.sa.gov.au

**Abstract.** Intelligent learning environments based on interactions within the digital world are increasingly popular as they provide mechanisms for interactive and adaptive learning, but learners find it difficult to transfer this to real world tasks. We present the initial development stages of CRISTAL, an intelligent simulator targeted at trainee radiologists which enhances the learning experience by enabling the virtual environment to adapt according to their real world experiences. Our system design has been influenced by feedback from trainees, and allows them to practice their reporting skills by writing freeform reports in natural language. This has the potential to be expanded to other areas such as short-form journalism and legal document drafting.

**Keywords:** Adult learning · Self-regulated learning · Simulated environments for learning · Radiology training · Natural language processing

## 1 Introduction

Learning by doing: this is the underlying concept of intelligent tutoring systems (ITSs), serious games and immersive activities, and learning environments such as these are becoming increasingly popular. Their ability to generate an individualized environment tailored to each user's learning needs is one of their main strengths. These adaptations are generally based on a learner's interactions within the digital world, and as a result, learners find it difficult to transfer their knowledge to real world tasks [1]. Consequently, there has been a strong interest in linking learners' real world behaviors with the digital world to facilitate a more integrated learning experience. On-the-job training is one example of this. The focus of the systems ImREAL [1], MIRROR [2], PORML [3], ALPS [4] and KP-Lab [5] is to assist informal workplace training by supporting learners to transform experience into knowledge through socio-pedagogical models. Out of these, only ImREAL [2], MIRROR [3] and PORML [4] link both worlds by giving learners simulated tasks that correlate with real world activities. However, these systems rely on the learners to actively participate in knowledge transfer between the workplace and the simulated environment. The PORML [4] framework supports reflection in a

digital environment for emergency service workers immediately after performing a job activity. However, the adaptation is specific to that job activity and does not influence the system's behavior in subsequent activities.

As the domain used in this research is radiology, we have also explored how ITSs have supported learning in this area. VIA-RAD [6], RadTutor [7] and MR Tutor [8] present a constrained format where users are asked to describe radiology images by selecting options from a pre-determined list. These systems focus entirely on the virtual world, and real world behaviors are not considered within the simulation. In contrast, a radiology ITS called GIMI (Generic Infrastructure for Medical Informatics) uses real world experiences to modify the learner model by allowing manual input of personal performance data [9]. In contrast to GIMI we plan to automate the integration of real world experiences into our simulator with data mining techniques.

In our research towards developing CRISTAL (Clinical Radiology Intelligent Simulation Tool with Adaptive Learning) we explore the following objectives:

1. How can we use real world behaviors to adapt the virtual learning experience?
2. How can we automate the process of integrating real world behaviors with the simulated environment without active intervention from the user?
3. How can we provide intelligent support for radiology trainees' learning that is relevant to the requirements of their job?

Using real world behaviors to tailor the simulated learning experience, we can enable trainees to seamlessly transition between the roles of worker and learner. The overall goal of our research is to explore how an intelligent simulator that links both real and virtual environments can support individuals as they progress through different phases of their training: starting as workers, progressing to learners and finally becoming experts.

## 2 Radiology Training Practices

Trainee radiologists are qualified doctors who are enrolled in the Royal Australian and New Zealand College of Radiologists' five-year training program to become consultant radiologists. A survey conducted by the Royal Australian and New Zealand College of Radiologists (RANZCR) in 2012 found that the majority (89 %) of trainees spend at least 36 h per week on clinical work, the majority of which is spent writing reports on radiology images [10].

We asked a group of trainees for their input regarding the type of experience they wanted from a simulated tutoring system. They told us they want a system that makes it easy to identify important weaknesses, access relevant cases, gives high-quality but targeted feedback and suggests (rather than demands and tests on) relevant study material. They disliked systems that impose set exercises and learning content, and responded negatively to the idea of adaptive dialogues. Their response was: "We know how to learn, and we have our own preferred resources." The trainees' responses echo some of Malcolm Knowles' well-known assumptions regarding adult learners: they have a clear preference for self-directed learning, a strong internal motivation to learn, and are

oriented towards learning tasks that have immediate relevance to their societal roles [11]. This emphasis on andragogy (the theory of adult learning) is echoed by the authors of the ImREAL project [2].

### 3 System Overview

As per the extended self-regulated learning (SRL) model described by Hetzner et al. [1], the architecture of CRISTAL spans across the real world and the simulated environment (Fig. 1). We will first describe the real world environment. In the workplace, the majority of images that a trainee reports on are subsequently sent to a consultant radiologist, who then types an addendum containing any necessary corrections. We plan to collect and store all trainees’ reports and their corrections in our database. This information will be used to update the learner model in the simulated environment, enabling the training module to adapt the learning task based on a trainee’s weaknesses in the real world. The training module will then request relevant problems from the report and image database. This database consists of real radiology images (such as x-rays and CT scans) and their corresponding reports. These reports have been written on-the-job by domain experts in the past.

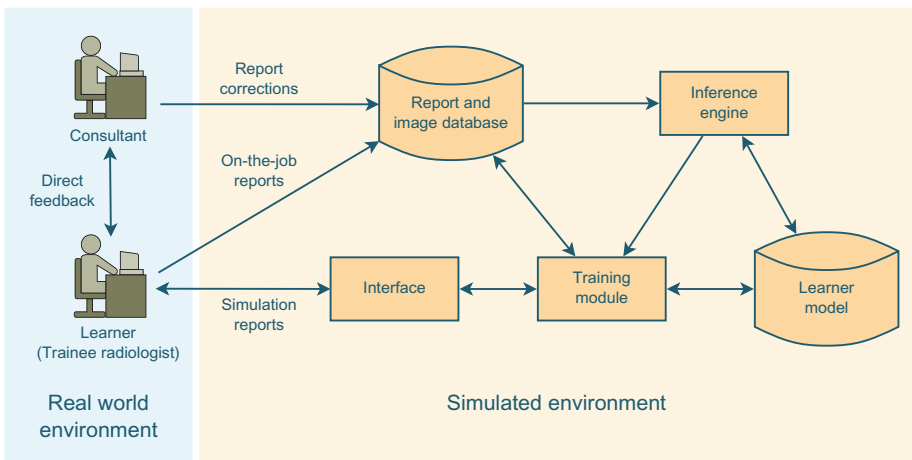


Fig. 1. Diagram of the system architecture

Mirroring the real world environment, the simulator interface will present one of these images to the learner (trainee radiologist), who will be prompted to write a report on the image in freeform natural language. Once complete, the report will be sent to the inference engine to be compared with the original report. The quality of the trainee’s report will be determined both by the presence of the correct diagnosis and the completeness of the report.

We are currently using a latent semantic indexing (LSI) model to characterise each sentence in our report corpus. To determine the completeness of the trainee’s report,



each sentence is compared with sentences from the original report and matched with the one with which it has the greatest cosine similarity. If a sentence pair’s similarity is above a pre-defined threshold the trainee’s sentence is considered to contain appropriate meaning. Below this threshold, the trainee’s sentence is considered to be incorrect. The training module will also detect missing sentences: important sentences in the original report that were not identified in the trainee’s report, via the same thresholding approach. The learner model will be updated with data regarding the correctness and completeness of the trainee’s report.

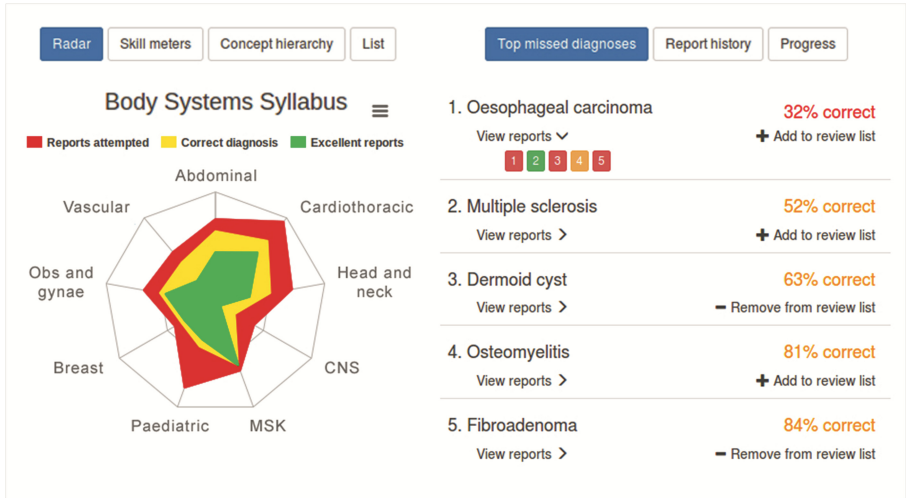


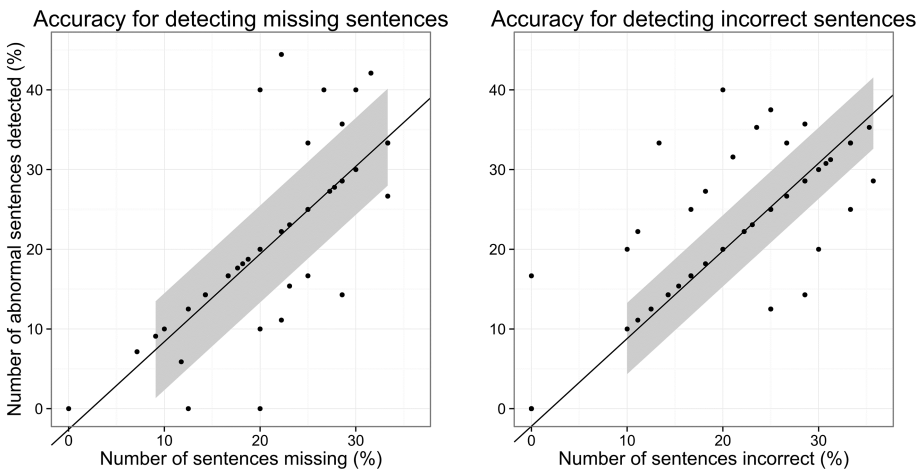
Fig. 2. Screenshot of the open learner model

Trainees will receive feedback directly after report submission, however they will also be able to review their overall performance through an open learner model (Fig. 2). Diagnoses are grouped according to the Body Systems Syllabus of the RANZCR Radiodiagnosis Training Program Curriculum [12], and multiple views are provided to enable trainees to select their preferred format. The importance of diagnoses will be determined based on their categorization within the Curriculum. The Curriculum will also influence problem selection: for example, first year trainees will be provided with images related to critical diagnoses (known as Key Conditions in the Curriculum), with a specific emphasis on conditions they have not seen in practice and those they have reported incorrectly. Senior trainees will instead be exposed to more esoteric conditions (Categories 2–3 in the Curriculum) selected from areas they have had limited exposure to, or reported with high rates of error. To respect their autonomy, trainees will also have the option to select specific learning topics.

#### 4 Preliminary Testing

We have conducted some initial basic tests to assess the quality of our language model using LSI. To simulate missing and incorrect sentences, whole sentences were removed

at random, or swapped in from reports containing different diagnoses. We tested how many of these abnormal sentences were detected by our system relative to the number of alterations. As expected, the variation in missing and incorrect sentences is directly proportional to the number of alterations made in the reports (Fig. 3), suggesting we are able to identify unmatched sentences. We do note however that the LSI model is unlikely to achieve acceptable performance in all of the required tasks. This training module will be improved by the implementation of a recurrent neural network language model to overcome the more difficult challenges: the identification of diagnostic sentences (which are most important for the teaching process), and the discrimination between positive and negative diagnostic sentences (as negations are not well captured with naive “bag-of-words” models like LSI [13]).



**Fig. 3.** Preliminary tests of the accuracy of missing and incorrect sentence detection

## 5 Future Work

At this stage we have developed a working model of CRISTAL’s simulated learning environment. Our current focus is developing the inference engine for processing trainees’ on-the-job reports and their corresponding consultant corrections. This will complete the feedback loop between the training module and the real world environment. Our next step for clinical implementation is preliminary evaluation of the simulation environment. We will present a set of pre-determined training scenarios to trainees and adjust our model based on their feedback.

The strengths of CRISTAL are that it allows automated integration of real world data into the learner model, and it has the ability to analyse and give feedback on freeform reports. Our system has the potential to be extended to ITS research in other domains, if those domains fulfil the following criteria: there is a real-world task in which adult learners could benefit from intelligent feedback and adaptive training, the task is performed frequently and results in text output, and there is a written “ground truth” for

this text to be compared against. Examples could include education settings with short answer questions (including in online education), as well as professions such as short-form journalism and legal document drafting, where each document is edited by a senior practitioner. Thus our system of connecting the real-world and the simulated environment can be seen to apply more widely to ITS research, contributing to the impact and relevance of virtual learning environments.

## References

1. Hetzner, S., Steiner, C.M., Dimitrova, V., Brna, P., Conlan, O.: Adult self-regulated learning through linking experience in simulated and real world: a holistic approach. In: Kloos, C.D., Gillet, D., Crespo García, R.M., Wild, F., Wolpers, M. (eds.) EC-TEL 2011. LNCS, vol. 6964, pp. 166–180. Springer, Heidelberg (2011)
2. Krogstie, B.R., Prilla, M., Wessel, D., Knipfer, K., Pammer, V.: Computer support for reflective learning in the workplace: a model. In: 2012 IEEE 12th International Conference on Advanced Learning Technologies (ICALT), pp. 151–153. IEEE, Italy (2012)
3. Eamsinvattana, W.: Reflective Dialogue for On-the-Job Training in Emergency Services. School of Computing, University of Leeds, United Kingdom (2011)
4. Murphy, A., Laxton, J.: Views of a structured assessment tool for observing practice. *Soc. Work Educ.* **33**, 190–208 (2014)
5. Kai, H., Hanni, M., Hannu, M.: Design principles for the knowledge-practices laboratory (KP-Lab) project. In: Proceedings of the 7th International Conference on Learning Sciences, pp. 934–935. International Society of the Learning Sciences, Bloomington (2006)
6. Rogers, E.: VIA-RAD: a blackboard-based system for diagnostic radiology. *Artif. Intell. Med.* **7**, 343–360 (1995)
7. Azevedo, R., Lajoie, S.: The cognitive basis for the design of a mammography interpretation tutor. *Int. J. Artif. Intell. Educ.* **9**, 32–44 (1998)
8. Sharples, M., Jeffery, N.P., du Boulay, B., Teather, B.A., Teather, D., du Boulay, G.H.: Structured computer-based training in the interpretation of neuroradiological images. *Int. J. Med. Informatics* **60**, 263–280 (2000)
9. Yap, M.H., Gale, A.G., Scott, H.J.: Generic infrastructure for medical informatics (GIMI): the development of a mammographic training system. In: Krupinski, E.A. (ed.) IWDM 2008. LNCS, vol. 5116, pp. 577–584. Springer, Heidelberg (2008)
10. Munro, P.L., Bradshaw, N., Stephenson, N.: 2012 RANZCR Radiology Workforce Census Report: Trainees. RANZCR, Sydney (2013)
11. Knowles, M.S.: *The Modern Practice of Adult Education*. New York Association Press, New York (1970)
12. The Royal Australian and New Zealand College of Radiologists: Radiodiagnosis Training Program Curriculum. RANZCR, Sydney (2014)
13. Wiegand, M., Balahur, A., Roth, B., Klakow, D., Montoyo, A.: A survey on the role of negation in sentiment analysis. In: Proceedings of the Workshop on Negation and Speculation in Natural Language Processing, pp. 60–68. Association for Computational Linguistics (2010)

# Posters

# A System for Gamifying Ubiquitous Learning Situations Supported by Multiple Technologies

Alejandro Ortega-Arranz, Juan A. Muñoz-Cristóbal,  
Alejandra Martínez-Monés, Miguel L. Bote-Lorenzo, and Juan I. Asensio-Pérez

GSIC-EMIC Research Group, Universidad de Valladolid, Valladolid, Spain  
{alex,juanmunoz}@gsic.uva.es, amartine@infor.uva.es,  
{migbot,juaase}@tel.uva.es

**Abstract.** Gamification is the use of game design elements in non-game contexts, and it has reported potential benefits for students. However, the proposals supporting teachers to create gamified ubiquitous learning situations are tied to specific activities and enactment technologies. To start addressing this issue, we propose a system to help teachers design and deploy these situations involving a variety of technologies frequently used in education.

**Keywords:** Gamification · Ubiquitous learning · Game elements · VLE

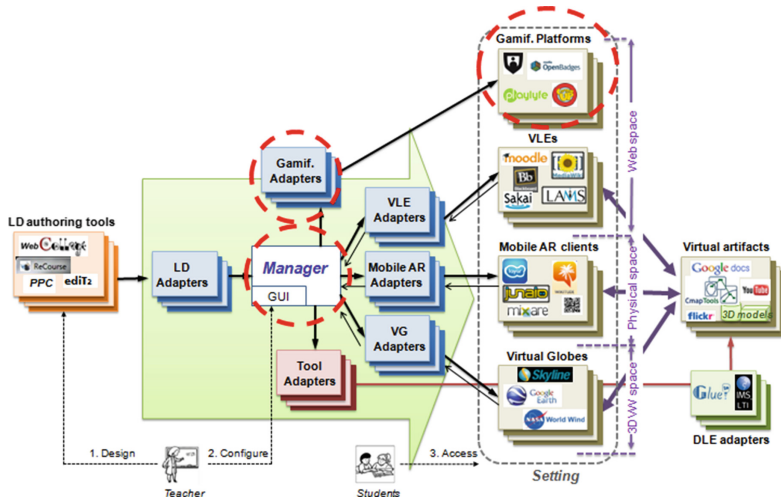
## 1 Introduction

Gamification is an emerging technique with potential benefits (e.g. helping drive students' behaviors or increasing their engagement) in different educational approaches such as ubiquitous learning [1]. Unfortunately, the proposals supporting teachers to create gamified ubiquitous learning situations (GULS) are usually tied to specific learning activities and enactment technologies (e.g. TaggingCreaditor, ARIS or ARLearn). Teachers are thus forced to learn and use new specific technologies and activities which might not match with their practice. As a consequence, this limitation can constrain the adoption of GULS in real educational settings. To start addressing this issue, we propose<sup>1</sup> a system that supports the creation and enactment of GULS that may involve multiple spaces and a variety of technologies, including different VLEs (e.g. Moodle), web 2.0 tools (e.g. Google Drive), AR clients (e.g. Layar) and 3D virtual globes (e.g. Google Earth).

## 2 A Gamified System for Ubiquitous Learning

The proposed system named Gamified GLUEPS-AR, is the result of extending GLUEPS-AR [2], a system for the deployment of ubiquitous learning situations with third-party gamification platforms (GPs) such as Open Badges or

<sup>1</sup> Research partially supported by projects TIN2014-53199-C3-2-R and VA277U14.



**Fig. 1.** Gamified GLUEPS-AR architecture. Dashed red circles show the extension.

Userinfuser. Similar to its predecessor, Gamified GLURPS-AR uses an adaptor-based software architecture (see Fig. 1). Thus, different gamification platform can be integrated by developing the appropriate adaptors. Gamified GLUEPS-AR allows teachers configure the GPs, the game elements to be used (e.g. points and badges), the students' actions that are associated to game elements (e.g. fill-in an AR artifact), the rules that such actions have to meet (e.g. fill-in 5 artifacts) and the rest of technologies used in such ubiquitous learning situation. The proposed system can deploy in multiple gamification platforms thanks to an extension of the underlying data model of GLUEPS-AR [2] that includes GPs-related elements, including the concepts of *user*, *group*, *rewardable action*, *condition*, and different types of *game elements*. This data model enables the conversions required during the process of creating and enacting the gamified ubiquitous learning situations. We have developed an initial prototype of the system and validated some of the described functionalities through a proof of concept with fictitious users. As a future work, we plan further research for evaluating the approach with real teachers and students.

## References

1. Dicheva, D., Dichev, C., Agre, G., Angelova, G.: Gamification in education: a systematic mapping study. *Educ. Technol. Soc.* **18**(3), 75–88 (2015)
2. Muñoz-Cristóbal, J.A., Prieto, L.P., Asensio-Pérez, J.I., Martínez-Monés, A., Jorrín-Abellán, I.M., Dimitriadis, Y.: Deploying learning designs across physical and web spaces: making pervasive learning affordable for teachers. *Pervasive Mob. Comput.* **14**, 31–46 (2014)

# Combining Speech-Acts and Socio-historical Theories to Monitor and Analyze the Cognitive Evolution of Students on VLE's Records

Gustavo Schwarz<sup>1</sup>, João C. Gluz<sup>1</sup>, and Liliana M. Passerino<sup>2</sup>

<sup>1</sup> Post-Graduation Program in Applied Computer Science (PIPCA),  
UNISINOS, São Leopoldo, Brazil

`gustavo.sne@gmail.com, jcgluz@unisinós.br`

<sup>2</sup> Interdisciplinary Center for Educational Technologies (CINTED),  
UFRGS, Porto Alegre, Brazil  
`liliana@cinted.ufrgs`

**Abstract.** The present work proposes a model for monitoring and analysis of student's activities realized with the help of digital environments, which is based on three assumptions: a linguistic approach to socio-historical theory, the use of standards to register educational activities and the application of Ontology and Bayesian Networks technologies to operationalize the model. Results of experiments are presented, showing accuracy and precision coefficients achieved with the experimental prototype of the model.

**Keywords:** Tincan · Bayesian classifier · Speech-Acts · Socio-historical theory

## 1 Introduction

Monitoring activities realized on Virtual Learning Environments (VLE) could make these environments more productive for students and teachers. But, only to collect and summarize statistics about student activities cannot be enough to produce positive results. It is important to consider semantic issues. VLE's activities logs have proprietary formats, making it difficult to create standardized monitoring and analysis tools. The present work addresses these issues. The basis of this work is an analysis of the standardized logging mechanisms available today to register the educational activities of participants in VLEs, under the perspectives of the Speech-Acts [3] and Socio-Historical [4] theories. The work uses Speech-Acts linguistic approach to identify Socio-Historical mediation actions in learning processes. With the results of this analysis, it is proposed a computational model called *InterActua* for monitoring the educational activities that occur in some VLE as linguistic and social interactions among the participants of this environment. Different from other monitoring models and systems, the computational model proposed is based on three fundamental assumptions: a theoretical approach based on Speech-Acts theory for the linguistic analysis and Socio-Historical theory for social interaction analysis, the use of standards to represent and store the activity log of VLE and the application of Semantic Web and Bayesian inference technologies to operationalize the ontological and dynamic aspects of the model.

**Table 1.** Accuracy, precision and recall coefficients obtained on experiments with *InterActua*

	Experiment 1				Experiment 2			
	Assert.	Dir.	Decl.	Comm.	Assert.	Dir.	Decl.	Comm.
Accuracy	0.90	0.93	0.74	0.81	0.91	0.93	0.77	0.92
Precision	0.64	0.55	0.46	0.63	0.50	0.83	0.76	0.83
Recall	0.87	0.92	0.63	0.78	0.60	0.63	0.59	0.62

## 2 The InterActua Model

Signs and instruments used in learning processes are represented by *InterActua*'s ontological model, which is formed by two ontologies: (a) the *TinCan metadata* ontology, which represents all metadata from TinCan standard [1], describing records of learning activities executed by VLE users, (b) the *Speech Mediation Acts* ontology, which defines a speech-act taxonomy for mediation actions, the classification of regulation categories and the basic relationships between these two conceptual frameworks. The dynamic model of *InterActua* is composed by several collaborating processes. Initially, log records from *Activity*, *Course*, *Forum*, and *Evaluation* tools of *Moodle* VLE are converted to *TinCan* records using *Aelius* part of speech tagger [2] to identify type of verb linked to the record. Then, the *TinCan* to RDF converter transforms Subject-Verb-Object *TinCan* triples in Predicate-Subject-Object RDF triples. Finally, a Bayesian Speech-Acts classifier identifies the subclass of *MediationAction* class that this record belongs. Classifications of actions were based on Speech-Acts types: assertive, directive, declarative and commissive. Two empirical experiments were realized in two different Universities and with two different classes. Experiment 1 was conducted in 2014, during three months with 30 students from a Research Methodology course, resulting in an activities log with 4,567 records. Experiment 2 was carried out in 2015, during one day class of Programming Language course with 21 students, resulting in a 630 activities log records. Table 1 show the accuracy, precision and recall coefficients achieved with the bayesian classifier. They show clear evidences that it is possible to classify VLE's activities records according to Speech-Acts types. This, indeed, add evidences to the viability of the linguistic approach to identify mediation actions and regulation categories in a learning process. Another important result is the representation of *TinCan* records in RDF, which allowed the use of OWL ontology technology to handle the semantics of these records.

## References

1. ADL: Advanced Distributed Learning – Experience API (2013). <https://www.adlnet.gov/adl-research/performance-tracking-analysis/experience-api/>
2. Alencar, L.F.: *Aelius User's Manual* (2013). <http://aelius.sourceforge.net/manual.html>
3. Searle, J.: *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press (1969)
4. Vygotsky, L.S.: *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press (1978)



# Incorporating Student Choice in E-learning

Avi Segal, Naor Guetta, Amir Taboul, Guy Shani, and Ya'akov (Kobi) Gal

Department of Information Systems Engineering, Ben Gurion University,  
Beersheba, Israel

**Abstract.** We investigate the possibility of increasing students' performance and motivation in e-learning through choice: allowing students to choose educational material and questions while learning online. We ran a user study in which 5th grade students were repeatedly able to choose between a pair of math questions that were chosen from ascending skill levels based on performance. Our results show that students answered more difficult questions and obtained better grades when they were allowed to choose the questions than a baseline approach which did not allow choice. Also, we found a strong correlation between the skill levels that were obtained by the students in the choice based system and their class grades. Finally, most students preferred the choice system over a non-choice system.

## 1 Introduction

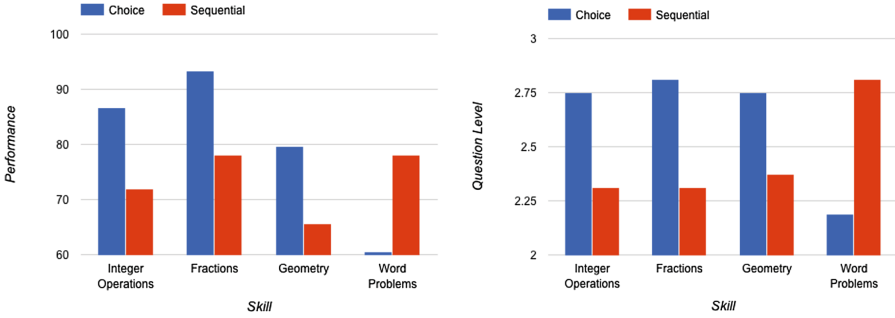
Previous research shows that allowing students to make choices while learning improves their motivation and overall learning gains [1, 4, 5].

We developed an e-learning system which enables students to iteratively choose between a pair of questions to answer at each practice step. The pair of questions presented at each step is selected from a skill set and difficulty level that is determined by students' performance so far. Thus, students can answer questions they prefer, while temporarily (but not indefinitely) avoid frustrating questions that they feel they do not properly understand. We hypothesized that combining choice in e-learning systems will improve students' performance and satisfaction levels. We conducted a controlled user study in a school where 5th graders used our choice system to answer math questions spanning 4 different skills. We compared our system to a standard sequencing method which presents questions to students by increasing level of difficulty.

The results of the study show that when using choice, students achieve better performance and reach higher mastery levels than when using the standard approach on the majority of skills tested. Additionally, we found a strong correlation between the skill levels that were obtained by the students and their class grades. Finally, in a post usage survey, the majority of the students preferred the choice system over the regular sequential question ordering.

## 2 Methodology

Our choice based system supports exercises over various topics (e.g. math, English as a foreign language etc.) and skills (e.g. integers, fractions, geometry). Each



**Fig. 1.** Performance and Level of difficulty obtained for Choice and Sequential Systems

question is labeled with a difficulty level that is determined by a domain expert. At each time step, the student is presented with two questions, each relating to a different skill. The student can decide which of these problems to solve. When a student shows sufficient mastery in a given skill for difficulty level  $i$  by answering correctly a sufficient number of questions of that difficulty level, the difficulty level for the specific skill is increased. The exercise can be terminated when the student has answered a pre-determined number of questions.

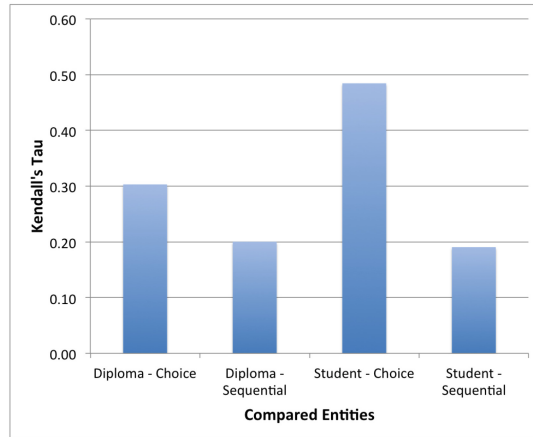
We run a study which included 16 students enrolled in the 5th grade. The study was conducted during the summer vacation outside the context of a classroom. We focused on math questions from 4 different math skills — integer operations, fractions, geometry, and word problems obtained from a recognized national exams organization for 5th graders. The questions were divided into 4 levels of difficulty for each skill by a math teacher. We collected overall 48 questions, 3 questions for each skill and difficulty level. All data collected from the students during the experiment was anonymized.

We compared our choice-based approach to a standard sequential approach in which students were presented with a sequence of questions with increasing difficulty. For each skill, students were presented with 4 questions of increasing difficulty. This approach follows the mastery learning paradigm [2] in which knowledge of simple skills should be acquired before moving on to more difficult questions relating to more complex skills. The skills were pre-arranged in a fixed order, identical for all students.

After experimenting with each system, the students were asked to fill a short survey concerning their experience and opinion about each of the systems. The average time spent on the entire study was approximately 60 min.

### 3 Results and Conclusions

Figure 1 (left) shows the performance of students in the choice system compared to the conventional sequential method. Performance is measured as the portion of correct answers on last question attempt, out of the number of different questions answered. As shown by the figure, the performance of students using



**Fig. 2.** Correlation between the two system ranking, the end of the year diploma ranking, and the student self assessment.

the choice method was substantially higher than the sequential method on the Integer, Fractions and Geometry Problems. In contrast, for Word Problems, the precision of the choice method was considerably lower than that of the sequential method. This can be attributed to the unique difficulty of Word Problems. These problems require mapping from a written description to a math formalization and many students find them hard and challenging [3, 6]. The results are statistically significant for the Fractions and Word Problems skills (ttest,  $p < 0.05$ ).

Figure 1 (right) shows the average maximal level of difficulty that was obtained by the students in each condition. The lowest level is 0 and the highest is 3. As can be seen, the average level on all skills, except for the Word Problems, was much higher in the choice based method. Results are statistically significant for the Fractions and Word Problems skills (ttest,  $p < 0.05$ ).

Thus, in 3 out of 4 skills, the students answered more questions correctly, and solved questions of a higher or similar difficulty level, when given the opportunity to choose which questions to answer. We attribute the different behavior on Word Problems to the intimidation effect of the lengthy textual description.

One of the goals of our system is to serve as a diagnostic tool for a teacher to understand areas of strengths and weaknesses of students. For this, we evaluate the correlation between the students proficiency in each math skill as reflected in their end of year diploma, and their success level in each of the systems in the experiment. We obtained the students end of year grades in each math skill and computed a ranking over skill mastery for students in both the choice and the sequential system.

Figure 2 shows Kendall rank correlation coefficient (Kendall's  $\tau$ ) between the diploma ranking and the two systems — the choice system and the sequential approach. Kendall's  $\tau$  rank correlation is a metric that counts the number

of pairwise disagreements between two ranking lists. Its value ranges from  $-1$ , denoting perfect disagreement, to  $1$  denoting perfect agreement, with  $0$  denoting independence. As can be seen, the correlation of the ranking of the choice system (“Diploma-Choice”) is higher than the correlation of the sequential system (“Diploma-Sequential”).

We also asked students for their own perception of their mastery level of the various skills in the post-experiment questionnaire. The students were asked to provide an evaluation for their mastery of each skill on a scale of  $1$  to  $5$ . Then, we ranked the skills based on the student’s self assessment and compared it again to the two systems. The results are also presented in Fig. 2. As can be seen, the correlation between the student self assessment and the choice system (“Student-Choice”) is higher than the correlation between the student self assessment and the sequential system (“Student-Sequential”). These results provide evidence as to the superior analytic power of the choice system compared with the traditional sequential system.

Finally, we asked the students which of the two systems they preferred.  $62.5\%$  of the students preferred the choice based system, while  $25\%$  expressed preference to the sequential system and  $12.5\%$  did not express any preference.

Our approach provides a proof of concept that combining student choice with selection mechanisms in an e-learning system can improve student performance and satisfaction levels. In future work we plan to test the impact of choice in e-learning on larger and more diverse populations.

## References

1. Assor, A.: Allowing choice and nurturing an inner compass: educational practices supporting students’ need for autonomy. In: Christenson, S.L., Reschly, A.L., Wylie, C. (eds.) *Handbook of Research on Student Engagement*, pp. 421–439. Springer, Heidelberg (2012)
2. Block, J.H., Airasian, P.W., Bloom, B.S., Carroll, J.B.: *Mastery Learning: Theory and Practice*. Holt Rinehart and Winston, New York (1971)
3. Greer, B.: Modelling reality in mathematics classrooms: the case of word problems. *Learn. Instr.* **7**(4), 293–307 (1997)
4. Katz, I., Assor, A.: When choice motivates and when it does not. *Educ. Psychol. Rev.* **19**(4), 429–442 (2007)
5. Ostrow, K.S., Heffernan, N.T.: The role of student choice within adaptive tutoring. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M.F. (eds.) *AIED 2015. LNCS*, vol. 9112, pp. 752–755. Springer, Heidelberg (2015)
6. Verschaffel, L., De Corte, E., Lasure, S.: Realistic considerations in mathematical modeling of school arithmetic word problems. *Learn. Instr.* **4**(4), 273–294 (1994)

# Towards a Mobile Python Tutor: Understanding Differences in Strategies Used by Novices and Experts

Geela Fabric, Antonija Mitrovic, and Kourosh Neshatian

Department of Computer Science and Software Engineering,  
University of Canterbury, Christchurch, New Zealand

geela.fabric@pg.canterbury.ac.nz,  
{tanja.mitrovic, kourosh.neshatian}@canterbury.ac.nz

**Abstract.** We have recently started developing PyKinetic, a mobile tutor for Python. The first type of activities implemented in the tutor is Parsons problems. We conducted a study to evaluate the interface and usability of PyKinetic and to identify and contrast strategies used by novice learners with those of experts. Great feedback and enthusiasm was received for the prospect of PyKinetic and interesting strategies were revealed from both groups.

**Keywords:** Mobile Python tutor · Parsons problems · Novice/expert differences

## 1 Experiment Design

Parsons problems or Parsons programming puzzles [1] consist of a set of randomized lines of code which need to be put in the correct order by dragging and dropping, to produce the desired outcome. We present a prototype of PyKinetic, a Python tutor aimed as a complement to traditional lecture and lab-based introductory programming courses. The prototype contains Parsons problems (with and without distractors), but in the future we plan to add additional types of learning activities. As an initial step towards an intelligent tutor for Python, we performed a study with PyKinetic, which had two goals: to evaluate the usability of Parsons problems implemented in PyKinetic, and also to identify and contrast problem-solving strategies of novice and expert users. Our hypothesis was that the experts would outperform novices in speed and effectiveness in solving problems, and use optimal problem-solving strategies.

The participants were 8 novice and 5 expert participants, students and tutors from an introductory programming course at the University of Canterbury. The study consisted of individual sessions (one-hour long). The version of PyKinetic used in the study contained 7 topics with 21 problems in total: for each topic, there were two problems with distractors and one without.

After providing informed consent, the participants interacted with the tutor. Think-aloud protocol was used and the screen of the device was recorded as well as verbal comments of the participants. The novices were free to choose problems as they wished, but were asked to attempt at least one problem from each topic. The experts were asked to

attempt the problems that majority of novices attempted, to compare the problem-solving strategies used. At the end, participants filled a questionnaire, which included questions about their background and questions about PyKinetic. A similar study [2] with a mobile tutor using Parsons problems was previously conducted by Karavirta et al. [3].

## 2 Results and Conclusions

As expected, the experts were generally faster in solving problems. A wide range of strategies was observed from both groups. A common strategy was to focus on a particular type of statement and move it i.e. variable declarations, function calls and print statements. A specific version of this strategy was used for problems with functions, when the participants moved the function statement first, followed by the docstring. Half of the novices grouped lines superficially based on indentations. Such a strategy shows reliance on a superficial feature rather than trying to understand the context of code. This strategy however, allowed novices to eliminate distractors and arrange the lines logically, especially with conditional statements. After applying this strategy, the novices either tried to reason about the lines in each group, or used the trial and error strategy. One novice used a unique strategy but eventually abandoned the problem, when he/she deleted all lines, and then retrieved the necessary ones from the trash. The experts used superior strategies in comparison to novices. A common strategy used by experts was to build the solution from top to bottom, which was verbally explained by some. This strategy shows that experts have a model solution, and are working towards matching it. All experts used this strategy, but not always exclusively. One expert alternated between this strategy and another strategy, which consisted of combining syntactically and logically similar LOCs with similar indentations and then logically placing them in the correct order. The strategies used by experts demonstrated a higher level of knowledge.

In future work, we will enhance PyKinetic by developing a constraint-based model [4] of the domain. Such domain model will allow the tutor to identify mistakes as well as sub-optimal strategies and take suitable instructional actions.

## References

1. Parsons, D., Haden, P.: Parson's programming puzzles: a fun and effective learning tool for first programming courses. In: Proceedings of the 8th Australian Conference on Computing Education, vol. 52, pp. 157–163. Australian Computer Society (2006)
2. Ihantola, P., Karavirta, V.: Two-dimensional parson's puzzles: the concept, tools, and first observations. *J. Inf. Technol. Educ.* **10** (2011)
3. Karavirta, V., Helminen, J., Ihantola, P.: A mobile learning application for parsons problems with automatic feedback. In: Proceedings of the 12th Koli Calling International Conference on Computing Education Research, pp. 11–18 (2012)
4. Mitrovic, A.: Fifteen years of constraint-based tutors: what we have achieved and where we are going. *User Model. User-Adap. Inter.* **22**(1–2), 39–72 (2012)

# Modeling Negative Affect of Novice Programming Students Using Keyboard Dynamics and Mouse Behavior

Larry A. Veal<sup>1,2</sup> and Ma. Mercedes T. Rodrigo<sup>2</sup>

<sup>1</sup> Mapua Institute of Technology, Makati City, Philippines  
lavea@mapua.edu.ph

<sup>2</sup> Ateneo de Manila University, Quezon City, Philippines  
mrodrigo@ateneo.edu

**Abstract.** We developed affective models of negative affective states, particularly boredom, confusion, and frustration among novice programming students learning C++ using keyboard dynamics and/or mouse behavior. While key-stroke dynamic features are already sufficient to model negative affect detector, adding mouse behavior, specifically the distance it traveled along the x-axis slightly improved the model's performance. The idle time and typing error are the most notable features that predominantly influence the detection of negative affect. The idle time has the greatest influence in detecting high and fair boredom, while typing error comes before the idle time for low boredom. Conversely, typing error has the highest influence in detecting high and fair confusion, while idle time comes before typing error for low confusion. Though typing error is also the primary indicator of high and fair frustrations, other features are still needed before it is acknowledged as such. Lastly, there is a very slim chance to detect low frustration.

**Keywords:** Affect · Novice programmer · Keyboard dynamics · Mouse behavior · Digraph · Trigraph

## 1 Introduction

Studies about novice programmers found that affect and behavior influence academic performance [1]. Negative affect, particularly boredom and confusion are negatively related to achievement [1, 2] and although frustration was not found to be a predictor of student achievement [1, 2], it can cause students to disengage from or even give up on a programming task.

This study aims to develop affective models of negative affective states among novice C++ programming students using keyboard dynamics and mouse behavior. Specifically, it attempts to answer the following research questions: (1) which features from keyboard dynamics and mouse behavior help in the recognition of boredom, confusion, and frustration? (2) how is student's affect related to keyboard dynamics and/or mouse behavior (3) how are these features different or similar among high/medium/low incidences of boredom, confusion, and frustration?

## 2 Methodology and Findings

We collected mouse-key logs and video logs from 55 novice C++ programming students during a programming activity. We synchronized the video with the mouse-key activity. Trained human observers then tagged video segments with affect labels. We then used this labeled dataset to develop and validate models for boredom, confusion, and frustration using decision tree classifier.

The keyboard dynamics and mouse behavior features that help in the recognition of negative affective states of novice programming students were: the student's typing errors incurred (the number of times the backspace and delete keys were pressed); the length of time the student is idle (not pressing any key in the keyboard); the student's typing variance (his/her typing varies with time); the number of keyevents he/she executed in the keyboard; the total distance the student moved the mouse along the x-axis; the sum of all time durations the student acted on the 1st key of the digraph; the average time duration between the 2nd and 3rd keydown of the trigraph; and the number of times F9 key (shortcut to compile and run the program) was pressed.

Student boredom was related to both keystroke dynamics and mouse behavior. That is, the keyboard has almost no activity while the mouse has a very minimal movement along the x-axis. Student's frustration is like boredom but without mouse features, since for this affect, students tend to release the mouse and scratch their head or do some other hand gestures. There is almost no keyboard activity too since when a student get frustrated, he/she usually pause for a while and do nothing. Lastly, student's confusion is both related to keystroke dynamics and mouse behavior. Results show that there are several indicators when a student is confused.

Idle time and the total mouse movement along the x-axis were the primary indicators of high boredom; idle time and total keyevents for fair boredom; and typing error and idle time for low boredom. On the other hand, typing error is the sole indicator for high confusion. Likewise, fair confusion was manifested by typing error and typing speed while idle time and typing error for low confusion. For high frustration, typing error was the primary indicator when corroborated by typing variance, idle time, and control key presses. Similarly, the primary indicator of fair frustration was typing error when supported by more features such as idle time, number of mouse movement, average distance moved by the mouse, etc. Lastly, there is a very slim chance to detect low frustration.

## References

1. Bosch, N., D'Mello, S., Mills, C.: What emotions do novices experience during their first computer programming learning session? In: Lane, H., Yacef, K., Mostow, J., Pavlik, P. (eds.) AIED 2013. LNCS, vol. 7926, pp. 11–20. Springer, Heidelberg (2013)
2. Rodrigo, M.M.T., Baker, R.S., Jadud, M.C., Amarra, A.C.M., Dy, T., Espejo-Lahoz, M.B.V., Lim, S.A.L., Pascus, S.A.M.S., Sugay, J.O., Tabanao, E.S.: Affective and behavioral predictors of novice programmer achievement. In: ITiCSE 2009 Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education, vol. 41(3), pp. 156–160 (2009). <http://doi.acm.org/10.1145/1562877.1562929>



# What Is More Important for Student Modeling: Domain Structure or Response Times?

Jiří Řihák and Radek Pelánek

Masaryk University Brno, Brno, Czech Republic

**Abstract.** Many features can be considered when designing a student model for an adaptive educational system. What is the relative importance of different modeling aspects? Where should we focus our attention in developing models for real word applications? We report comparison of two aspects: the choice of a domain model and the utilization of response times. The case study (an adaptive system for practice of basic arithmetic) suggests that response times deserve more attention in student modeling.

## 1 Introduction

A student model is a key part of an adaptive educational system. There is wide range of student modeling approaches and many features which can be included. In this work we compared impacts of two selected aspects of student modeling. The first one is the modeling of the domain structure [1], i.e., definition of skills and a mapping between skills and items. The second one is the utilization of response times [2], which is an additional information to the correctness of answers. As a case study we use a real adaptive educational system in its early stage of application, where the choice of a student modeling approach is a real, pressing development issue. We explore a range of domain models and response time uses, discuss their relations and comparison, and study parameter stability.

## 2 Setting

To explore the issue of model selection we utilize data from an adaptive practice application MatMat ([matmat.cz](http://matmat.cz)), which covers the area of basic arithmetic. The system is available freely online and its behaviour and default student model are described in [4]. The currently available data comprise 150000 answers to 2000 items, which are divided into 5 high level concepts (counting, addition, subtraction, multiplication, division).

The used models are extensions of the Elo rating system [3], which can be seen as a heuristic for parameter estimation of the Rasch model. For comparison we used three domain models: the model with a single global skill; the model with skill parameters for each of the 5 main concepts; and the most complex model where skills are described in tree-like structure [4]. To incorporate response times

we combine them with correctness of answer into a single performance measure  $r$ . For correct answers we transform the value 1 into an interval  $[0, 1]$  by one of the following approaches: no use of time:  $r = 1$ ; the discrete decrease:  $r = 1$  for fast responses ( $<7$  s),  $r = 0.5$  for slow responses; the exponential decrease:  $r = 1$  for fast responses,  $r = e^{1-t/7}$  otherwise [4]; and the linear decrease:  $r = \max(0, 1 - t/14)$ .

### 3 Results

For comparison of predictive accuracy of models we use RMSE and AUC with student stratified train/test set division. With respect to domain modeling, the results show that more complex models are able to improve predictions, although increasing complexity of models brings only diminishing improvements. The evaluation of models which consider timing information is more difficult because different models are trained to predict different absolute values. Thus only AUC (which consider only relative order of predictions) seems to be meaningful and according to this metric the best results are achieved using the linear decrease.

To get insight into differences between models we analyze correlations between item difficulty parameters, which have clear interpretation. Unsurprisingly, we found a large gap between the baseline model and other more sophisticated models. The impact of domain modeling is nontrivial, but not pronounced. Different utilization of time, however, brings considerably different parameters. The degree of change is proportional to the intensity of time utilization. Results also suggest that domain modeling and time modeling are almost independent aspects and provide change (and possible improvement) in different directions.

We also studied how many answers are necessary to stabilize these difficulty parameters and found higher increase in stability for models utilizing response times. This increase in stability is probably mainly due to the use of more “bits of information” per each answer.

For the studied case study, the main conclusion is that differences in modeling of response times have larger impact than differences in domain modeling. This result is interesting, since much more research has been devoted to domain modeling than to response times modeling.

### References

1. Desmarais, M.C., d Baker, R.S.J.: A review of recent advances in learner and skill modeling in intelligent learning environments. *User Model. User-Adap. Inter.* **22**(1–2), 9–38 (2012)
2. Klinkenberg, S., Straatemeier, M., Van der Maas, H.L.J.: Computer adaptive practice of maths ability using a new item response model for on the fly ability and difficulty estimation. *Comput. Educ.* **57**(2), 1813–1824 (2011)
3. Pelánek, R.: Application of time decay functions and Elo system in student modeling. In: *Educational Data Mining (EDM)*, pp. 21–27 (2014)
4. Řihák, J.: Use of time information in models behind adaptive system for building fluency in mathematics. In: *Educational Data Mining, Doctoral Consortium* (2015)

# Evaluating Affect in a Learning Environment for Java

Ramón Zatarain-Cabada<sup>1</sup>, María Lucía Barrón-Estrada<sup>1</sup>,  
Francisco González-Hernández<sup>1</sup>, and Carlos A. Reyes-García<sup>2</sup>

<sup>1</sup> Instituto Tecnológico de Culiacán, Culiacán, Sinaloa, Mexico  
rzatarain@itculiacan.edu.mx

<sup>2</sup> Instituto Nacional de Astrofísica, Óptica y Electrónica, Cholula, Mexico

**Abstract.** In this paper we want to show that besides cognitive factors like reasoning and memorization, the motivational factor is a crucial one for learning a first programming language. We have developed and evaluated an affective and intelligent learning environment (ILE) for learning Java. The ILE was tested with high-school students where affect recognizing was only used with university students. We present results and discussions of the experiments.

**Keywords:** Artificial-intelligence · Intelligent-learning-environment · Collaborative-environment · Affective-multimodal-detection

## 1 Introduction

Learning a programming language is historically a complex challenge for non-experienced students because they need to understand new concepts like statements, expressions, data structures, control structures, as well as developing new skills like problem solving, software development, and code debugging. Also, it is important to consider getting skills with respect to the best practices for programming or the correct using of the programming paradigm. Previously, we introduced an Intelligent Learning Environment (ILE) called Java Sensei [1] which performs detection of affect through the features from student faces. We present two new features added to the ILE. The first component added is the social collaboration among students currently known as “Computer-supported Collaborative Learning (CSCL)”. The second component added is multimodal affect recognition. Multimodal recognition systems detect the emotion of students from different devices that interact with them. Thereby, they help to improve the precision and accuracy to detect the emotion state of the student.

## 2 Integration of CSCL and Multimodal Detection in Java Sensei

The collaboration features added to Java Sensei will add more tools that will help the students in their learning processes. With this respect, we implemented three new features: chat room, tips, and social roles. A **Chat room** is where student can talk and debate among themselves when they solve their exercises. **Tips** are affective advices,

recommendations, and comments from the system to group members. Social Roles are a label where the system assigns different responsibilities to some members like **the leader** (takes the last decision in case of disagreement) and **the communicator** (ask for help to solve problems and exercises). Java Sensei implements a semantic algorithm (ASEM) which uses semantic tags to identify the student emotion from text dialogs [2]. We also used an algorithm to enhance/inhibit the value of the emotion. The semantic algorithm uses a corpus of words named SEL. Emotion classification in voice is implemented by using a Support Vector Machine (SVM). In order to get the main features from voice and to train the SVM, the system extracts intensity and rhythm from the audio [3]. We built our own corpus using a group of 20 men and women who were evaluated using the self-assessment manikin (SAM) to categorize the emotion in the audio. Java Sensei uses a set of fuzzy rules for pedagogical evaluation of students. The fuzzy rules have 5 input variables and 3 output variables.

### 3 Experiments, Results, and Conclusions

We conducted two experiments. The first experiment was directed with a group of thirty-two high-school students. The topic evaluated was Java inheritance. We applied a pre-test and post-test evaluation. The experiment was divided in three sessions of 45-min. In first session, the students solved a pre-test on paper. In second session, the students worked with the Java Sensei. In the last session the students solved a post-test on paper. The pre-test obtained a mean value of 46.7 and standard deviation of 12.36. The post-test obtained a mean value of 37.29 and a standard deviation of 10.78. Those values could indicate that apparently the students in the experiment did not have an important change in their outcomes. However, other results where we evaluated the complexity of the tests indicate that both tests were very hard to solve by the students. In the future, we will perform more testing to validate the reliability of the multimodal emotion recognizer as well as the learning gains when student work in collaboration with partners.

### References

1. Zatarain Cabada, R., et al.: An affective learning environment for Java. In: 2015 IEEE 15th International Conference on Advanced Learning Technologies (ICALT). IEEE (2015)
2. Wu, C.-H., Chuang, Z.-J., Lin, Y.-C.: Emotion recognition from text using semantic labels and separable mixture models. *ACM Trans. Asian Lang. Inf. Process. (TALIP)* 5(2), 165–183 (2006)
3. Pan, Y., Shen, P., Shen, L.: Speech emotion recognition using support vector machine. *Int. J. Smart Home* 6(2), 101–107 (2012)

# Implicit Social Networks for Social Recommendation of Scholarly Papers

Shaikhah Alotaibi and Julita Vassileva

Department of Computer Science, University of Saskatchewan,  
Saskatoon, SK, Canada

Shaikhah.otaibi@usask.ca, jiv@cs.usask.ca

**Abstract.** Scholarly papers are an important learning resource for graduate students and researchers. The data available in social bookmarking websites can be exploited to connect similar users with implicit social relations based on their bookmarking behavior. We propose three different implicit social networks and show that they connect users with similar interests while not requiring the availability of user rating data or explicit social connections.

**Keywords:** Implicit social network · Social recommendation · Social bookmarks

## 1 Introduction

As a lifelong learners, researchers consider scholarly papers as learning objects; especially graduate students and young researchers. With the advance of the Web, the number of newly published papers has become overwhelming. For this reason, many recommender systems (RSs) have been proposed to help readers in finding relevant papers to read. By exploiting the social ties of users, RSs can overcome some drawbacks of the most popular recommendation algorithms, for example, the cold start problem [1]. Social bookmarking tools (e.g. CiteULike, Mendeley) have been developed as social tools for users to store and share bookmarks of papers, their bibliographic data, and even paper files, to annotate and organize them, thus accumulating a wealth of user data providing clues for interest similarities between users. However, surprisingly, none of the popular social bookmarking tools have utilized the wealth of social data they store to build a social RS. There are only a few studies that incorporate social relations in the domain of research paper recommendations (e.g. [2, 3]). However, all of them consider the explicit social relationships, which are based on the invitation and its acceptance between users; such explicit social networks are shown to have low user coverage [3]. We want to address this gap by proposing three implicit social networks and we show that they provide useful data to calculate similarity of users' interests.

## 2 Proposed Implicit Social Networks, Experiments and Results

We built three implicit social networks (ISNs) based on bookmarking data crawled from CiteULike. First, the *readership* ISN connects users to the authors of the papers that they have bookmarked in their libraries. The relation could be unidirectional if only one of the users in this relation has bookmarked the other user's publication, and reciprocal if both users have bookmarked each other's publications. Second, the *co-readership* ISN connects users who bookmark papers written by the same authors. Third, the *tag-based* ISN connects users if they use the same tags to annotate their bookmarked papers.

For the first experiment, we conducted a one-way ANOVA which showed that there is statistical difference between the means of interest similarities of connected users among the proposed three ISNs, ***F value*** = 6258.5 at  $p < 0.01$ . Moreover, the Scheffé post hoc pairwise comparisons show that the reciprocal readership network has the highest similarity mean value between connected users, followed by the unidirectional readership network, the co-readership and finally, and the tag-based network. This means that users are more similar to the authors of the papers that they bookmarked in their libraries, especially if the relation is reciprocal. It can also be inferred that the user-generated data (i.e., tags) does not do better than the metadata that is used to construct the other ISNs. Possible reason for that is the average number of tags per user in CiteULike is only 3.81.

In the second experiment a two-way ANOVA test is used which showed that there was statistical difference between the means of interest similarities of directly connected users and indirectly connected users using distances of one hop (one intermediate user) and two hops (two intermediate users) for each of the three ISNs. The ***F values*** are 13.11, 855.38, 1039.02, and 826.78 at  $p < 0.01$  respectively for readership ISN (reciprocal), readership ISN (unidirectional), co-readership ISN and tag-based ISN. The Scheffé post hoc test showed that across all networks, users who are involved in direct relationships have the highest similarity and the similarity decreases with the increase of the social distance. While direct relations between users are beneficial in paper recommendation from the most similar users, indirect relationships can also be usefully deployed to enrich the user's library with diverse and serendipitous papers [4].

The last experiment compared our ISNs with two explicit networks: co-authorship network and connections (friends) network. A co-authorship relationship between two users manifests itself when they collaborate in writing and publishing a research paper (s) indicating strong shared interests. The social relation in the connection network happens between two users based on invitation and its acceptance. The results of comparing the means of interest similarity of users connected using ISNs and explicit social networks showed that there was statistical difference between the mean values of the similarity of the different networks ***F value*** = 4193.99 at  $p < 0.01$ . The Scheffé post hoc test showed that both explicit social networks had lower interest similarity than both readership ISNs, but it was higher than the other ISNs. However, only users

connected using these explicit social networks can receive social recommendations, comprising only 1.873 % of the users in the case of co-authorship network and 18 % of users in the case of connections network.

## References

1. Massa, P., Avesani, P.: Trust-aware recommender systems. In: Proceedings ACM RecSys 2007, pp. 17–24. ACM, New York (2007)
2. Pera, M.S., Ng, Y.-K.: A personalized recommendation system on scholarly publications. In: Proceedings 12th ACM CIKM 2011, pp. 2133–2136. ACM, New York (2011)
3. Lee, D.: Personalized recommendations based on users' information-centered social networks. PhD. thesis, School of Information Sciences, University of Pittsburgh (2013)
4. Granovetter, M.: The strength of weak ties: a network theory revisited. *Sociological Theory*, pp. 105–130 (1982)

# A Context-Based Similarity Algorithm for Enhancing Learning Scenarios Reuse

Mariem Chaabouni<sup>1,2</sup>, Mona Laroussi<sup>2</sup>, Claudine Piau-Toffolon<sup>1</sup>,  
Christophe Choquet<sup>1</sup>, and Henda Ben Ghezala<sup>2</sup>

<sup>1</sup> LIUM, Maine University, Le Mans, France  
{Mariem.Chaabouni, Claudine.Piau-Toffolon,  
Christophe.Choquet}@univ-lemans.fr

<sup>2</sup> RIADI, Manouba University, Manouba, Tunisia  
Mona.Laroussi@univ-lille1.fr,  
Henda.Benghezala@ensi.rnu.tn

## 1 Aims and Motivation

With the evolution and the diversification of their practices, teachers and trainers stress the need of maintaining repositories of learning scenarios, allowing the sharing and the reuse. The representations of these scenarios have been widely treated in prior works such as IMS-LD (IMS - Learning Design) [1] or PoEML (Perspective-oriented Educational Modeling Language) [2]. Several learning scenarios repositories appeared implementing these modeling approaches and containing a large number of scenarios. This leads us to consider the scenario reuse issue that becomes an essential practice with the continued evolution of these repositories.

In a situation of reuse, the teachers design scenarios with having in mind various learning contexts. These scenarios implement different modalities as temporal, spatial, and collaborative modalities. This talk deals with the retrieval of learning scenarios that most fit a given learning situation specification from a contextual perspective. So it presents an overview of a context similarity algorithm using a weighted similarity metric between contextual indexes and a planned given context. The context represents a higher layer treated in a separate way helping the management of contextual requirements and constraints of a scenario. We consider that the context of a learning scenario is “*a set of characteristics and constraints related to the environment which influence, directly or indirectly, the progress of the learning scenario*”.

## 2 Context-Based Similarity Algorithm for Learning Scenarios Reuse Exploiting Contextual Indexes

To proceed to the construction of a contextual index model of a learning scenario, different phases are required: (1) the extraction of the effective context of the learning scenario, (2) the collection of learner’s usage traces of the scenario, (3) the calculation of pedagogical indicators, (4) the analysis of the scenario progress in this precise source



context and then (5) the contextual indexing of the scenario. This process is detailed in prior works [3]. The context model of a learning scenario is enhanced with pedagogical indicators that represent additional calculated variables that reflect the progress of the scenario. Such indicators interpret the effective context and inform about the progress of the scenario in this context. We can refer for example to the collaboration indicator, the learner trajectory indicator or the device state indicator.

The proposed algorithm calculates the similarity between a Planned Context Model (PCM) representing the target context of a learning situation and each Indexing Context Model ( $ICM_j$ ) associated with different capitalized learning scenarios ( $LS_i$ ). The algorithm is based on the weighted DICE similarity [3], a metric calculating the similarity between two weighted vectors. In the following an overview of the main steps of the algorithm.

**Input:** A set of N learning scenarios (**LS**), M index instances (ICM) and associated Success Rates (SR), a Planned Context (PCM) and Min similarity Threshold (MinTh)

**Output:** A list L of the most relevant scenarios (L)

**Procedure:**

- Match correspondent elements and establish links between the planned context PCM and indexes of capitalized learning scenarios  $ICM_i$
- For each  $ICM_i$ :
  - o Verify the presence of all index constraints in the PCM
  - o If all constraints are satisfied: For each matched  $ICM_i$  vectors and PCM vectors: Apply Dice similarity formula, reinforced by the success rate SR associated to the given index
  - o Calculation of similarities Average in the tree (through a bottom-up approach) to retrieve global similarity value  $Similarity(ICM_i, PCM)$
  - o Add  $LS_i$  corresponding to  $ICM_i$  to the L list
- Return L List

An authoring tool “Context-LS”, implementing the algorithm, has been developed for teachers in situations of pedagogical design by reuse. This tool recommends scenarios that can be the most adapted to the context of the target learning situation. The tool also offers functionalities allowing the teacher-designer to analyze the context and the scenario progress through calculated indicators, and then indexing the scenario for future reuse.

## References

1. IMS Global Learning Consortium: IMS learning design specification. 2008206225 (2003). <http://www.imsglobal.org/learningdesign/index.html>
2. Caero-Rodríguez, M., Marcelino, M.J., Llamas-Nistal, M., Anido-Rifón, L., Mendes, A.J.: Supporting the modeling of flexible educational units. J. Univ. Comput. Sci. **13**(7), 980–990 (2007)

3. Chaabouni, M., Piau-Toffolon, C., Laroussi, M., Choquet, C., Ben Ghezala, H.: Indexing learning scenarios by the most adapted contexts: an approach based on the observation of scenario progress in session. In: 15th International Conference on Advanced Learning Technologies, pp. 39–43. IEEE (2015)
4. Jusselme, A.L., Maupin, P.: Distances in evidence theory: comprehensive survey and generalizations. *Int. J. Approximate Reasoning* **53**(2), 118–145 (2012)

# Towards the Recommendation of Resources in Coursera

Carla Limongelli<sup>1</sup>, Matteo Lombardi<sup>2</sup>, and Alessandro Marani<sup>2</sup>

<sup>1</sup> Engineering Department, Roma Tre University,  
Via Della Vasca Navale, 79, 00146 Rome, Italy  
`limongel@ing.uniroma3.it`

<sup>2</sup> School of Information and Communication Technology, Griffith University,  
170 Kessels Road, Nathan, Brisbane, QLD 4111, Australia  
`{matteo.lombardi,alessandro.marani}@griffithuni.edu.au`

**Abstract.** Technology Enhanced Learning (TEL) largely focuses on the retrieval and reuse of educational resources from Web platforms like Coursera. Unfortunately, Coursera does not provide educational meta-data of its content. To overcome this limitation, this study proposes a data mining approach for discovering Teaching Contexts (TC) where resources have been delivered in. Such TCs can facilitate the retrieval of resources for the teaching preferences and requirements of teachers.

**Keywords:** MOOCs · Educational data mining · Coursera

## 1 Introduction

Many contributions in TEL propose Information Retrieval (IR) systems of educational web resources [4, 5] and most of these works are focused on students; only some recent contributions address the role of instructors [2–4]. The web hosts many e-learning platforms that help instructors in delivering their resources or courses. Particularly attractive is the idea of Massive Open Online Course (MOOCs) where courses are delivered and publicly available worldwide. Coursera is an on-line platform with plenty of reliable MOOCs delivered by prestigious universities: a very attractive source of educational data. However, it does not offer educational meta-data of its content, so this paper suggests a clustering technique for deducing some representative TCs useful for IR systems in TEL. In this contribution, a TC consists of (i) the teaching preferences of an instructor, (ii) course information and (iii) lesson information.

## 2 Clustering Coursera Data

This study suggests to apply Hierarchical Clustering (HC) on DAJEE [1], which is a TEL dataset built from Coursera data. The TCs are deduced from the analysis of the three main educational entities or hierarchies in the dataset:

instructors, courses and lessons where the resources have been delivered in. The features obtained after the pre-processing of data are the following: *average duration of resources*, *average number of resources per concept* and *average duration of concepts* for instructors, *duration* and *semantic density*<sup>1</sup> for both courses and lessons. To identify distinctive educational models at each level, it is suggested to cluster data in an hierarchical manner.

1. **Instructors:** for clustering instructors instances, both K-Means (with K from 2 to 242) and Expectation-Maximization (EM) have been run. The best configuration indicated by the Calinski-Harabasz (CH) index [6] is K-Means with  $K = 2$ .
2. **Courses:** each cluster of instructors is further divided considering courses models taught by instructors in a same cluster. It is harder to suggest a value of  $K$  for K-Means that is appropriate for any change introduced by the first level of clustering. Moreover, the CH index is sensible to the data [6], so CH cannot be used for finding  $K$  once-for-all. Therefore, we suggest EM for this level, so that the most appropriate mixture models are defined for any partition of data derived by the upper level.
3. **Lessons:** each course cluster is split using the models of lessons; this is the same situation of the upper level, so EM is suggested.

Finally, the resources used for lessons in a same lesson cluster are grouped together. A total of 27 clusters indicate the most representative TCs in Coursera.

### 3 Conclusions

With the proposed HC method, 27 TCs have been discovered from data in Coursera. These contexts can be used for retrieving resources from MOOCs appropriate for the specific teaching situation of an instructor. In the near future, this assumption has to be proved with a large experimentation of an IR system based on our method.

### References

1. Estivill-Castro, V., Limongelli, C., Lombardi, M., Dajee, A.M.: A dataset of joint educational entities for information retrieval in technology enhanced learning. In: Proceedings of the 39th International ACM SIGIR Conference. ACM (2016)
2. Limongelli, C., Lombardi, M., Marani, A., Sciarrone, F.: A teacher model to speed up the process of building courses. In: Kurosu, M. (ed.) HCII/HCI 2013, Part II. LNCS, vol. 8005, pp. 434–443. Springer, Heidelberg (2013)
3. Limongelli, C., Lombardi, M., Marani, A., Sciarrone, F.: A teaching-style based social network for didactic building and sharing. In: Lane, H.C., Yacef, K., Mostow, J., Pavlik, P. (eds.) AIED 2013. LNCS, vol. 7926, pp. 774–777. Springer, Heidelberg (2013)

---

<sup>1</sup> In this work, semantic density is the ratio of number of concepts on the duration of the educational entity (i.e. courses or lessons) following IEEE 1484.12.1-2002.

4. Limongelli, C., Lombardi, M., Marani, A., Sciarrone, F., Temperini, M.: A recommendation module to help teachers build courses through the moodle learning management system. In: *New Review of Hypermedia and Multimedia*, pp. 1–25 (2015)
5. Lombardi, M., Marani, A.: A comparative framework to evaluate recommender systems in technology enhanced learning: a case study. In: Pichardo Lagunas, O., Herrera Alcántara, O., Arroyo Figueroa, G., et al. (eds.) *MICAI 2015. LNCS*, vol. 9414, pp. 155–170. Springer, Heidelberg (2015). doi:[10.1007/978-3-319-27101-9\\_11](https://doi.org/10.1007/978-3-319-27101-9_11)
6. Maulik, U., Bandyopadhyay, S.: Performance evaluation of some clustering algorithms and validity indices. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(12), 1650–1654 (2002)

# Triangle Block Model for Bridging Conceptual Representation to Numerical Representation in Arithmetic Word Problems: A Brief Report of Practical Use by Fourth Grade Students

Tsukasa Hirashima<sup>1</sup>, Kazutoshi Furukubo<sup>1</sup>, Yusuke Hayashi<sup>1</sup>,  
Sho Yamamoto<sup>1</sup>, and Kazushige Maeda<sup>2</sup>

<sup>1</sup> Graduate School of Engineering, Hiroshima University, Hiroshima, Japan  
tsukasa@le1.hiroshima-u.ac.jp

<sup>2</sup> Attached Elementary School, Hiroshima University, Hiroshima, Japan

**Abstract.** This paper is a brief report of a practical use of triangle block model for learning of arithmetic word problems. The triangle block model has been proposed as a bridging model between conceptual representation of a word problem and Numerical representation of its solution. Based on this model, we have developed an interactive environment where a pupil is able to manipulate an integrated representation of the conceptual and quantitative ones as learning of arithmetic word problems. We also designed a series of lessons to use the environment. The lessons were practically conducted for 75 4<sup>th</sup> grade pupils (in two classes) in an elementary school for 7 class times. As the results, it was confirmed that (1) the pupils and their responsible teacher accepted the lessons as useful ones, and (2) learning effect as improvement of structural understanding for the problems.

**Keywords:** Problem comprehension · Bridging model · Word problem

## 1 Introduction

Triangle Block Model has been proposed as a bridging representation between conceptual representation and numerical representation. We call the bridging representation “CN representation” in this paper. Triangle Block Model satisfies following requirements: (1) a pupil is allowed to build CN representation, (2) concepts constituting both CN representation and the problem representation are the same ones, and (3) CN representation is able to be diagnosed [1]. This paper is a brief report of a practical use of an interactive learning environment designed based on Triangle Block Model. Target students are 4<sup>th</sup> grade pupils in an elementary school who start to learn arithmetic word problems that are solved by using multiple arithmetic operations. One lesson is composed of (I) teacher’s teaching of arithmetical word problems with the model and (II) exercises with an interactive learning environment developed based on the model. In the environment, a pupil is able to manipulate Triangle Block Model. In order to carry out the both steps in the same usual classroom in a seamless way, the learning environment has been implemented on a tablet PC. The series of lessons was

conducted for 75 4<sup>th</sup> grade pupils in two classes. Through this practical use, we have confirmed that the teaching and environment designed based on the model were accepted as a useful method and tool for learning word problems by the pupils and their responsible teacher. We have also found the learning effect of the activities. The detailed of the practical use will be reported in another paper. In this paper, triangle block model is briefly introduced.

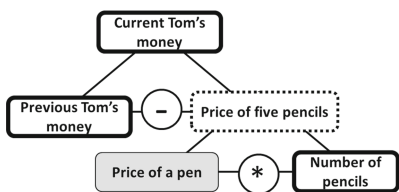


Fig. 1. Combined Triangle Block.

## 2 Triangle Block Model

Figure 1 shows an example of CN representation of an arithmetic word problem with Triangle Block Model. The problem is as follows: *Tom had fifteen dollars. Tom bought 5 pencils. Tom has ten dollars. What was the price of each pencil?* The representation is composed of two triangles. Each triangle is called “triangle block”. The representation is called “combined triangle block”. A triangle block is composed of several triangle blocks. Then, a triangle block is composed of three numerical concepts and one arithmetic operation and it expresses a basic problem that is solved by an arithmetic operation. This means that a basic problem is represented by three numerical concepts. We call this model “triplet sentence model” [2]. As our previous research, an interactive environment for learning by problem-posing based on this model has been developed [3]. This research is an extension of the model to more complex problem that is solved by more than one arithmetic operation. A numerical concept has a conceptual label (ex. “number of pencils”) and a numerical value (ex. 5). In the learning environment, a student is provided (1) a problem, (2) several numerical concepts, (3) arithmetic operations with a triangle frame, and then makes combined triangle block expressing the conceptual and numerical representation of the provided problem. The environment diagnoses the combined triangle block and gives feedback to the student based on the diagnosis.

## 3 Conclusion

This paper is a brief report of our investigation about bridging model of between conceptual representation and numerical representation in arithmetic word problem. Detailed of the practical use will be reported in another paper.

## References

1. Hirashima, T., Hayashi, Y., Yamamoto, S., Maeda, K.: Bridging model between problem and solution representations in arithmetic/mathematics word problem. In: Proceedings of ICCE 2015, pp. 9–18 (2015)
2. Hirashima, T., Yokoyama, T., Okamoto, M., Takeuchi, A.: Learning by problem-posing as sentence-integration and experimental use. In: AIED 2007, pp. 254–261 (2007)
3. Hirashima, T., Yamamoto, S., Hayashi, Y.: Triplet structure model of arithmetical word problems for learning by problem-posing. In: Yamamoto, S. (ed.) HCI 2014, Part II. LNCS, vol. 8522, pp. 42–50. Springer, Heidelberg (2014)



# POLARISQL: An Online Tutoring System for Learning SQL Language

Soraya Chachoua, Jamal Malki, and Pascal Estrailier

L3I Laboratory, University of La Rochelle, La Rochelle, France  
{soraya.chachoua,jamal.malki,pascal.estrailier}@univ-lr.fr

In this paper, we describe our online tutor for SQL language learning called *POLARISQL*<sup>1</sup>. This online tutor was developed by our team in the framework of a global e-learning project environment called *POLARIS*<sup>2</sup> in *L3I*<sup>3</sup> laboratory. The main role of an online tutor is to facilitate and to support students in the process of knowledge acquisition. It allows one-to-one individualized training [3, 4].

## 1 System Requirements

In this context, the specifications of such a system must meet the requirements of the various system's users. As an example, the adaptability notion needs a thorough knowledge of the learner profile, history, evolution, etc., and the learning model. Therefore, the concept of the learning model will introduce a new working methodology mainly based on activities such as *Lecture course*, *Directed working group*, *Laboratory course* and *Tutorial course*.

## 2 Functional Components

Figure 1 shows POLARISQL tutor portal webpage loaded in a browser. Through this screenshot the user (student, teacher, etc.) can authenticate by inserting a database name, a user name and password<sup>4</sup>. After a successful authentication the user will be directed to the workspace page of the platform as illustrated in Fig. 2. The platform's workspace contains several frames:

- frame 1: working document is an SQL editor to write code;
- frame 2: working document's actions: in the latter, users could perform several actions such as: Edit a working document, execute a (script, line or selected lines), download a content, (save, upload and create) a working document.
- frame 3: output document: displays the queries result execution.
- frame 4: output document's actions: users can manage an output document by different actions such as: clear and save an output document and disconnect from the platform.

---

<sup>1</sup> <http://polarisql.univ-lr.fr>.

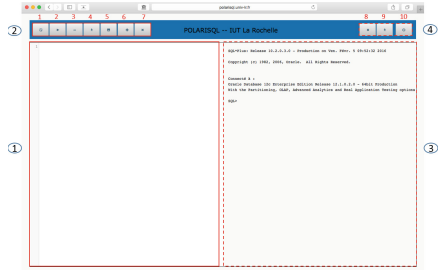
<sup>2</sup> <http://l3i-education.univ-lr.fr/portail/>.

<sup>3</sup> <http://l3i.univ-larochelle.fr/>.

<sup>4</sup> Demo account : Database Name=MODELE, Username=demo, Password=demo.



**Fig. 1.** POLARISQL's authentication page



**Fig. 2.** POLARISQL's workspace page

### 3 Design and Implementation

The design of POLARISQL is inspired from the traditional teaching processes carried out during practical working classes: query formulation, query execution and results visualization. The backend of POLARISQL platform takes an SQL program as input and produces its execution over the DBMS. At the end of each working session we obtain two kinds of traces, namely: execution traces and application traces [1, 2]. The frontend is a website located in POLARISQL URL as illustrated in Fig. 1. We implemented the design layout of our platform inspired from the different authoring systems used during a traditional practical work class such as text editor, execution terminal and oracle server user interface.

### 4 Discussion and Perspective

POLARISQL is a web-based platform which takes in the working document (text editor) as input an SQL program code, then executes it in the backend and displays results in the output document. Moreover, the platform retrieves all the activities' traces in order to allow an evaluation process. The latter, helps to adapt resources and learning strategies for the students.

### References

1. Courbet, D.: Les applications des sciences humaines à la publicité: De la psychanalyse à la socio-cognition implicite et au neuromarketing. *Humanisme et entreprise* **276** (2006)
2. Djouad, T., Settouti, L.S., Prié, Y., Reffay, C., Mille, A.: Un système à base de traces pour la modélisation et l'élaboration d'indicateurs d'activités éducatives individuelles et collectives. mise à l'épreuve sur moodle. Mise à l'épreuve sur Moodle. *Technique et Science Informatiques* (2010)
3. Etienne, W.: *Artificial intelligence and tutoring systems* (1987)
4. Ohlsson, S.: Some principles of intelligent tutoring. *Artif. Intell. Educ.* **1**, 203–238 (1987)

# Empirical Evaluation of Intelligent Tutoring Systems with Ontological Domain Knowledge Representation: A Case Study with Online Courses in Higher Education

Ani Grubišić<sup>1</sup>, Slavomir Stankov<sup>2</sup>, Branko Žitko<sup>1</sup>, Suzana Tomaš<sup>3</sup>, Emil Brajković<sup>4</sup>, Tomislav Volarić<sup>4</sup>, Daniel Vasić<sup>4</sup>, and Ines Šarić<sup>1</sup>

<sup>1</sup> Faculty of Science, University of Split, Split, Croatia  
{ani.grubisic,branko.zitko,ines.saric}@pmfst.hr

<sup>2</sup> Retired Full Professor  
slavomirstankov@gmail.com

<sup>3</sup> Faculty of Philosophy, University of Split, Split, Croatia  
suzana.tomas@ffst.hr

<sup>4</sup> Faculty of Science and Education,  
University of Mostar, Mostar, Bosnia and Herzegovina  
{emilbrajko,tvolaric,daniel}@fpmoz.ba

**Abstract.** We present results of empirical evaluation of intelligent tutoring systems (ITS) with ontological domain knowledge representation. This research was done as a first step in the process of developing a new model of intelligent tutoring system that will include all the characteristics of evaluated systems: adaptive content, communication based on controlled natural language, graphical presentation of ontological domain knowledge representation. The case study results revealed extraordinary effectiveness of evaluated adaptive intelligent tutoring systems when compared with traditional learning and teaching process.

**Keywords:** Intelligent tutoring systems · Adaptive courseware · Natural language processing · Ontology · Effectiveness evaluation

## 1 Introduction

Design and implementation of ITSs in modern conditions, takes place under the strong influence on natural language processing and natural language communication, along with courseware that adapts learning contents to current level of student's knowledge. The authors are university level teachers with more than a decade of experience in research, development and application of ITSs (TEx-Sys [1], CoLaB Tutor [2], ACware Tutor [3], CM Tutor [4]). We plan to develop of a new fully automated ITS which will be able to tutor any declarative domain knowledge and to communicate on natural language.

## 2 Research Methodology, Results and Findings

Students who participated in this case study were undergraduate and graduate students from three faculties (Faculty of Science and Faculty of Philosophy, University of Split, Croatia and Faculty of Science and Education, University of Mostar, Bosnia and Herzegovina). The case study started on 9th of November 2015 and lasted until the 4th of December 2015.

After a short introduction, the pre-test was conducted. Following the pre-test, a brief introduction into administrative issues related to the treatments, was given. After the pre-test, we have randomly divided students into one control and the treatment groups using the caliper matching with  $\pm 7$  points range. The students from the treatment groups used the AC-ware Tutor (T1), the CoLab Tutor (T2) and the CM Tutor (T3) in learning and teaching process and the students from the control group (C) were taught by live teacher in classroom.

After completing the learning and teaching process, all groups performed the post-test. Null-hypotheses are that there are no significant differences between the control group C and the treatment groups T1, T2 and T3 (NH01, NH02, and NH03).

The one-way ANOVA has confirmed that there is no statistically significant difference between the control and treatment groups' mean values concerning pre-test results ( $F = 0,842$ ,  $p\text{-value} = 0,474$ ).

We have calculated the large effect size for all of the observed systems: AC-ware 0,586, CoLab Tutor 1,443 and CM Tutor 1,063. None of the confidence intervals includes zero, so we can say that the resulting effect sizes are statistically significant.

Therefore, the AC-ware Tutor, the CM Tutor and the CoLab Tutor can be used in a case the teacher is unavailable, because the advantage of this system is the possibility of learning any-where, any-place and any-time.

The results of this empirical evaluation have shown that the observed intelligent tutoring systems based on ontological domain knowledge representation are effective when compared with traditional learning and teaching process.

Since the case study has shown great effect sizes and promising student feedback, we will use these research findings for developing a new and unique model of intelligent tutoring system. This new ITS will include all the characteristics of evaluated intelligent tutoring systems: adaptive content, communication based on controlled natural language, graphical ontological domain knowledge presentation.

**Acknowledgements.** The paper is part of the work supported by the Office of Naval Research grant No. N00014-15-1-2789.

## References

1. Stankov, S., Rosic, M., Zitko, B., Grubisic, A.: TEx-Sys model for building intelligent tutoring systems. *Comput. Educ.* **51**, 1017–1036 (2008)
2. Žitko, B.: Model of intelligent tutoring systems based on controlled knowledge processing over ontology (2010)
3. Grubišić, A.: Adaptive student's knowledge acquisition model in e-learning systems (2012)
4. Volarić, T.: Design of lesson model in intelligent e-learning system. Mechanical Engineering and Naval Architecture, University of Split, Croatia, Faculty of Electrical Engineering (2014)

# Adaptive Testing by Bayesian Networks with Application to Language Assessment

Francesca Mangili<sup>1</sup>, Claudio Bonesana<sup>1</sup>, Alessandro Antonucci<sup>1</sup>,  
Marco Zaffalon<sup>1</sup>, Elisa Rubegni<sup>2</sup>, and Loredana Addimando<sup>2</sup>

<sup>1</sup> Istituto Dalle Molle di Studi Sull'Intelligenza Artificiale (IDSIA), USI-SUPSI,  
Manno, Switzerland

{francesca,claudio,alessandro,zaffalon}@idsia.ch

<sup>2</sup> Scuola Universitaria Professionale Della Svizzera Italiana (SUPSI),  
Manno, Switzerland

{elisa.rubegni,loredana.addimando}@supsi.ch

**Abstract.** We present a general procedure for computerized adaptive testing based on probabilistic graphical models, and show on a real-world benchmark how this procedure can increase the internal consistency of the test and reduce the number of questions without affecting accuracy.

**Keywords:** Computerized adaptive testing · Bayesian networks · Entropy

## 1 Introduction

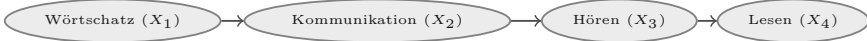
The goal of *Computer Adaptive Testing* (CAT) is to reduce the assessment time and to challenge test takers by adapting the sequence of questions to their ability level. *Item Response Theory* (IRT) is CAT traditional background. *Bayesian networks* (BNs) can offer IRT a powerful language for describing dependencies between skills and modeling richer tasks [1]. Although several researchers have explored BNs in educational assessment, real-world applications and extensive studies of their effectiveness are hardly found in the literature. In this work, we present a general procedure for BNs-based CAT and we test it in a real-world benchmark about German language proficiency assessment.

## 2 Adaptive Testing by Bayesian Networks

Students skills are modeled by a set  $\mathbf{X} := (X_1, X_2, \dots, X_n)$  of categorical variables whose joint *probability*  $P(\mathbf{X})$  is described by a BN through (i) a directed acyclic graph whose nodes represent the variables in  $\mathbf{X}$ ; (ii) conditional probability tables (CPTs)  $P(X_i | \Pi_{X_i})$ ,  $i = 1, \dots, n$ , where  $\Pi_{X_i}$  is the joint variable of the *parents* (i.e., the immediate predecessors) of  $X_i$  (see, e.g., Fig. 1 for the model used in the German language assessment). We point the reader to [2] for the theoretical notions about BNs. To evaluate the informativeness level about  $\mathbf{X}$  provided by  $P$ , we adopt the *entropy*  $H(\mathbf{X}) := -\sum_{\mathbf{x}} P(\mathbf{X}) \cdot \log P(\mathbf{X})$ . Low entropy

levels indicate high informativeness. To evaluate the student we formulate a number of *questions*, described as a collection of variables  $\mathbf{Y} := (Y_1, \dots, Y_m)$ . Each question node is represented as a leaf child of the background skills “required” to answer it.

To make our approach adaptive, we chose the  $(k + 1)$ -th question to be asked based on the  $k$ -th previous answers  $y_1, \dots, y_k$ , by minimizing the conditional entropy  $H(\mathbf{X}|y_1, \dots, y_k, Y_{k+1}) := -\sum_{y_{k+1}} H(\mathbf{X}|y_1, \dots, y_k)P(y_{k+1})$ . Finally, we stop the test when the entropy  $H(\mathbf{X}|y_1, y_2, \dots, y_n)$  is sufficiently low.



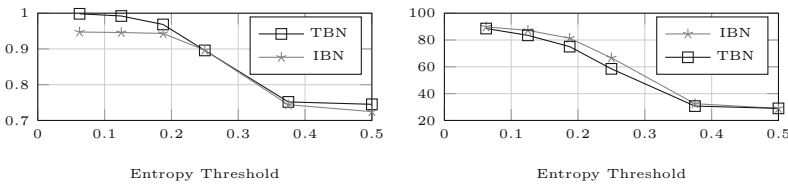
**Fig. 1.** Graph of a BN for German language skills.

*An Application to Language Assessment.* We use the answers of 170 students to 95 questions about German language to reproduce our CAT approach. The *Traditional Evaluation Method* (TEM) assigns to each skill a level A1, A2, B1, B2 by setting thresholds on the fraction of correct answers. We compare TEM with the *independent skills* model (IBN) and the *tree* (TBN) topology in Fig. 1.

Table 1 shows in the non-adaptive case the relative agreement between the TEM, IBN and TBN, and the internal consistency of the three tests evaluated using the split-half methodology. Both BN approaches have larger reliability than TEM. Concerning the adaptive case, Fig. 2 shows the relative agreement

**Table 1.** Relative agreement between models and their split-half reliability

Algorithm	Relative agreement					Algor	Split-half reliability				
	Wört.	Kom.	Hör.	Les.	All		Wört.	Kom.	Hör.	Les.	All
TEM/IBN	.80	.87	.89	.85	.85	TEM	.28	.82	.88	.79	.84
TEM/TBN	.79	.87	.88	.83	.84	IBN	.71	.89	.83	.87	.90
IBN/TBN	.98	.95	.94	.92	.95	TBN	.79	.91	.87	.89	.92



**Fig. 2.** Agreement with the non-adaptive TBN (left) and average number of questions asked by the adaptive methods (right)

of the adaptive IBN and TBN with the non-adaptive TBN, and the average number of questions asked. Both models show a strong reduction in the number of questions as the entropy threshold increases. For instance, using the TBN model, we can save 20 questions on average at the price of only a 3% accuracy reduction. This shows that a relevant number of question are little informative and could be avoided by means of an adaptive approach.

## References

1. Almond, R.G., Mislevy, R.J.: Graphical models and computerized adaptive testing. *Appl. Psychol. Meas.* **23**(3), 223–237 (1999)
2. Koller, D., Friedman, N.: *Probabilistic Graphical Models: Principles and Techniques*. MIT Press (2009)

# When the Going Gets Tough...: Challenge, Emotions, and Difference of Perspective

Naomi Wixon<sup>1</sup>, Sarah Schultz<sup>1</sup>, Danielle Alessio<sup>2</sup>, Kasia Muldner<sup>3</sup>, Winslow Burlesson<sup>4</sup>, Beverly Woolf<sup>2</sup>, and Ivon Arroyo<sup>1</sup>

<sup>1</sup> Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA, USA  
{mwixon, seschultz, iarroyo}@wpi.edu

<sup>2</sup> University of Massachusetts-Amherst,  
140 Governors' Drive, Amherst, MA, USA

allessio@educ.umass.edu, bev@cs.umass.edu

<sup>3</sup> Carleton University, 1125 Colonel by Drive, Ottawa, ON, Canada  
kasia.muldner@carleton.ca

<sup>4</sup> New York University, 665 Broadway, 11th Floor, New York, NY, USA  
wb50@nyu.edu

**Abstract.** We examine students' self-report of task difficulty in conjunction with self-reported affect state, performance, and calculated item difficulty. Students who self-reported having a growth mindset learning orientation behaved more similarly to how we would expect students with a performance goal orientation to behave. Further, by distinguishing between excitement and interest as separate emotions our data suggest a possible refinement of flow state.

## 1 Introduction

An open question is how individual differences between students influence their reaction to challenge, and in particular if they find it daunting or engaging. Flow theory [1] suggests an interaction between challenge and emotion: too much challenge leads to anxiety or frustration and too little to boredom, appropriate challenge may result in engagement. Dweck's theory [2] posits that students with a growth mindset, ones who view intelligence as a dynamic trait that can be increased will seek out challenge, viewing struggle as an opportunity for improvement. Students who view intelligence as a characteristic that can be grown or developed over time are seen as valuing learning and challenge over high performance with relative ease. We expected students who have a growth-mindset will have more interest and excitement when facing challenging situations inside of the tutor, regardless of success.

## 2 Methods and Results

Students' reported student emotional state on a scale from 1 (least) to 5 (most). In addition, students were asked to explain in text why they felt the way they reported.

The open responses were given single-word tags of whether students attributed their feelings to work being "easy" or "hard" [3]. With regard to self-reported



**Table 1.** Correlations among Measures – Student attribution of difficulty (easy/hard), item difficulty on preceding item (Diff) and averaged over time (ADiff), % of Problems Solved on first attempt (Right), and Avg of wrong attempts per problem (AWrong) as correlated with self-reported values of emotion (Frustration, Confidence, Interest “int”, & Excitement “exc”) by median performance (Hi/Low Perf) and having a growth mindset (Growth) or not (NoGrowth)

	Easy	Hard	Diff	ADiff	Right	AWrong
Frustration (N = 131)	-.370**	.060	-.109	-.068	-.147*	.165*
Confidence (N = 124)	.388**	.012	-.079	.090	.108	-.265**
HiPerfExc (N = 125)	-.106	.162*	.033	.073	-.242**	.265**
HiPerfInt (N = 122)	-.096	-.205**	.033	-.018	-.093	.164*
LowPerfExc (N = 116)	.189**	.038	.067	.002	.066	-.083
LowPerfInt (N = 130)	.088	.029	.166*	.101	-.318**	.040
GrowthExc (N = 58)	.112	-.247*	-.140	.258**	.263**	.005
GrowthInt (N = 62)	.004	-.208*	-.044	.169	.331**	-.368**
NoGrowthExc (N = 152)	.021	.008	-.018	-.061	.196**	-.041
NoGrowthInt (N = 166)	.084	-.119	.178**	-.033	-.040	-.068

\*\*p < 0.05, \*p < 0.1

excitement and interest, reports were split by median into “HiPerfExc/Int” and “LowPerfExc/Int”. We found that among “HiPerf” instances wrong answers and right answers were positively and negatively correlated with both excitement and interest respectively. However self-reports of these emotions where performance was below the median did not show significant correlations in the opposite direction as hypothesized base on Csikszentmihalyi’s theory of flow [1]. Students with a growth mindset had marginally significant negative correlations between excitement and interest and perceptions of difficulty. Additionally, they seemed to find solving problems on the first attempt to be significantly more exciting and interesting, and found multiple attempts to solve a problem disinteresting. These findings are counter to our expectation that students who identified with a growth mindset would find a challenging learning process more interesting and exciting.

**Acknowledgement.** This research was supported by the National Science Foundation (NSF) NSF #1324385 IIS/Cyberlearning DIP: Collaborative Research: Impact of Adaptive Interventions on Student Affect, Performance, and Learning. Any opinions, findings, and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF.

## References

1. Csikszentmihályi, M.: Flow: The Psychology of Optimal Experience. Harper & Row (1990)
2. Dweck, C.: Mindset: The new psychology of success. Random House, New York (2006)
3. Schultz, S., Wixon, N., Alessio, D., Muldner, K., Burseson, W., Woolf, B., Arroyo, I.: Blinded by science?: exploring affective meaning in students’ own words. In: 13<sup>th</sup> International Conference on Intelligent Tutoring Systems (accepted)

# Lost in Springdale: An Interactive Narrative for Adult Literacy Learners

Amy M. Johnson<sup>1</sup>, Matthew E. Jacovina<sup>1</sup>, G. Tanner Jackson<sup>2</sup>,  
Elizabeth L. Tighe<sup>1</sup>, and Danielle S. McNamara<sup>1</sup>

<sup>1</sup> ISTL, Arizona State University, Tempe, AZ 85287, USA  
{amjohn43, Matthew.Jacovina, etighe,  
Danielle.McNamara}@asu.edu

<sup>2</sup> Cognitive Science, Educational Testing Service, Princeton, NJ 08541, USA  
gtjackson@ets.org

**Abstract.** This paper reports development of an interactive narrative to be added to the existing intelligent tutoring system, the Interactive Strategy Trainer for Active Reading and Thinking-2 (iSTART-2). The design of this module is specifically tailored for adult learners who read below functional literacy levels. Little research focuses on the educational needs of adult literacy learners, but available evidence indicates they require support for both lower- and higher-order reading skills. The interactive narrative provides varied practice opportunities supporting these skills and uses life-relevant texts which hold personal significance for adult readers. We describe the design and development of the interactive narrative, and preliminary results from a small pilot study indicating overall enjoyment of the narrative.

**Keywords:** Adult literacy · Interactive narrative · Reading instruction

## 1 Interactive Narrative

This paper describes the development of a new interactive narrative within the Interactive Strategy Trainer for Active Reading and Thinking-2 (iSTART-2), which provides self-explanation strategy instruction to improve reading comprehension for high school students [1, 2]. The design of the interactive narrative was informed by prior recommendations [3] and utilizes the following key ingredients we believe will ensure its effectiveness in improving adult learners' reading comprehension: (1) a storyline adaptive to learner decisions; (2) artifacts that are life-relevant to adult learners; (3) content to help develop life skills; (4) pronunciation scaffolding and choice of auditory text presentation; (5) practice increasing in difficulty across the storyline; (6) a variety of interaction methods and response types; (7) motivational elements to enhance effort and persistence; and (8) an open learner model to promote reflection on learning.

'Lost in Springdale' is an interactive first person narrative developed to address unique needs of adult readers who read below functional literacy levels (i.e., below an eighth grade level). Throughout the narrative, the learner encounters various artifacts (e.g., school map, fire extinguisher instructions, update from the Centers for Disease Control [CDC], emails/letters), which attempt to serve learning, assessment, and

engagement goals. The artifacts typically require the learner to read a text, then answer a question, self-explain the text, ask a question, provide a summary, or make a decision about where to go next. Because learners are motivated by instruction that holds personal significance [4], media that are life-relevant to adult readers are utilized. Additionally, because this population shows particular difficulty with phonological awareness and fluency [5], pronunciation scaffolding and auditory presentations are used. Finally, using storyline adaptive to learner decisions, the consequences of poor comprehension are simulated in a realistic context, without actual threats to livelihood.

## 2 Pilot Study on Interactive Narrative

We conducted a small pilot study to examine learners' attitudes toward the interactive narrative, using 17 undergraduate psychology subject pool participants. After completing the narrative, they responded to several survey questions regarding perceptions of the narrative. Results revealed generally positive overall perceptions of the narrative. Over 70 % of participants responded either 'agree' or 'strongly agree' to several favorable statements about the narrative. In addition, learner ratings indicated these learners found the learning tasks fairly easy. Given the narrative is developed for adult literacy learners, these difficulty ratings from college students are encouraging. Open response items suggested these learners felt the narrative supported their reading comprehension skills and that they enjoyed the characters used in the story.

## 3 Conclusion

Results from the pilot study indicate overall user satisfaction of the experience with the interactive narrative. Because the pilot study was small, and conducted using college students, implications of these results for adult literacy learners are difficult to discern, but they provide some tentative support for the usefulness of an interactive narrative to engage adult learners with the educational materials. The next step for the project is to pilot the narrative with students in local adult literacy courses to establish user satisfaction and engagement in the target population.

## References

1. McNamara, D.S., O'Reilly, T., Best, R., Ozuru, Y.: Improving adolescent students' reading comprehension with iSTART. *J. Educ. Comput. Res.* **34**, 147–171 (2006)
2. Jackson, G.T., McNamara, D.S.: Motivation and performance in a game-based intelligent tutoring system. *J. Educ. Psychol.* **105**, 1036–1049 (2013)
3. Lesgold, A.M., Welch-Ross, M. (eds.): *Improving Adult Literacy Instruction: Options for Practice and Research*. National Academies Press (2012)
4. Guthrie, J.T., et al.: Growth of literacy engagement: changes in motivations and strategies during concept-oriented reading instruction. *Reading Res. Q.* **31**, 306–332 (1996)
5. Greenberg, D., Ehri, L.C., Perin, D.: Are word-reading processes the same or different in adult literacy students and third–fifth graders matched for reading level? *J. Educ. Psychol.* **89**, 262–275 (1997)

# An Investigation of Learner's Actions in Problem-Posing Activity of Arithmetic Word Problems

Ahmad Afif Supianto<sup>1,2</sup>, Yusuke Hayashi<sup>1</sup>, and Tsukasa Hirashima<sup>1</sup>

<sup>1</sup> Graduate School of Engineering, Hiroshima University, Hiroshima, Japan  
{afif, hayashi, tsukasa}@lel.hiroshima-u.ac.jp

<sup>2</sup> Department of Informatics, Brawijaya University, Malang, Indonesia

**Abstract.** This study investigated the intermediate product on the way to pose problems as well as the posed problems as the product of problem-posing activity. Every single action of learners during pose the problems have been analyzed. Pearsons' correlation test between the number of actions and the validity of posed problems is conducted. Significant difference is found, which shows that many actions performed by learners tried to satisfy high validity product. Learners try to pose problems with high validity product and they had a tendency to enhance the validity of posed problems.

**Keywords:** Problem-posing process · Arithmetic word problems · Problem states space · Learning analytics

## 1 Introduction

Several investigations have been confirmed that learning by problem-posing in conventional classrooms is promising activity in learning mathematics [1]. However, since learners can make a large range of problems, it is difficult for teachers to complete assessment and feedback for the posed problems in classrooms practically. To address this issue, technology-enhanced approaches have been used. Learning environment systems, named MONSAKUN, that practically use automatic assessment for one operation of addition and subtraction has been developed.

In the practical use and long-term evaluation, it was confirmed that learning by problem-posing with MONSAKUN was interesting and useful learning method [2]. Nevertheless, although posing problems in the learning environment is considered to contribute to the understanding of the structure of problem, it is not clear how learners could finally understand it through the activity. If we know the process and the bottleneck of thinking in problem-posing, we can consider adaptive feedback for learners. Therefore, it is essential to investigate the learning activity for every single action and to generate inferences of learner's thinking from their behavior in learning environments. This study investigates whether learner's understanding of the structure of problem reflect on their problem-posing process or not. If they have not enough understanding of the problems structure but partially correct, they make meaningful

things even in the middle of the process. To test this assumption, this study investigated the validity of the intermediate product in the problem-posing process.

## 2 Investigation of Learner's Actions

In order to promote learning deeper, MONSAKUN used as an interactive learning environment to exercise and receive lectures on problem structure as usual classes. Each learner was asked to create story problem using sentence card based on calculation expression in the requirement assignment, and all activity was logged into the database. MONSAKUN records learner's problem-posing activity as combinations of sentence cards set. The product of an activity is a combination of three sentence cards, which is called "state" and all possible combinations of sentence cards could be clearly defined as "problem state space" [3]. Each state has a meaning as the satisfaction or the lack of some constraints. Only in the case that a state does not satisfy any constraint, it is meaningless. The more constraints are satisfied in the intermediate product, the higher values of the validity are acquired.

Ideally, a learner would only need three actions to pose a correct problem, because a problem consists of the arrangement of three simple sentence cards. In this study, a Pearson's correlation test between the number of actions and the validity of posed problems is conducted. Significant difference at  $p < 0.05$  in eleven out of twelve assignments is found, which shows that many actions performed by learners tried to satisfy more constraints. This means that learners make meaningful intermediate products even in the middle of the problem-posing process.

## 3 Conclusion

An analysis of MONSAKUN log data of elementary school students in problem-posing activity to investigate their way of thinking in posing of arithmetic word problems has been conducted. We confirm that learners have a tendency to keep or enhance the meaningfulness of posed problems even in the middle of the problem-posing process.

## References

1. Silver, E.A., Cai, J.: An analysis of arithmetic problem posing by middle school students. *J. Res. Math. Educ.* **27**(5), 521–539 (1996)
2. Hirashima, T., Yokoyama, T., Okamoto, M., Takeuchi, A.: An experimental use of learning environment for problem-posing as sentence-integration in arithmetical word problems. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) *ITS 2008. LNCS*, vol. 5091, pp. 687–689. Springer, Heidelberg (2008)
3. Supianto, A.A., Hayashi, Y., Hirashima, T.: Tracing problem-posing activity sequences toward detection of trap-states in thinking. In: *23rd International Conference on Computers in Education*, pp. 85–90 (2015)

# Toward a Trace-Based PROMETHEE II Method to Answer “What Can Teachers Do?” in Online Distance Learning Applications

Hoang Nam Ho<sup>1</sup>, Mourad Rabah<sup>1</sup>, Samuel Nowakowski<sup>2</sup>,  
and Pascal Estraillier<sup>1</sup>

<sup>1</sup> L3i Laboratory, University of La Rochelle, La Rochelle, France  
{hoang\_nam.ho,mourad.rabah,  
pascal.estraillier}@univ-lr.fr

<sup>2</sup> University of Lorraine – LORIA, UMR 7503, Nancy, France  
samuel.nowakowski@loria.fr

**Abstract.** In an Online Distance Learning (ODL) application, the learning scenario is a *situation* chaining from an initial system’s state to one of the final states where course’s criteria are met. A *situation* is a contextualized sequence of interactions. Teachers are successively involved in different *situations*: at the end of one *situation*, they choose the next one. An Intelligent Tutoring System will be used to support teachers in choosing an appropriate situation. In this paper, we show how to apply a decision method (PROMETHEE II) combining with the system traces analysis to automatically recommend to teachers one situation in order to execute during an e-Learning course. Experimental results of our ODL environment are presented to illustrate our method.

**Keywords:** Intelligent Tutoring System · PROMETHEE II · Traces · Situation

## 1 Introduction

Computer science research has been shown to be effective in e-Learning context to increase the students’ performance and their motivations. With advancement of artificial intelligence and in cognitive research on human learning, the next generation of computer-based learning moved to develop an Intelligent Tutoring System (ITS). These ITSs are called cognitive tutors that must be able to achieve three main tasks: improve the student’s knowledge level, decide what to do next/adapt instruction accordingly and provide feedback [1]. Given the promising ITS’s performance, we interest in analysing how to integrate ITS into e-learning platform to help teacher and also the students during a course. We have developed a case study from our Online Distance Learning (ODL) environment POLARIS [2]. POLARIS is an online blended learning adaptive and interactive service platform. It simulates a virtual classroom with teacher and students’ roles for online courses. In this context, we propose to handle this type of application with a hypothesis for adapting its execution according to user behaviours and the current context. We rely on the notion of *situation* [2] to structure the learning scenario. A *learning situation* is a component where the teacher and students interact

using local resources associated within a common context to achieve one or more identified course’s criteria. The learning scenario in POLARIS is contextualized by the following set of situations: *Presentation, Individual Work, Group Work, Discussion, Go to the Board* and *Test*. Thus, teachers execute and participate in successive situations until they reach all of the defined criteria. For each course, the teacher also wants to reach two criteria that are: *Time* (the course must finish on time) and *Comprehension* (the students’ comprehension). Normally, after finishing one situation, teachers have to select another one among a set of available situations to carry on the learning course continuation. However, the fact is that teachers are not able to choose precisely what to do in the next execution. Our goal is to integrate an ITS as a teacher assistant to suggest an optimized situation to execute in increasing some of given criteria. In other words, we try to answer the question “What can teachers do?” and find the suggested hint being suitable for students and teacher in order to reach two defined criteria. This question will rely on the choice that will be based on the heuristic multi-criteria decision. Among the many works on multi-criteria decision, we are particularly interested in the PROMETHEE II method [3] that we improve by incorporating a system for analysing the e-learning traces [4] generated during the course session. This is the main method, named Trace-Based PROMETHEE II, which will be integrated into the ITS of POLARIS.

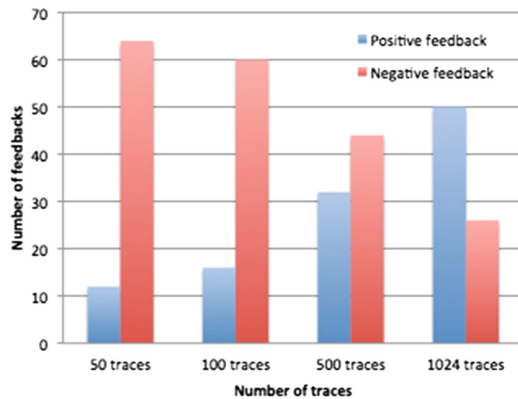


Fig. 1. Impact of number of traces on the result

Table 1. Result of feedbacks between novice and experimented teachers

	Novice teachers		Experienced teachers	
	Course with ITS	Course (not ITS)	Course with ITS	Course (not ITS)
+ Feedback	48	11	21	40
- Feedback	2	39	29	10

## 2 Principle of Trace-Based PROMETHEE II

We present the Trace-Based System (TBS) that aims to manage system execution traces. We start by introducing TBS related concepts used in this paper [4]. A *trace* is a sequence of data generated by any action regarding an event occurring during system's execution. Each trace can be associated to a model, called *trace model* that formally represents the corresponding traces. A *modelled-trace* (called *m-trace*) is a trace associated with its model. A system, which manages m-traces, is called *m-Trace-Based System* (m-TBS). The collection phase is devoted to traces collecting. All the m-traces in this phase are called primary traces, defined by  $PrT = (S, E, \Omega, C, F)$  where  $S = \{si\}$  is a set of executed situations;  $E \subseteq S \times S$  is a set of the situations transitions;  $\Omega \subseteq V \times S$  is the set of the states of the overall system  $V$  and the associated executed situation;  $C$  is the set of criteria defined by  $(criterion\_name, criterion\_value)$  and  $F$  is the teacher's feedback about the obtained criteria when he finished a course with  $F = true$  (all of criteria are reached), otherwise  $F = false$ . The transformation phase selects and transforms the primary traces into new format according to our context. We need a trace that contains a good strategy that means we select all of the primary traces in which the value of  $F$  is true. From each obtained primary trace, we have several transformed traces, denoted by  $TrT = \Omega$ . The analysis phase aims to analyse all of the transformed traces. We then present this phase as the main contribution of Trace-Based PROMETHEE II method.

The teacher has two criteria to reach during his learning session that are: *Time* and *Comprehension*. During the course in POLARIS, we have the state vector  $V$  that contains 9 properties such as: *remainT* (the remained time of the course), *presentationT* (the executed time of the situation Presentation), *indiworkT* (the executed time of the situation Individual Work), *groupworkT* (the executed time of the situation Group Work), *discussT* (the executed time of the situation Discussion), *goboardT* (the executed time of the situation Go to the Board), *testT* (the executed time of the situation Test), *questionNum* (the number of students' questions), *questionStu* (the number of students having questions). Our method will analyse after finishing one situation the state vector  $V$  to choose the most suitable situation among 6 defined situations.

Each criterion is evaluated by one or several properties in the state vector. We decompose the vector  $V$  into two sub-vectors in terms of two criteria:  $v_{time} = (remainT, presentationT, indiworkT, groupworkT, discussT, goboardT, testT)$  and  $v_{comprehension} = (questionNum, questionStu)$ . We consider the transformed traces base  $TrT$  in which each transformed trace has the following format:  $(V, executed\_situation)$ . We decompose also the base  $TrT$  into two sub-bases in terms of two criteria ( $TrT_{time}$  and  $TrT_{comprehension}$ ). For each sub-base (containing  $q$  records), we extract in  $V$  all of the properties that contribute to evaluate the correspondent criterion. For each criterion, we compute the Euclidian distance between the state sub-vector and each transformed trace. The value of  $dis_{time}^q(situation_i)$  represents the distance of the situation  $i$  between  $v_{time}$  and the  $q^{th}$  transformed trace in  $TrT_{time}$ . We get for each criterion a matrix of size  $q \times I$  containing all the distances between the current state sub-vector and  $q$  records in each transformed traces sub-base. The result is the set of two matrixes that correspond to two criteria. From the two matrixes above, we choose the  $k$  smallest distances to



compute the evaluation of each situation for each criterion, denoted  $E_h(situation_i)$ . We choose  $k$  smallest distances because it represents  $k$  similar states in the past and for each state, teachers have chosen different situation to execute. Among  $k$  choices, we compute the probability for each choice in the past. Based on these probabilities, we will obtain the evaluation value of each situation by:

$$E_h(situation_i) = \frac{e_h(situation_i)}{\sum_i e_h(situation_i)} \text{ with } e_h(situation_i) = \sum_{j=1}^k dis_h^j(situation_i), k \leq q$$

Once these calculations done for each criterion, the PROMETHEE II process continues to build a priority list of situations. The first situation of this list will be the chosen one and suggest that the teacher continue the course. The result of this method can answer our main question of the article “What can teachers do?”

### 3 Results and Discussion

Our experiments focus on two questions. Does Trace-Based PROMETHEE II suggest a situation to execute? Is this choice satisfied the teacher? We now illustrate the test to answer these questions to confirm the performance of our proposed method in POLARIS platform. We have obtained a base of 1024 transformed traces. From this base, we decompose into 2 sub-bases corresponding to 2 criteria (*Time* and *Comprehension*). We have tested our method in four cases corresponding to four traces bases as described in Fig. 1. We have four traces bases with different number of traces such as: a base containing 50 traces, 100 traces, 500 traces and 1024 traces.

In this test, we use the number of teachers’ feedback to evaluate the effectiveness for each traces base. We have totally observed 76 decision-makings using our method. We notice that the more the number of traces we have, the more the number of positive feedbacks we get. We can conclude that the combination between traces and decision method establish a new ITS. This one not only suggests that teacher do but also increases the teacher’s satisfaction.

Our method has not always obtained the high positive feedbacks from different teachers as the Table 1. We realize that the ITS integration in a course demonstrates its performance in comparison with a course without ITS for the novice teacher. We observe that the number of positive feedbacks for novice teachers is better with ITS integration than without ITS. In contrast, the positive feedback in course without ITS is higher than that is on the course with ITS for the experienced teachers. The reason is that experienced teachers are better trained in the course. Another limitation is that our method supposes the presence of a configured and up-to-date Trace-Based System. We must have enough data to compute the priority situations list to recommend to teachers. Our future work focuses on integrating the recommendation in our current ITS to improve the inconvenient about the effectiveness of experienced teachers.

## References

1. Chakraborty, S., Roy, D., Basu, A.: Development of knowledge based intelligent tutoring system. *Adv. Knowl. Based Syst.* **1**, 74–100 (2010)
2. Trillaud, F., Pham, P.T., Rabah, M., Estrailier, P., Malki, J.: Situation-Based Scenarios for E-learning. *IADIS e-learning 2012*, Lisbon, Portugal, pp. 121–128 (2012)
3. Taillandier, P., Stinckwich, S.: Using the PROMETHEE multi-criteria decision making method to define new exploration strategies for rescue robots. In: *IEEE International Workshop on Safety, Security, and Rescue Robotics*, pp. 193–202 (2011)
4. Cordier, A., Lefevre, M., Champin, P.-A., Georgeon, O.L., Mille, A.: Trace-based reasoning - modeling interaction traces for reasoning on experiences. In: *FLAIRS Conference*, pp. 363–368. AAAI Press, Florida, USA (2013)

# Enhancing Student Modeling for Collaborative Intelligent Tutoring Systems

Jennifer K. Olsen<sup>1</sup>, Vincent Alevan<sup>1</sup>, and Nikol Rummel<sup>1,2</sup>

<sup>1</sup> Human Computer Interaction Institute, Carnegie Mellon University,  
Pittsburgh, PA, USA

{jkolson, alevan}@cs.cmu.edu, nikol.rummel@rub.de

<sup>2</sup> Institute of Educational Research, Ruhr-Universität Bochum,  
Bochum, Germany

**Abstract.** This paper presents an extension of the Additive Factors Model to predict learning for students by accounting for aspects of collaboration. The results indicate that student performance is predicted more accurately when the model includes parameters that capture influences of working collaboratively.

**Keywords:** Additive factors model · Collaborative learning · Intelligent tutoring

A strength of intelligent tutoring systems (ITSs) is that they can be modified through offline student modeling to provide better instruction for students. Although ITSs have been shown to support students working in groups [2], the statistical models that are used to refine and support ITSs often do not take into account features of collaboration (e.g., partner knowledge). Student modeling might be improved and learning might be supported even better, if we took into account collaborative features. Thus, we extended the Additive Factors Model (AFM), which is a logistic regression model frequently used in offline analyses of ITSs [1]. The standard AFM [1] calculates the log-odds that a given student correctly solves a given step in a problem as a function of three estimated parameters that capture the student's initial proficiency, the ease of the skills involved in the step, and the learning rates for those skills. We modified the standard AFM so that the model has separate learning rates depending on if a skill is being learned in an individual or collaborative environment (AFM+C) since the learning processes may differ. Further, to better understand how a student's partner's knowledge may impact the prediction of a student's learning, we analyzed four different variations to take into account partner knowledge: partner pretest score classified as low/average/high (AFM+PPS), absolute difference between student's and partner's pretest scores classified as homogeneous or heterogeneous (AFM+AD), and two directional differences between student's and partner's pretest scores including lower/similar/higher (AFM+DD) and lowest/lower/similar/higher/highest (AFM+LD).

We hypothesized that the models with collaborative/individual learning rates and the models with partner knowledge would be a better fit than the standard AFM. We used

two datasets consisting of log data from conceptually or procedurally-oriented ITSs. In each data set, students were working individually or collaboratively. We measured the accuracy with which the models predicted student performance for both datasets using log likelihood, Akaike information criterion (AIC), and Bayesian information criterion (BIC). The log likelihood does not take into account the complexity of the model while the AIC and BIC do account for the complexity of the model.

The models with collaborative/individual learning rates were a better fit than the standard AFMs as shown in the AFM+C rows of Table 1. The models with variations of the partner’s pretest were a better fit only for the conceptually-oriented data as shown in the comparison column in Table 1. This may be caused by different types of talk occurring around conceptual and procedural knowledge with a partner, which may have an influence on learning. Overall, our results show that by including collaborative features within a model, we can improve the learning prediction. With a more accurate learning prediction for an ITS, in future work, we may be able to better refine the instructional support used in individual and collaborative ITSs.

**Table 1.** Prediction accuracy for all models. The asterisks mark the best performing model while a plus sign indicates that the model performed worse than the baseline for that measure.

Model name	Log likelihood	AIC	BIC	Comparison to standard AFM
Conceptually-oriented ITS				
Standard AFM	-8769.7	17549.3	17589.1	
AFM+C	-8731.2*	17478.3*	17542.0*	$\chi^2(3) = 77.0, p < 0.001$
AFM+PPS	-8759.3	17534.6	17598.2+	$\chi^2(3) = 20.8, p < 0.001$
AFM+AD	-8768.6	17553.3+	17616.9+	$\chi^2(3) = 2.1, p = 0.56$
AFM+DD	-8761.2	17538.4	17602.1+	$\chi^2(3) = 16.9, p < 0.001$
AFM+LD	-8760.5	17536.9	17600.6+	$\chi^2(3) = 18.4, p < 0.001$
Procedurally-oriented ITS				
Standard AFM	-7991.3	15992.6	16032.0	
AFM+C	-7942.8*	15901.5*	15964.5*	$\chi^2(3) = 97.1, p < 0.001$
AFM+PPS	-7989.0	15994.0+	16057.1+	$\chi^2(3) = 4.6, p = 0.20$
AFM+AD	-7989.2	15994.4+	16057.4+	$\chi^2(3) = 4.2, p = 0.24$
AFM+DD	-7988.7	15993.5+	16056.5+	$\chi^2(3) = 5.2, p = 0.16$
AFM+LD	-7987.9	15991.9	16054.9+	$\chi^2(3) = 6.8, p = 0.08$

**Acknowledgments.** We thank the CTAT team, Ran Liu, and Amos Glenn for their help. This work was supported by Graduate Training Grant # R305B090023 and by Award # R305A120734 both from the US Department of Education (IES).

## References

1. Cen, H., Koedinger, K.R., Junker, B.: Is over practice necessary?-improving learning efficiency with the cognitive tutor through educational data mining. In: Proceedings of the 13th International Conference on Artificial Intelligence in Education, pp. 511–518. IOS Press (2007)
2. Walker, E., Rummel, N., Koedinger, K.: Adaptive intelligent support to improve peer tutoring in algebra. *Int. J. Artif. Intell. Educ.* **24**(1), 33–61 (2014). doi:[10.1007/s40593-013-0001-9](https://doi.org/10.1007/s40593-013-0001-9)

# Toward Embodied Game-Based Intelligent Tutoring Systems

Ivon Arroyo, Yuting Liu, Naomi Wixon, and Sarah Schultz

Worcester Polytechnic Institute, Worcester, UK  
{iarroyo, yliu3, mwixon, sechultz}@wpi.edu

**Abstract.** We present a new architecture for the creation of embodied educational games, using wearable devices in the form of ‘Wearables’ for learning, which enable to do mathematics while being physically engaged with the environment. Wearables act as assistants as students engage in Mathematics Games. Evidence shows that students learn and improve their affect and motivation.

**Keywords:** Embodied cognition · Game-based learning environments

## 1 Introduction

Ideally, digital learning environments should manage the delicate balance between motivation and cognition, promoting both interest and deep learning. Embodied Cognition might be able to bridge these two worlds: physical embodied learning supports the combination of movement and gestures with higher-order cognitive activities [1–3], and manipulating objects can be engaging and interesting. At the same time, technologies are becoming more hands-on, tangible, sharable and even wearable. There is potential for learning technologies to redefine the way that teachers and students interact, through rich interactive face-to-face discourse that blends technology into classroom culture. The vision is that students manipulate shapes/expressions with their hands, measure using real tools, do math in their owned physical spaces, as we essentially move the interface “off the screen” and into the real world, but retain the benefits of personalization, hints and feedback that ITS already provide.

## 2 Wearables for Mathematics Learning

The Cyberlearning Watch is a device that students use on their wrist to receive instructions while playing math games. We manufactured these wearables using the Arduino Uno Microcontroller. As a second option, a device emulator was developed in JavaScript, to have the option of using cell phones strapped to forearms via armband. Wearables connect to a remote web-server engine via Wify and web sockets.

The back-end consists of a Ruby on Rails web-server. The software of this system (1) communicates with devices; (2) keeps state of the game; (3) aids the teacher in the general functioning of activities (start the game, verify progress, determine the winner);

(4) allows to assign watch devices to students; (5) specifies teams of students; (6) keeps track of individual and team progress; (7) allows the creation of new games.

**The Tangrams Race Game.** This is a team-based game for 3<sup>rd</sup>, 4<sup>th</sup> grade, consisting of a relay race where students take turns to run and find specific shapes out of a basket at the other end of the room, to collect the correct pieces to solve a Tangram Puzzle. Students get descriptions of the shapes they have to bring back, such as *“I have 3 angles, but only one of them is a right angle”*, and can push the “hint” button to get help, e.g. *“3 angles, 3 sides and 1 right angle make a right triangle”*. Students (N = 96) playing this game with wearables demonstrated large improvements in affective variables in pre-posttests (e.g. +30 % in ‘math is very interesting’), and improved achievement in standardized test items after a 30-min intervention.

**Estimate IT!** is a measurement estimation and number sense game for 4<sup>th</sup>–6<sup>th</sup> grade students. The game is a Scavenger Hunt, where students search for objects around a physical space. For example, the display could show the following message to a player: *“Find a cube with a 6 side”*. Hint 1: *“I have 6 equal faces”*. In this game there is synchronization of Wearables for members of the same team. Students (N = 13) played Estimate IT with improvement in achievement at standardized test items, and improved in mathematics liking and self-concept in math after 30 min.

### 3 Discussion

Adapting the difficulty of questions for each individual student, based on a student model, is the obvious next step of this research, following approaches traditional to ITS [4], as meeting the right level of challenge has been frequent feedback throughout studies: some students found the games too hard, others too easy. This would imply that a variety of questions and hints of different difficulties be generated for the same object to search, fitting a variety of grades. The result will be that everybody, regardless of achievement level, will have a similar chance to succeed and win the games.

We are encouraging about this platform for formal and informal education, given large improvements in student affect, motivation and achievement. Next, we will implement the architecture using SmartWatches, which are quickly becoming wide spread. This is a new paradigm for educational technology where the ITS model still holds, but taking the interface “off the screen”. Many research questions remain unanswered, such as what are the mechanisms by which embodied cognition contribute to improved learning, engagement and teaching, and how technology can support it. In many ways, we have created a platform to start answering these research questions.

### References

1. Alibali, M., Nathan, M.: Embodiment in mathematics teaching and learning: evidence from learners’ and teachers’ gestures. *J. Learn. Sci.* **21**(2), 247–286 (2012)
2. Nemirovsky, R., Tierney, C., Wright, T.: Body motion and graphing. *Cogn. Instr.* **16**(2), 119–172 (1998)

3. Abrahamson, D.: Embodied design: constructing means for constructing meaning. *Educ.Stud. Math.* **70**(1), 27–47 (2009)
4. Arroyo, I., Woolf, B.P., Burelson, W., Muldner, K., Rai, D., Tai, M.: A multimedia adaptive tutoring system for mathematics that addresses cognition, metacognition and affect. *Int. J. Artif. Intell. Educ.* **24**(4), 387–426 (2014)



# Towards Computer-Assisted Curricula Design Using Probabilistic Graphical Models

Waleed Alsanie<sup>1</sup>, Issa Alkurtass<sup>1</sup>, and Abdullah Al-Hamoud<sup>2</sup>

<sup>1</sup> National Center of Computation Technology and Applied Mathematics,  
King Abdulaziz City for Science and Technology, Riyadh, Saudi Arabia  
{walsanie, ialkurtass}@kacst.edu.sa

<sup>2</sup> College of Computer and Information Sciences,  
Al-Imam Mohammed Ibn Saud Islamic University, Riyadh, Saudi Arabia  
z24abdullah@gmail.com

**Abstract.** A conventional way of designing curricula is to ask domain experts to set dependencies between courses. A domain expert normally sets these dependencies based on the contents of the courses. Regardless of the disagreements amongst experts in deciding these dependencies, several studies suggest that data-driven approaches can be very effective. In this paper, we propose a system to induce independence/dependence assumptions between courses in curricula based on student performance.

**Keywords:** Curricula design · Educational tools · Dependency graph · Intelligent tutoring systems · Machine learning · Probabilistic graphical models

## 1 Introduction

Designing educational curricula is a crucial and an important task in building any educational plan. An important issue in designing curricula is deciding how courses depend on each others. A conventional way of deciding these dependencies is based on the contents of the courses. Although, this approach is justified by common sense, there can also be dependencies related to factors other than the manifest contents of the courses. An empirical study by Vuong *et al.* show that some prerequisites set by domain experts are not true prerequisites with respect to the performance of the students they analysed. They argue that an empirical study of student performance can be of significant assistance to educators in designing curricula [2]. Ohland *et al.* reported that after redesigning an engineering curriculum such that putting a mathematics course, which had long been thought as a necessary prerequisite to an engineering course, as a co-requisite to it increased the graduation rate [4].

Historical data collected from the performance of students can be of great benefit in revising the dependencies between courses. A dependency graph can be learned from this data. This graph is easier to read and interpret than numbers and tables. It gives educators a picture of direct and indirect associations between courses.

In this work, we propose a system to build a dependency graph of courses. This dependency graph is a *Bayesian network* (BN) learned from data of student grades.

## 2 Experiments

Unfortunately, at the time of writing this paper, no real data was at our hand with which we can experiment. We picked three subsets of three curricula from three departments in a college. These departments were physics (PHYS), mathematics (MATH) and computer science (CS). We built three BNs representing independence assumptions based on our belief on the performance of the students taking these courses. We set each random variable representing the grade of a student to take a value in a sample space  $\Omega = \{poor, intermediate, high\}$ . From each BN, we generated two datasets, one containing 1000 instances and the other containing 10000 instances. We used the GOBNILP<sup>1</sup> software package to learn BNs from these instances [1]. In order to compare the learned BNs with the original ones, which serve as gold standards, we used the *structural Hamming distance* (SHD) metric [3]. SHD measures the number of edge operations: deletion, insertion and reverse, which need to be done to one of the graphs to transform it to the other. When two BNs encode the same independence assumptions, the SHD between them is 0. Table 1 shows the result of our experiments. It is clear that, except for the BN learned for the physics curriculum from 10000 instances, all the independence assumptions encoded by the BNs could be recovered from the data.

**Table 1.** The results of learning BNs from synthetic data for three college departments.

Curriculum	SHD (1000)	SHD (10000)
Mathematics	0	0
Physics	0	1
Computer Science	0	0

**Acknowledgments.** We thank Thana Alharbi for preparing data for preliminary experiments.

## References

1. Cussens, J.: Bayesian network learning with cutting planes. In: Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI-11), pp. 153–160 (2011)

<sup>1</sup> <https://www.cs.york.ac.uk/aig/sw/gobnilp/>.

2. Vuong, A., Nixon, T., Towle, B.: A method for finding prerequisites within a curriculum. *J. Educ. Data Min.* (2011)
3. Tsamardinos, I., Brown, L., Aliferis, F.: The max-min hill-climbing Bayesian network structure learning algorithm. *Mach. Learn.* **65**(1), 31–78 (2006)
4. Ohland, W., Yuhasz, G., Sill, L.: Identifying and removing a calculus prerequisite as a bottleneck in clemson’s general engineering curriculum. *J. Eng. Educ.* **93**, 253–257 (2004)

# Predicting Spontaneous Facial Expressions from EEG

Mohamed S. Benlamine<sup>1</sup>, Maher Chaouachi<sup>1</sup>, Claude Frasson<sup>1</sup>,  
and Aude Dufresne<sup>2</sup>

<sup>1</sup> Computer Science and Operations Research, University of Montreal, Montreal,  
QC, Canada

{ms.benlamine, chaouachi}@umontreal.ca,  
Frasson@iro.umontreal.ca

<sup>2</sup> Communication Department, University of Montreal, Montreal, QC, Canada  
dufresne@umontreal.ca

**Abstract.** The current study focuses on building a real-time emotions model to automatically detect emotions directly from brain signals. This model analyses the learner's emotional state which is very useful to intelligent tutoring systems. An experiment was conducted to record neural activity and facial micro-expressions of participants while they are looking at pictures stimuli from the IAPS (International Affective Picture System). Camera-based facial expression detection software (FACET) was used to assess facial micro-expressions of a participant with high accuracy. Machine learning algorithm was fed with time-domain and frequency-domain features of one second EEG signals with the corresponding facial expression data as ground truth in the training phase. The classifier provides outputs representing facial emotional reactions dynamics in fully automatic and non-intrusive way without the need to a camera. Using our approach, we have reached 92 % accuracy.

**Keywords:** Affective model · Facial expressions · EEG · Brain computer interface

## 1 Introduction

Emotions are crucial in the learning process because they have profound impact on what we learn and keep in mind. In Intelligent Tutoring systems (ITS) an important goal remains to monitor learners' emotions on a real-time basis. Several studies have been successfully conducted to detect emotions using models that track facial micro-expressions with camera or webcam with CERT [1], or FaceReader [2] etc. A micro-expression [3] is a brief spontaneous facial expression, unconscious (involuntary) and hard to hide as it lasts between 1/24 and 1/15 of a second. However, so far, all the focus has been on external assessment methods and to the best of our knowledge, no attempt has ever been made at detecting facial micro-expressions from EEG signal. In the meanwhile, several works has shown that emotional states can be recognized from EEG signal with reasonable accuracy [4–6]. So it seems sensible then to consider cerebral activity as input for detecting facial expressions rather than user's

face images or videos. In this paper, we aim to answer this question: How well can we predict facial expressions from the brain signals of participants?

## 2 Experiment and Results

In order to elicit the participants' emotions, we conducted an experiment with 20 participants (7 women, 13 men aged from 21 to 35) where they was asked to look at selected 30 pictures from IAPS (International Affective Picture System) [7] with regard to their affective ratings (valence, arousal) and choose one from the eight emotion categories that best represent his global feeling about the displayed pictures.

To build our EEG based affective model, we proceeded according to the following steps: (1) measuring facial expressions of users confronted to IAPS pictures using FACET system, (2) Extracting frequency-domain and time-domain features from the corresponding *mobile windows* of 1 s EEG data for each FACET frame time (every 1/6 s) and train machine learning models that correlates the micro-expressions probabilities with the EEG features, and (3) predicting the emotion only from the model and check the accuracy of the model.

The total number of computed features from all the electrodes is 238 (17 features per electrode: 5 frequency-domain features + 12 time-domain features). The collected dataset contains 21553 data points (1078 data point per participant; 36 data point per stimulus). We have created 10 CSV files where we put together all the extracted EEG Features and one emotion category extracted from FACET data. Each file was entered as an input to the WEKA machine learning toolkit for generating EEG-based regression model to predict the values of one emotion category.

Three machine learning algorithms were used to predict the numeric values of each emotion category; namely IBk (K-nearest neighbours classifier), Random Forest (classifier constructing a forest of random trees) and RepTree (Fast decision tree learner). We used 10 fold validation in our test phase.

Compared to IBk ( $k = 1$  neighbour) and RepTree methods, Random Forest obtains higher correlation coefficient and lower error rates such as MAE and RMSE for all emotion categories, as illustrated in Table 2.

With these results, our EEG-based facial expressions prediction approach provides a simple and reliable way to capture the emotional reactions of the user that can be used in intelligent tutorial systems, games, neurofeedback, and VR environments.

**Table 1.** Computed features from EEG Signals

Frequency-domain EEG Features (5 Features)	delta [1–4 Hz], theta [4–8 Hz], alpha [8–12 Hz], beta [12–25 Hz], and gamma [25–40 Hz]
Time-domain EEG Features (12 Features)	Mean, Standard Error, Median, Mode, Standard Deviation, Sample Variance, Kurtosis, Skewness, Range, Minimum, Maximum and Sum

**Table 2.** Comparison between machine learning methods

Emotion	IBk			Random forest			RepTree		
	CC	MAE	RMSE	CC	MAE	RMSE	CC	MAE	RMSE
Joy	0.8528	0.0174	0.0545	0.9076	0.0225	0.0483	0.7174	0.0298	0.0702
Anger	0.8794	0.0518	0.0907	<b>0.9216</b>	0.0534	0.0753	0.8282	0.0684	0.1041
Surprise	0.854	0.0163	0.0395	0.8965	0.0175	0.0346	0.757	0.024	0.0484
Fear	0.8891	0.0431	0.075	0.9164	0.0477	0.0686	0.7754	0.065	0.1015
Neutral	0.8741	0.0537	0.1	0.9074	0.0639	0.0927	0.7428	0.0849	0.1352
Contempt	0.79	0.0341	0.0664	0.8526	0.0351	0.0575	0.6307	0.0488	0.0807
Disgust	0.8792	0.0316	0.0585	0.916	0.0327	0.0504	0.6307	0.0488	0.0807
Sadness	0.8892	0.0311	0.0628	<b>0.9203</b>	0.0341	0.0552	0.827	0.0443	0.0759
Positive	0.8528	0.0518	0.105	0.9124	0.0582	0.0869	0.7591	0.0765	0.1266
Negative	0.8569	0.0998	0.1608	0.9034	0.1052	0.1362	0.797	0.1264	0.1834

### 3 Conclusion

In this paper, we presented models for detecting user’s facial micro-expressions in front of IAPS pictures using signals from his cerebral activity. We showed our methodology and experimental design to get a reliable dataset that we used in the training of machine learning algorithms. The obtained results are very promising and we aim to investigate them in the future in an intelligent tutorial system. This integration will contribute to learning because practical real-time non-invasive assessment of users’ emotion is now feasible, since we can rely on this assessment as a substitute for self-reports that can disturb a learning/gaming session. Moreover, the system can become more adaptive in terms of its response to user’s emotional reactions. With more precise emotions measures that our system offers, intelligent tutorial systems will be more adaptive that react to the user’s spontaneous facial expressions in real-time.

**Acknowledgements.** The presented research is supported by funding awarded by the Natural Sciences and Engineering Research Council of Canada (NSERC).

### References

1. Littlewort, G., et al.: The computer expression recognition toolbox (CERT). In: 2011 IEEE International Conference on Automatic Face and Gesture Recognition and Workshops. FG 2011. IEEE (2011)
2. Lewinski, P., den Uyl, T.M., Butler, C.: Automated facial coding: validation of basic emotions and FACS AUs in FaceReader. *J. Neurosci. Psychol. Econ.* **7**(4), 227 (2014)
3. Ekman, P.: *Emotions Revealed: Recognizing Faces and Feelings to Improve Communication and Emotional Life.* Macmillan (2007)
4. Chaouachi, M., Frasson, C.: Mental workload, engagement and emotions: an exploratory study for intelligent tutoring systems. In: Cerri, S.A., Clancey, W.J., Papadourakis, G., Panourgia, K. (eds.) ITS 2012. LNCS, vol. 7315, pp. 65–71. Springer, Heidelberg (2012)

5. Heraz, A., Frasson, C.: Predicting the three major dimensions of the learner's emotions from brainwaves. *World Acad. Sci. Eng. Technol.* **31**, 323–329 (2007)
6. Liu, Y., Sourina, O., Nguyen, M.K.: Real-time EEG-based emotion recognition and its applications. In: *Transactions on computational science XII*, pp. 256–277. Springer (2011)
7. Lang, P.J., Bradley, M.M., Cuthbert, B.N.: International affective picture system (IAPS): affective ratings of pictures and instruction manual. Technical report A-8 (2008)

# An Empirical Evaluation of Learning Style and Knowledge Level Adaptation

Mohammad Alshammari<sup>1</sup>, Rachid Anane<sup>2</sup>, and Robert J. hendley<sup>1</sup>

<sup>1</sup> School of Computer Science, University of Birmingham, Birmingham, UK  
{m. t. m. alshammari, r. j. hendley}@cs. bham. ac. uk

<sup>2</sup> Faculty of Engineering and Computing, Coventry University, Coventry, UK  
r. anane@coventry. ac. uk

**Abstract.** This paper presents an initial evaluation of different forms of adaptation based on learning style and knowledge level, which were implemented in an adaptive e-learning system. An experiment conducted in a learning context with 174 participants produced significant results in terms of learning gain. They indicate that adaptation based on both learning style and knowledge level yields significantly better learning gain than adaptation based on learning style only, and better than adaptation based on knowledge level only.

**Keywords:** Adaptivity · Learning style · Learner knowledge · Evaluation

## 1 Introduction

One of the key characteristics of adaptive e-learning systems (AESs) is the provision of personalized services and the recommendation of learning material in the learning process. Although most AESs are based on learning style (LS) or on knowledge level (K) adaptation, the implementation of the adaptive process based on their combination presents a significant challenge. The lack of empirical research on LS and K adaptation is also a key issue in the deployment of AESs [1].

This paper is part of a broader investigation into adaptation based on LS and K in e-learning systems, and their empirical evaluation. An AES called AdaptLearn was designed and implemented to provide adaptation based on LS only, K only or on their combination [2, 3]. Adaptation is achieved through the manipulation of links to learning material on Computer Security, the application domain of the system. This involves the inclusion and generation of links to relevant material in a customized order, and the hiding or removal of links deemed unsuitable for the current level of the learner. It also provides adaptive guidance and offers recommendations and feedback to learners as they progress through their learning tasks.

An experimental evaluation of the impact of adaptation was carried out in terms of learning gain. This involved 174 participants in an academic learning environment. A pre-test and a post-test were used to measure the learning gain; each test contains 22 multiple-choice questions with five options. Learning gain was computed as follows:

Learning Gain = the score of the post-test – the score of the pre-test



Three independent variables/groups were established in the experiment. Group 1 involved the participants who interacted with a version of AdaptLearn which provides adaptation based on LS only. Group 2 involved the participants who interacted with a version which provides adaptation based on K only. Group 3 involved the participants who interacted with a version which provides adaptation based on the combination of LS and K. The experiment involved a number of stages: (1) access to AdaptLearn through an Internet browser and completion of the demographic data form and of the Index of Learning Style questionnaire [4]; (2) random assignment by the system of each participant to one of the experimental groups; (3) completion of a pre-test by each participant; (4) personalized study by the groups of the learning material on Computer Security; (5) completion of a post-test by each participant.

Table 1 presents the results of the learning gain variable for the experimental groups; they indicate that Group 3 had the highest mean value followed by Group 2 and then Group 1. According to the results, adaptation based on both LS and K in AdaptLearn yields significantly better learning gain than adaptation based on K only and better than LS only.

**Table 1.** One-way ANOVA results of learning gain relating to the experimental groups.

	N	Mean	SD	F(2,171)	Sig.
Group 1	58	53.50	18.92	22.89	<0.0005
Group 2	58	64.74	18.94		
Group 3	58	75.48	14.18		

This experiment contributes to current research on adaptation by providing more evidence on learning gain when both the LS and K characteristics are integrated into an AES. Future research will involve a long-term evaluation with more participants and a larger set of learning resources. Different variables such as learner satisfaction, motivation, emotion and perceived usability of the system can be taken into account in future experiments.

## References

1. Akbulut, Y., Cardak, C.S.: Adaptive educational hypermedia accommodating learning styles: a content analysis of publications from 2000 to 2011. *Comput. Educ.* **58**, 835–842 (2012)
2. Alshammari, M., Anane, R., Hendley, R.J.: Design and usability evaluation of adaptive e-learning systems based on learner knowledge and learning style. In: Abascal, J., Barbosa, S., Fetter, M., Gross, T., Palanque, P., Winckler, M. (eds.) *INTERACT 2015*. LNCS, vol. 9297, pp. 584–591. Springer, Heidelberg (2015)
3. Alshammari, M., Anane, R., Hendley, R.: An e-learning investigation into learning style adaptivity. In: *The 48th Hawaii International Conference on System Sciences (HICSS-48)*. pp. 11–20 (2015)
4. Felder, R.M., Spurlin, J.: Applications, reliability and validity of the index of learning styles. *Int. J. Eng. Educ.* **21**, 103–112 (2005)

# Text-Based Emotion Recognition Approach

Mohammed Abdel Razek<sup>1</sup> and Claude Frasson<sup>2</sup>

<sup>1</sup> Research and Development Department, Deanship of Distance Learning,  
King Abdulaziz University, Jeddah P.O. Box 80254 Kingdom of Saudi Arabia

<sup>2</sup> Département d'informatique et de recherche opérationnelle,  
Université de Montréal, C.P. 6128, Succ. Centre-ville Montréal, Montréal  
QC H3C 3J7 Canada

**Abstract.** Recognize learner's emotion during a learning session is very important task for intelligent tutoring system and help in improving learners' interactions and understanding their preferences. This paper proposes a text-based emotion recognition approach that uses personal text data to recognize learner's current emotion. The proposed approach employs Dominant Meaning Technique to recognize learner's emotion.

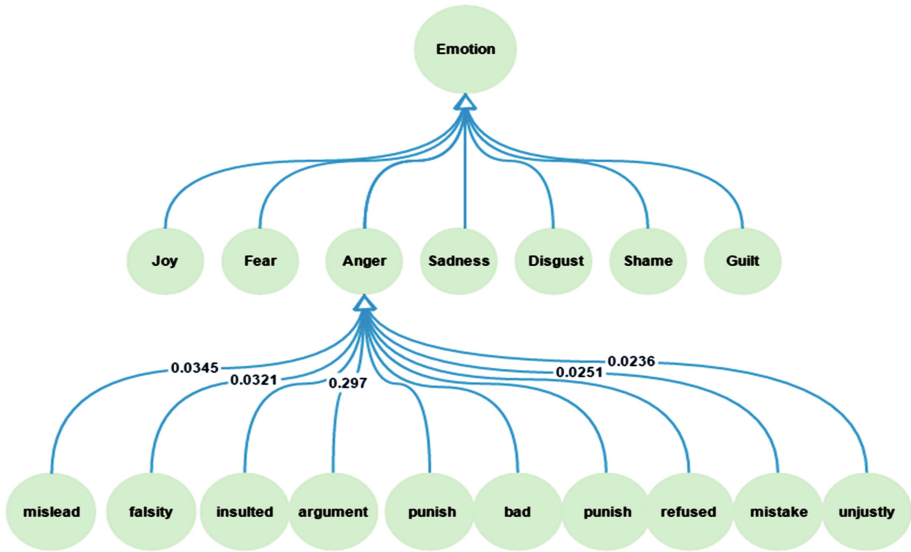
## 1 Introduction

In Collaborative learning session between learners, emotions are an important aspect. The detection of the exchange of emotions among learner through text messages can help Intelligent Tutoring System (ITS) for deliver right concept to right learners. Jraidi et al. [1] mentioned that the impact of using emotion in ITS and how that are oriented toward developing emotionally sensitive tutors.

This paper presents a new technique based on Dominant Meaning Technique [2] and Appraisal Method [3] to classify a text to a suitable emotion. Detecting emotion from text is useful in understanding learners' feelings towards particular discussion in Intelligent Tutoring System. To test our algorithm, we use ISEAR (International Survey on Emotion Antecedents and Reactions), dataset collected by Klaus R. Scherer and Harald Wallbott [4]. ISEAR dataset contains 7 major emotions: joy, fear, anger, sadness, disgust, shame, and guilt. Attitude defined as a mode that anyone act in a specific condition and shows how he feels. These aspects embody the capability to express emotional, moral, and aesthetic feelings respectively [4]. For example, "when I was in grad 11 in the school, I was punished for no serious mistake of mine" another sentence "when I was in grad 11 in the school, I got an award for my excellence". Using the dominant meaning methods, the words "punish, and mistake" leads the first sentences to a negative emotion, however, the words "award and excellence" classify the second sentences to a positive emotion.

## 2 Building Emotion Dominant Meaning Tree

Most of text classification methods uses keyword-based methods with thesaurus. In contrast, we use the dominant meaning methods as features to improve accuracy and refine the categories. Each node contains one emotion. Each emotion is associated with



**Fig. 1.** The dominant meaning tree of emotion categories.

top-N dominant meaning words based. The node between word and the emotion is labeled with its dominant meaning probability as shown in Fig. 1. To determine N value, we have to conduct some experimentations with different N values to figure out which N reflects a considerable results. The proposed framework showed how to form the dominant meaning tree which we used to classify sentences to emotions classes.

## References

1. Jraidi, I., Chalfoun, P., Frasson, C.: Implicit strategies for intelligent tutoring systems. In: Cerri, S.A., Clancey, W.J., Papadourakis, G., Panourgia, K. (eds.) ITS 2012. LNCS, vol. 7315, pp. 1–10. Springer, Heidelberg (2012)
2. Abdel Razeq, M.: Dominant meaning method for intelligent topic-based information agent towards more flexible MOOCs. *J. Intell. Learn. Syst. Appl.* **6**(04), 186–186 (2014)
3. Wu, B.: Appraisal perspective on attitudinal analysis of public service advertising discourse. *Engl. Lang. Lit.* **3**(1) (2013)
4. Scherer, K.R., Wallbott H.G.: Evidence for universality and cultural variation of differential emotion response patterning. *J. Pers. Soc. Psychol.* **66**, 310–328 (1994)

# The Questions of Ethics in Learning Analytics

Madeth May and Sébastien Iksal

UBL University of Maine – EA4023, Avenue Olivier Messiaen, Le Mans, France  
{madeth.may, sebastien.iksal}@univ-lemans.fr

**Abstract.** Learning Analytics (LA) plays a crucial role in providing e-learning practitioners both technological and methodological assistance in terms of learner monitoring and evaluation. Most research efforts in LA strongly place emphasis on analytic processes and technologies that support the tasks of data analysis and visualization. Creating cutting-edge and efficient techniques for the latter appears to be a prioritized research challenge for the LA research community while ethical questions are often neglected. Yet, these questions have been proven with research evidence to be compelling for the researchers and at the same time preoccupied for the learners. The research effort presented in this paper takes a closer look at a particular issue in LA related to user tracking and how ethical issues are being handled.

**Keywords:** Data analysis · Data indicator · Ethics in e-learning · Learning analytics · Privacy issues in e-learning · Tracking data

## 1 Introduction

In e-learning, ethics involves many aspects, among which, are the participants themselves and the technologies used to support their learning activities [1, 2]. As for the ethical questions, they need to be studied in a more situated context by taking into account a variety of factors, including specific institution's regulations, learning environments and cultural points of view. For that reason, the research effort presented in this paper intentionally places a special focus on user-tracking in Learning Analytics and its related ethical and privacy concerns. It is based on the empirical data acquired through a questionnaire distributed to 178 students. Our research goal is to study how LA causes privacy concerns, as the learners perceived them, and how it would affect their behavior in e-learning.

## 2 Privacy Concerns in Learning Analytics

Student monitoring, the most common practice in LA, has become easier and more efficient thanks to the use of tracking approach [3] that consists of collecting data of users and of their interactions within learning platforms. Acknowledging the contributing factor of user-tracking data to online tutoring enhancements as pointed out by [4], researchers and learning content providers choose to integrate systematically a tracking system in their educational settings. LA applications and tools that support

tracking data analysis and visualization are also widely introduced to the participants in the learning process. To back up this claim, the most recent studies of [5] review a variety of LA platforms that make use of students' tracking data for different purposes, among which are student assessment and evaluation. While LA gives considerable assistance to the tutors in the tasks of monitoring online learning, it also creates major drawbacks for the learners. As a matter of fact, 68 % of the participants in our study expressed their fears towards a learning environment with an integrated tracking system. The participants recognized that the latter had sometimes affected how they perform certain types of activities. For example, they suggest limiting private activity or to reduce public intervention like on a discussion forum, so that they would leave the least of their traces possible on an open E-learning environment.

The data from the questionnaire also reveal that "avoidance of personal data" and "consent agreement" are both strongly relevant to privacy concerns in E-learning. In fact, consent is one of the keystones of privacy research practices in LA. Somehow, we were surprised to learn that most of the participants had never been reached out by anyone to sign a consent form. Yet, they have been regularly using e-learning platforms (e.g. Moodle), and their tracking data have been exploited in both educational and research settings. The ethical question in this case has either been ignored or intentionally overlooked by the learning content provider.

### 3 Conclusion

While the main goal of this paper is to share scientific findings based on field studies and empirical data, it also aims to raise awareness of the privacy matters, which are often overlooked in the research efforts involved in Learning Analytics. Our future work will focus on a more in-depth analysis of the current experimental data to explore other aspects related to ethics in e-learning. We are also attempting to quantify and qualify the impact of the privacy issues on the behavioral, social and cognitive aspects of online learning.

### References

1. Toprak, E., Ozkanal, B., Aydin, S., Kaya, S., Toprak, E., Ozkanal, B., Aydin, S., Kaya, S.: Ethics in e-learning. *Turk. Online J. Educ. Tech.* **9**, 78–86 (2010)
2. Brown, T.: Ethics in eLearning. In: *Workshop for Net Business Ethics*, pp. 1–6 (2008)
3. May, M., George, S., Prévôt, P.: TrAVis to enhance online tutoring and learning activities: real time visualization of students tracking data. *Int. J. Interact. Technol. Smart Educ.* **8**, 52–69 (2011)
4. Jermann, P., Soller, A., Muehlenbrock, M.: From mirroring to guiding: a review of state of the art technology for supporting collaborative learning. In: *European Conference on Computer-Supported Collaborative Learning*, pp. 324–331. The Netherlands (2001)
5. Alowayr, A., Badii, A.: Review of monitoring tools for e-learning platforms. *Int. J. Comput. Sci. Inf. Technol.* **6**, 79–86 (2014)

# An Analysis of Feature Selection and Reward Function for Model-Based Reinforcement Learning

Shitian Shen, Chen Lin, Behrooz Mostafavi, Tiffany Barnes, and Min Chi

North Carolina State University, Raleigh, NC 27695, USA  
{sshenn,clin12,bzmostaf,tmbarnes,mchi}@ncsu.edu

## 1 Introduction

In this paper, we propose a series of correlation-based feature selection methods for dealing with high dimensionality in feature-rich environments for model-based Reinforcement Learning (RL). Real world RL tasks usually involve high-dimensional feature spaces where standard RL methods often perform badly. Our proposed approach adopts correlation among state features as a selection criterion. The effectiveness of the proposed methods are compared against previous methods referred as 10PreviousFS [2] using the data from an intelligent logic tutor called Deep Thought (DT) [1]. We evaluated the effectiveness of different feature selection methods by expected cumulative reward (ECR) [3], considering two types of reward: immediate and delayed. Our results show that our proposed methods significantly outperform previous feature selection methods with both types of rewards. Moreover, the “best” policy induced using immediate reward differs significantly from that induced from delayed reward.

## 2 Methodology, Experiments, and Results

**Methodology:** The proposed feature selection framework forwardly select the feature based on the correlation between the feature and current selected feature set. Particularly we applied three common used correlation metrics Chi-square (CHI), Information gain (IG) and Symmetrical uncertainty (SU) for measuring the correlation among features. When considering the three correlation metrics, we face a simple decision: should we select the next feature that is the most **correlated** or **uncorrelated** to the current selected features? However, for model-based RL, the answer is not straightforward. We used both high and low correlation on three correlation metrics thus resulted in six methods: CHI-high, IG-high, SU-high, CHI-low, IG-low, and SU-low. We will compare these six methods with 10PreviousFS in [2].

**Experiments:** We applied RL [3] to induce policy given the action set, rewards and state features. Here we focus on one simply tutorial decision: once a tutor determines the problem to be completed, should the tutor show the student how to solve the next problem directly—worked example (WE), or should it ask the

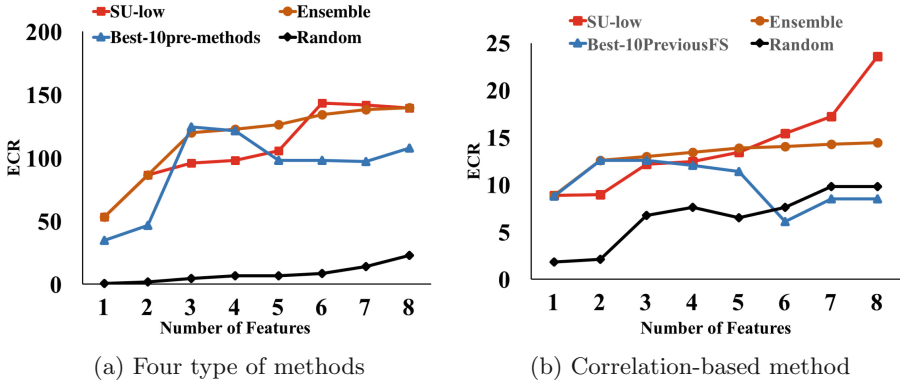


Fig. 1. Feature selection methods for immediate reward.

student to solve the problem–problem solving (PS)? In addition, we designed two types of rewards: Immediate vs. Delayed. The former reflects students’ performance level by level while The latter determines the students’ performance across all levels. Besides, 134 state features were extracted representing a cumulative statistical measure of students’ behavior and context information.

**Result:** Among the three correlation-based methods, our results showed that SU-based methods outperforms CHI and IG based ones. Moreover, Fig. 1a, b show that (1) SU-based > ensemble > the best of 10PreviousFS > random; and (2) the ECR of Immediate policies much higher than that of Delayed policies. This is most likely because of the issue of credit assignment. The more we delay success measures from a series of sequential decisions, the more difficult it becomes to identify which of the decision(s) in the sequence are responsible for our final success or failure.

### 3 Conclusion and Future Work

In this paper, we proposed six correlation-based feature selection methods for model-based RL and showed that they are more effective than the ensemble method and 10PreviousFS. In future work, we are applying correlation-based feature selection methods on other data sets. Currently we are implementing the optimal Immediate and Delayed policies into DT to experimentally evaluate their performance.

### References

1. Mostafavi, B. , Liu, Z. , Barnes, T.: Data-Driven Proficiency Profiling. EDM (2015)
2. Chi, M., VanLehn, K., Litman, D., Jordan, P.: Inducing effective pedagogical strategies using learning context features. In: De Bra, P., Kobsa, A., Chin, D. (eds.) UMAP 2010. LNCS, vol. 6075, pp. 147–158. Springer, Heidelberg (2010)
3. Tetreault, J.R., Litman, D.J.: A reinforcement learning approach to evaluating state representations in spoken dialogue systems. *Speech Commun.* **50**, 683–696 (2008)

# On the Evaluation of the Expert and the Learner Models of Logic-Muse Tutoring System

Roger Nkambou<sup>1,2</sup>, Ange Adrienne Nyamen Tato<sup>1,2</sup>, Janie Brisson<sup>1,2</sup>,  
Clauvice Kenfack<sup>1,2</sup>, Serge Robert<sup>1,2</sup>, and Pamela Kissok<sup>1,2</sup>

<sup>1</sup> Université Du Québec à Montréal, Montréal, Canada  
{nkambou, roger, nyamen\_tato, ange\_adrienne}@uqam.ca  
<sup>2</sup> University of Yaoundé, Yaoundé, Cameroon

**Abstract.** In our previous works, we presented Logic-Muse as an ITS that helps improve logical reasoning skills in multiple contexts. The main purpose of this paper is to present an evaluation of the expert and the learner components. First, we will show how well the expert can reason on conditional reasoning problems by comparing its answers to 72 exercises to those provided by human experts. Then we will demonstrate the prediction capabilities for our learner model (via a Bayesian network) by using datamining techniques on data from 71 students.

**Keywords:** Intelligent tutoring system (ITS) · Evaluation · Reasoning skills · Formal logic learning · Bayesian network

## 1 Introduction

In our previous work [2], we have proposed Logic-Muse, an ITS that helps people improve their reasoning skills. Logic-Muse can provide a rich and personalized learning. It was developed with the active participation of experts in different fields (Cognitive Science, Intelligent tutoring systems, Logic & Psychology of Reasoning). All its three main components (Learner, Tutor and Expert) have been developed while relying on the help of experts as well as based on important past work in the field of reasoning and computer science. Before deploying the system, we have decided to evaluate Logic-Muse in terms of its capabilities for reasoning and solving logical reasoning problems, and capabilities to assess and predict the state of learner skills. The study focuses on a comparison of problem-solving data from human experts and Logic-Muse as well as the use of some relevant data mining techniques for the validation of the learner model.

## 2 Evaluation of the Expert and the Learner Components

The aim of the expert component is to encode and manipulate the domain knowledge. The Logic-Muse expert is able to produce a valid logical reasoning for any given problem. It is also able to recognize fallacies as well as inference suppressions (the two



main categories of reasoning errors human usually made according to the experts) and to correct them. In Logic-Muse, rules (valid and buggy) are encoded using the Java Expert System Shell. From a bank of 72 reasoning problems, we tested the reasoning skills of the Logic-Muse expert by comparing its answers with those provided by human experts. All the 72 exercises in conditional reasoning (18 exercises for each mode of reasoning MPP, MTT, AC, DA) were solved by a human expert and then by the Logic-Muse expert. Results show the overall accuracy of the designed algorithm for Logic-Muse Expert was 100 %.

One of the biggest challenges in designing ITS is the effective assessment and representation of the student's knowledge state and specific needs in the problem domain based on uncertainty information. We use a BN (Bayesian Network) to represent the user's knowledge as accurately as possible. It was built from the domain knowledge, where causal relationships between nodes (reasoning skills) as well as prior probabilities are provided by the experts. It allows diagnosis and modeling of the learner's current knowledge state. To assess the predictive ability of our BN and its ability to best represent the current skills of a learner, we opted for an incremental cross validation. So in the incremental version, the training data increase one by one and the test data decrease one by one. The prediction was done one student at a time by entering, as evidence to the network, the responses of the particular student. For each of the 71 students who answered on 49 logic problems, we have compared the real answer of each question with the one predicted by the network. Results show that, after an average of 10 to 15 questions answered, the Bayesian network is able to predict the behavior of a learner with an accuracy of 95 %. Some errors can be due to the guess (giving a correct answer, despite not having the necessary skills) and slip (having the necessary skills, but giving a wrong answer) parameters [1].

### 3 Conclusion and Future Works

We validated expert and learner models implemented in Logic-Muse. We have demonstrated the expert's reasoning ability and proven predictive efficiency of the BN. We see that the network is able to predict learner knowledge and make a faithful representation of the learner's knowledge state. The prior probabilities in the network will be refined according to the results obtained from this first evaluation. Since we plan to launch the use of Logic-Muse starting in autumn 2016, we will conduct the summative evaluation (regarding the added value of such a system in the learning of logical reasoning) at this time. This evaluation will also assess the pedagogical model.

### References

1. Baker, R.S., Corbett, A.T., Aleven, V.: More accurate student modeling through contextual estimation of slip and guess probabilities in Bayesian knowledge tracing. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 406–415. Springer, Heidelberg (2008)

2. Nkambou, R., Brisson, J., Kenfack, C., Robert, S., Kissok, P., Tato, A.: Towards an intelligent tutoring system for logical reasoning in multiple contexts. In: Conole, G., Klobucar, T., Rensing, C., Konert, J., Lavoué, E., Aristeidou, M., Scanlon, E., Sharples, M. (eds.) EC-TEL 2015. LNCS, vol. 9307, pp. 460–466. Springer, Heidelberg (2015)

# Tools for Improving Teachers' Daily Tasks: Does It Really Help?

Fábio Goulart Andrade, Júlia Marques Carvalho da Silva,  
and Maurício Covolan Rosito

Avenida Osvaldo Aranha, Bento Gonçalves, RS 540, Brazil  
{fabio.andrade, julia.silva,  
mauricio.rosito}@bento.ifrs.edu.br

**Abstract.** Most of the studies about ITS focus on students and their needs. Teachers are often forgotten. On a daily basis, teachers must perform several tasks to ensure that students are learning. Such tasks can take a significant amount of time and usually involve looking for specific students in multiple courses. In order to comprehend and propose solutions to the problem, this paper investigates if the development of new tools could promote a better use of the VLE by teachers-tutors. We developed seven plug-ins focused on assisting teachers in their daily tasks. These tools were introduced in a real Moodle environment and had their impact assessed through a questionnaire and log analysis.

**Keywords:** Tutors · Learning management systems · Moodle

## 1 Introduction

Virtual Learning Environments are widely used in Distance Education and serve as a complement to classroom/hybrid courses [1]. Nonetheless, teachers reported difficulties in using these environments - such as complexity and time required to perform tasks - which end up limiting the interaction with students and the technological potential of the VLE. The present study aimed to address the following question: “Could the development of new tools promote a better use of the VLE by teachers-tutors?”

## 2 Developed Tools

Seven plug-ins for Moodle have been developed to support teachers-tutors. They consist of task reminders, summary of activities and course administration reports.

The first tool provides a summary of tasks that require the attention of a teacher-tutor, displayed on Moodle's homepage. It includes: tasks from the last 30 days, student work not evaluated and unread forum threads. Activities are listed by name, date and course they belong to, plus the number of items already checked and how many still need to be evaluated.

Similarly, the mobile app IFRS-BG Moodle aims to facilitate course management on the VLE through notifications. With it, teachers, students and even parents can be alerted on tests, pending tasks and other activities from all the courses related to them.

Activity reports show the latest interactions performed by users in the VLE. Additionally, three course administration reports measure the performance of tutors and students throughout a course.

### 3 Tools Usage Evaluation

Four plug-ins were made available at IFRS-BG to be evaluated. The first assessment consisted on an acceptance questionnaire administered to 40 teachers. The 35 % answer rate included mostly teachers-tutors from classroom courses (87 %) of higher education (93 %), none of which claimed to have insufficient knowledge of Moodle. More than half of them used the VLE for less than a year.

The reminders on Moodle's homepage were the most popular feature. Users praised the level of clarity and relevance of information (86 % and 79 %, respectively).

It was noted, however, that most of the respondents did not remember Moodle before the new homepage. Thus, 50 % could not confirm or deny any improvement. Indifferent users summed up to 14 %, against 57 % satisfied and 28 % very satisfied.

Then, the analysis of affective dimensions sought to measure aspects of punctuality, commitment, communicability, sociability, initiative and meticulousness of teachers-tutors from the interactions logged in Moodle. Databases from two semesters of 2015 were verified with scales and formulas proposed by Cunha [2].

Several dimensions showed a positive difference between periods, but only meticulousness had a significant improvement (24 %). The use of VLEs in the institution still falls short in several respects, so the impact of the tools merely softened the indicator.

### 4 Conclusions

The developed tools affected precisely the dimension where no teacher-tutor got a satisfactory average (even after the plug-ins). It is believed that this fact is due to the nature of the tools, which aimed to offer information about interactions in the VLE. Through them, teachers could decide when/where their attention is most needed.

It is noteworthy that the information offered on the new tools had its relevance recognized even by users who claimed not to have used them properly. Similarly, the level of satisfaction with the plug-ins has been relatively high, indicating that, they may add some value to the teaching practice should they be incorporated.

### References

1. de Almeida, M.E.B.: Educação a distância na internet: abordagens e contribuições dos ambientes digitais de aprendizagem. In: Educação e Pesquisa, vol. 29, no. 2. FE/USP, São Paulo (2003)
2. da Cunha, F.O.: Análise das dimensões afetivas do tutor em turmas de EaD. No f. 76. Itajaí: UNIVALI, 2009. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação). Universidade do Vale do Itajaí, Itajaí (2009)

# A Brief Overview of Logic-Muse, an Intelligent Tutoring System for Logical Reasoning Skills

Clauvice Kenfack<sup>1,2</sup>, Roger Nkambou<sup>1</sup>, Serge Robert<sup>1</sup>,  
Ange Adrienne Nyamen Tato<sup>1</sup>, Janie Brisson<sup>1</sup>, and Pamela Kissok<sup>1</sup>

<sup>1</sup> UQAM, Montréal, Canada

clauvicek@gmail.com, nkambou.roger@uqam.ca

<sup>2</sup> University of Yaoundé, Yaoundé, Cameroon

**Abstract.** Learning logical reasoning is an important core activity in the field of education. The purpose of this paper is to present an overview of a web-based intelligent Tutoring system (ITS) called Logic-Muse for learning logical reasoning. Logic-Muse helps learners develop reasoning skills on various contents.

**Keywords:** Reasoning skills · Cognitive diagnosis · ITS · Bayesian network

## 1 Introduction

The purpose of this poster is to present an overview of a web-based intelligent Tutoring system (ITS) called Logic-Muse for learning logical reasoning. Logic-Muse provides a learning environment that helps learners develop reasoning skills on various contents. Logic-Muse is design to support reasoning in a wide range of complex domains including several logical systems (classical logic and non-classical logic). This poster provides a briefly description of the architecture of the system.

## 2 Architecture of Logic-Muse

Logic-Muse's architecture is based on a classical ITS including the three usual components: the logic expert, the learner model and the tutor. The learning environment includes a database of reasoning problems of six types of situations (or contexts). The **Logic-Muse expert** implements logical reasoning skills and knowledge as well as related reasoning mechanisms (syntactic and semantic rules of the given logical system). It is able to recognize the reasoning errors and correct it. One of the Expert component abilities is to transform narrative sentence in a logical form using the mechanisms on stemmer snowball based on Porter or Regex Libraries.

The learner model in Logic-Muse has several dimensions. The episodic memory keeps track of all the exercises performed by the learner. The cognitive model is a Bayesian network built from the domain knowledge, where influence relationships between nodes (reasoning skills) as well as prior probabilities are provided by the experts (Tchetagni and Nkambou 2002). Some nodes are directly connected to the reasoning activities or items. The skills are those pointed out by mental models theory

in order to reason in conformity to the logical rules, namely the inhibition of exceptions to the premises, the generation of counterexamples to the conclusion and the ability to manage all the relevant models for the concrete, contrary to fact, abstract informal and abstract formal situations. The items nodes (bottom of the structure) represent inferences about each of the logical rules in each of the reasoning situations.

A CDM-Based psychometric model is built using the item bank, a Q-Matrix (items/skills), as well as data related to all student responses on items. The resulting model allows for initial predictions about the learner strengths and weaknesses regarding reasoning skills given his/her performance on items.

**The Tutor component** in Logic-Muse relies on the expert who is able to solve the case to provide relevant feedback to the learner. The goal is to help the learner to become a better logical reasoner in several situations.

The **Learning interface** provides a rich set of services ranging from simple exercises such as the building of true tables to complex situations involving complex reasoning rules. An overview of the interface is shown in Fig. 1.

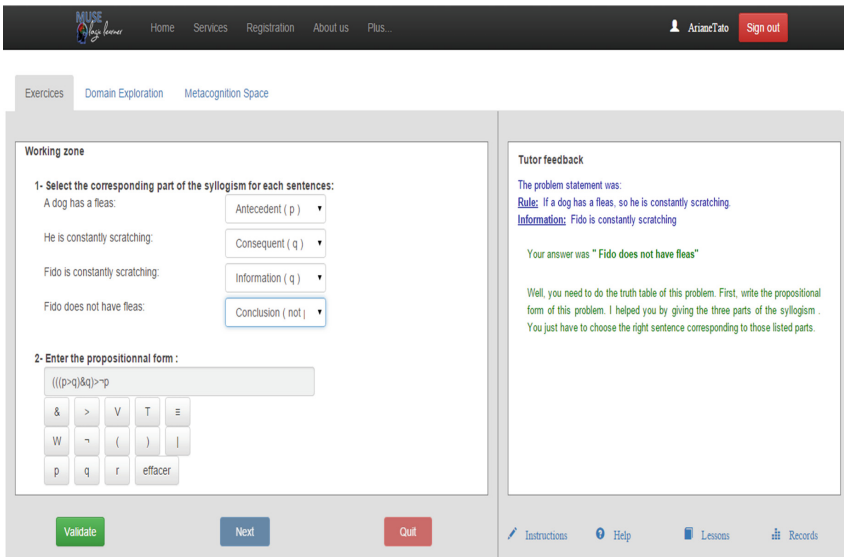


Fig. 1. Logic-muse reasoning service

### 3 Conclusion and Future Works

We have presented an overview of Logic-Muse, which is a multidisciplinary initiative. We have described how the learner Bayesian network is built and can be initialize for a given learner. In the next steps, we intend with the data collected from the student to make a summative evaluation, which help us to determine the effectiveness and adaptability of Logic-Muse in terms of learning environment.

## Reference

1. Tchétagni, J.M., Nkambou, R.: Hierarchical representation and evaluation of the student in an intelligent tutoring system. In: Cerri, S.A., Gouardères, G., Paraguaçu, F. (eds.) ITS 2002. LNCS, vol. 2363, pp. 708–717. Springer, Heidelberg (2002)

# Pilot Study with RALL-E: Robot-Assisted Language Learning in Education

Ning Wang<sup>1</sup> and W. Lewis Johnson<sup>2</sup>

<sup>1</sup> University of Southern California, Los Angeles, USA  
nwang@ict.usc.edu

<sup>2</sup> Alelo Inc., Los Angeles, USA  
ljohnson@alelo.com

## 1 Introduction

Social robots, designed to engage in face-to-face communication, have great potential in language training, because spoken language is a face-to-face communication skill. Early experiences with robotics for language learning have demonstrated the potential of robot-assisted approaches[1]. Social robots have shown promise in research laboratory settings for language education, but historically, they have been too expensive to consider as a relevant educational technology. In this paper, we describe RALL-E (Robot-Assisted Language Learning in Education), a low-cost autonomous social humanoid robot designed to engage learners in complex task-based conversational interactions in a foreign language. The hardware of the RALL-E robot is the Hanson RoboKind R25 model. The RALL-E robot's conversational capability is developed based on the Virtual Role-Player (VRP) architecture [2, 3]. This architecture has been applied in many foreign-language training technologies that can engage in multimodal communication with trainees in a foreign language. The topics RALL-E covers include basic greetings and introductions in Chinese. A learner interacts with RALL-E through natural language in Chinese.

## 2 Pilot Study

To study how learner skills impact the use of RALL-E, we placed the robot in both an introductory Chinese class (Chinese I) of 10 students and an advanced Chinese class (Chinese III) of 47 students in a United States high school. The study was carried out in three 1-h long class periods (one 1-h Chinese I and two 1-h Chinese III). Students interacted with RALL-E in groups of 3–5. The interaction lasted between 5–20 min. One teacher who teaches the Chinese classes participated in the study.

Overall, RALL-E was somewhat successful in engaging students in basic conversational dialogue in Chinese. RALL-E received 3219 voice inputs and was able to recognize 60 % of them. From the recognized input, RALL-E successfully responded to 65 % of them. The other 35 % of the recognized input was considered out of context. For example, if the student and the robot are discussing



music, an utterance about sports is considered out of context. This means that RALL-E was able to provide responses to 39% of the input (60% times 65%). From the 7-point Likert scale usability ratings collected after the study, students considered RALL-E somewhat useful ( $M = 4.12$ ,  $std = 0.98$ ), easy to use ( $M = 4.59$ ,  $std = 0.97$ ), easy to learn ( $M = 5.72$ ,  $std = 1.14$ ) and were somewhat satisfied with practicing Chinese with RALL-E ( $M = 4.54$ ,  $std = 1.15$ ). When asked whether they were interested in using the RALL-E in the classroom when a new version was released, 55% of the students responded “Yes”. Student t-tests showed that the ratings on how easy it was to learn to interact with RALL-E ( $M_{ChineseI} = 6.28$ ,  $M_{ChineseIII} = 5.60$ ,  $p = 0.0024$ ) and overall satisfaction with RALL-E ( $M_{ChineseI} = 5.06$ ,  $M_{ChineseIII} = 4.43$ ,  $p = 0.0185$ ) from Chinese I students were higher than those from Chinese III students. There was no significant difference in ratings of usefulness and ease of use of RALL-E. Fisher’s exact tests showed that, compared to Chinese III students, a higher percentage of Chinese I students signed up to use the next version of RALL-E inside the classroom ( $PCT_{ChineseI} = 90\%$ ,  $PCT_{ChineseIII} = 48\%$ ,  $p = 0.0314$ ). The teacher who participated in the study rated RALL-E’s usefulness as 3.57, ease of use as 4.78, ease of learning as 5.0 and overall satisfaction as 4.29.

### 3 Discussion

In the pilot study, the voice-recognition rate was decent, considering the noisy classroom environment RALL-E operated in. Students were quite impressed with RALL-E’s ability to understand them. The main criticism was that RALL-E did not respond the way students expected it to. This points to the need for improvement in RALL-E’s speech recognition and dialogue management — problems facing both conversational virtual agents and humanoids.

Results also indicated that RALL-E was much better received by beginners (e.g., Chinese I students) than advanced learners (e.g., Chinese III students). Interviews carried out after the study also indicated that different user populations had different needs for RALL-E. For example, Chinese III students requested features to allow them to converse with RALL-E on topics they are interested in, and to have RALL-E adapt to the students’ language proficiency. Students from the Chinese I class, who had a large vocabulary but poor pronunciation, requested that RALL-E provide feedback on their pronunciation.

A possible extension of RALL-E is to further integrate its humanoid features with language learning. Another possible direction is to further enrich the dialogue with RALL-E. Students suggested many topics to converse with RALL-E, and expressed strong desire to go “deeper” into the discussion of a particular topic, and to have RALL-E take more initiative in the conversation. This suggests that students were not only open to conversing and practicing Chinese with RALL-E, but were also looking forward to more of it.

## References

1. Han, J.: Emerging technologies: robot assisted language learning. *Lang. Learn. Technol.* **16**(3), 1–9 (2012)
2. Johnson, W.L., Zaker, S.B.: The power of social simulation for chinese language teaching. In: *Proceedings of TCLT7* (2012)
3. Sagae, A., Johnson, W.L., Valente, A.: Conversational agents in language and culture training. *Conversational Agents and Natural Language Interaction: Techniques and Effective Practices*, pp. 358–377 (2011)

# Adapting Exercise Selection to Learner Self-esteem and Performance

Juliet Okpo, Matt Dennis, Kirsten Smith, Judith Masthoff,  
and Nigel Beacham

University of Aberdeen, Aberdeen, UK  
{r02jao15,m.dennis,r01kas12,j.masthoff,n.beacham}@abdn.ac.uk

## 1 Introduction

Adapting tasks to learner characteristics is essential when selecting appropriate tasks for learners [5]. This paper investigates how humans adapt exercise selection to learner self-esteem (SE) and performance, to allow a future Intelligent Tutoring System (ITS) to use these adaptations. Self esteem is an important factor in learning as it is a significant predictor of academic performance [4]. Previous research adapts task selection to other characteristics e.g. past performance [1], but little work focuses on task selection based on learner personality.

## 2 Design of Studies

In two *User-as Wizard* studies, where participants play the role of the system [3], we showed a validated story conveying the self esteem of a fictional learner ('Kate', developed using a similar method to [2]) and an indication of her past performance. In a  $2 \times 4$  between-subjects design, participants were asked to select the next exercise for Kate, given her self-esteem (high or low) and prior performance at a 10 item  $1 \times 1$  digit multiplication exercise (perfect, good, just passing or fail). Participants selected one exercise from a range (with varied difficulty levels) for Kate to attempt next.

In study 1, five multiplication exercise types were shown that the participants could select for Kate to try next. These were either the same type as before ( $1 \times 1$  digit), a slow-easy method of  $1 \times 2$  digits, a fast-difficult method of  $1 \times 2$  digits, a slow-easy method of  $2 \times 2$  digits or a fast-difficult method of  $2 \times 2$  digits. In study 2, we reduced the number of exercises from five to three, and made the choices easier to understand with explicit difficulty ( $1 \times 1$  digit – *same as before*;  $1 \times 2$  digits – *more difficult*;  $2 \times 2$  digits – *much more difficult*).

Overall, we hypothesized that participants would select a more challenging exercise for learners who performed well than for learners who performed poorly (H1), and participants would select a more challenging exercise for High SE learners than for Low SE learners (H2). We hypothesized that for each performance level, participants would select a different exercise depending on SE level and performance (H3): for the fail condition, we expect participants to select the same exercise again for both levels of self-esteem (H3a); for the 'just passed' and

‘high’ performance condition, we expect participants to select a more challenging exercise for learners with high self esteem than learners with low self esteem (H3b); and for the ‘perfect’ performance condition, we expect participants to select a more difficult exercise for low self esteem and a much more difficult exercise for high self esteem (H3c).

The studies were administered as questionnaires on Amazon Mechanical Turk, with US Participants who passed a Cloze test for English fluency and had an acceptance rate of over 90%. In study 1, there were 242 participants ( $\geq 30$  per condition; 125 Female, 116 Male), 39 were aged 16–25, 139 aged 26–40, 60 aged 41–65, 4 over 65. 30 were students and 19 teachers. In study 2, there were 241 participants (122 Female, 117 Male). 46 were aged 16–25, 129 aged 26–40, 61 aged 41–65, 3 over 65. 34 were students and 9 were teachers.

### 3 Results and Conclusion

In study 1 and study 2, H1 is supported ( $\chi^2$  of Performance  $\times$  Exercise Selected with SE level as layer variable:  $\chi^2(12, 242) = 86.65$ ,  $p < 0.001$ ;  $\chi^2(6, 241) = 155.76$ ,  $p < 0.01$ , respectively). For H2, a  $\chi^2$  test of SE level  $\times$  Exercise Selected with Performance as a layer variable was not significant in either study. H3a is supported in both studies, with learners who failed receiving an exercise of the same difficulty. There is no evidence to support H3b, however for the ‘just passed’ condition, more participants did give the low SE learner an exercise of the same difficulty than for high SE. H3c was not supported.

In conclusion, we did not find robust evidence for SE being taken into account for exercise selection. There may be a trend for low SE learners who ‘just passed’ to receive an exercise of the same difficulty more frequently than high SE learners. On reflection, it could be that the exercise difficulty we chose was too coarse-grained and we will investigate SE again where more gradual changes in difficulty are possible. Future findings should be evaluated by real teachers before encapsulation in an algorithm for use by an ITS.

### References

1. Corbalan, G., Kester, L., van Merriënboer, J.J.: Selecting learning tasks: effects of adaptation and shared control on learning efficiency and task involvement. *CEP* **33**(4), 733–756 (2008)
2. Dennis, M., Masthoff, J., Mellish, C.: The quest for validated personality trait stories. In: *Proceedings of IUI 2012*, pp. 273–276. ACM (2012)
3. Masthoff, J.: The user as wizard: a method for early involvement in the design and evaluation of adaptive systems. In: *5th Workshop on User-centred Design and Adaptive Systems*, pp. 460–469 (2006)
4. Rosenberg, M., Schooler, C., Schoenbach, C., Rosenberg, F.: Global self-esteem and specific self-esteem: different concepts, different outcomes. *ASR*, pp. 141–156 (1995)
5. Tamini, E., Kester, L., Corbalan, G., Spector, J.M., Kirschner, P.A., Van Merriënboer, J.: Designing on-demand education for simultaneous development of domain-specific and self-directed learning skills. *JCAL* **31**(5), 405–421 (2015)

# Do Summaries Support Learning from Post-problem Reflective Dialogues?

Sandra Katz, Patricia Albacete, and Pamela Jordan

Learning Research and Development Center,  
University of Pittsburgh, Pittsburgh, USA  
{katz, palbacet, pjordan}@pitt.edu

**Abstract.** This poster reports on a study that compared three types of summaries at the end of natural-language tutorial dialogues and a no-dialogue control, to determine which type of summary, if any, best predicted learning gains. Although we found no significant differences between conditions, analyses of gender differences indicate that female students benefit most from the most concise summary (restatement of a reflection question and its answer).

**Keywords:** Natural-language tutoring systems · Summarization · Reflection

## 1 Introduction

Natural-language tutoring systems typically wrap up a discussion about a problem or complex question with a summary of the line of reasoning (LOR) that leads to its solution (e.g., [1–3]). However, observations of human tutoring reveal that tutors seldom present complete LOR summaries, or give other types of dialogue summaries. For example, the tutor might remind the student of the main question and its conclusion but leave out the detailed, intermediate LOR. We refer to these as *Conclusion* summaries. Alternatively, the tutor might present the question and its answer, as in *Conclusion* summaries, but add some “take home advice”, such as how the discussion could be applied more generally to similar types of problems. We call these *Advice* summaries.

This poster reports on a study that compared the potential benefits of LOR summaries with these alternative types of dialogue summaries and a no-summary control. We hypothesized that a full LOR summary would be more beneficial for students with low prior knowledge than for higher incoming knowledge students. The former type of student may make more mistakes and need help pulling together the LOR. We hypothesized that a *Conclusion* summary would be more beneficial for students with mid-level incoming knowledge because they are likely to be able to self-explain the connection between the question and its conclusion. Finally, we hypothesized that an *Advice* summary would be more beneficial for high prior knowledge students because they are ready to generalize from a line of reasoning that they may be able to self-generate.

## 2 A Study of Summarization and Student Characteristics

Research Platform. Rimac, a web-based natural-language tutoring system for conceptual physics, served as a research platform for this study. Rimac engages students in conceptual discussions (reflective dialogues) after they solve physics problems [3].

**Participants.** One hundred and ninety students, from three high schools in southwestern Pennsylvania, USA, participated in the study. However, the data from only 96 of these students was complete and used for analysis. Students were randomly assigned to one of the four conditions, within each class: No-summary (24), Line of Reasoning summary (23), Conclusion summary (25), and Advice summary (24).

**Procedure.** The study took place during class. Students took a 21-item pretest that covered dynamics concepts. Students in the summary conditions then solved three problems on paper and, after each problem, reviewed the video of a sample solution and engaged with the automated tutor in several reflective dialogues per problem. Students in the No-summary condition solved an additional problem which was isomorphic to one other problem, to control for time on task. Finally, students took a posttest that was isomorphic with the pretest.

**Results and Discussion.** Students across conditions learned from interacting with the tutor. However, contrary to our hypotheses, there were no interactions between students' prior knowledge and learning gains. Furthermore, there were no significant differences in test gain scores between conditions ( $F(3,92) = 0.289, p = .833$ ). This suggests that end-of-dialogue summarization is not a predictor of learning gains.

Although we did not find aptitude-treatment interactions, we observed differences in gain scores between genders. The mean gain was significantly greater for female students ( $t(94) = 2.096, p = .039$ ). Within conditions, this difference held only for Conclusion summaries ( $t(23) = 2.081, p = .049$ ), with a trend for these summaries to be better for females than males, but only for test items rated as difficult ( $t(23) = 2.000, p = .057$ ).

It is possible that students would learn more from dialogue summaries if they participated in generating them—for example, if the system prompted students to fill in missing pieces of information. We are conducting a study to address this question.

**Acknowledgement.** We thank Dennis Lusetich and Svetlana Romanova for their contributions. This research was supported by the Institute of Education Sciences, U.S. Dept. of Education, through Grant R305A130441. The opinions expressed do not necessarily represent the views of the Institute or the U.S. Dept. of Education.

## References

1. Evens, M., Michael, J.: *One-on-One Tutoring by Humans and Computers*. Psychology Press (2006)
2. Graesser, A.C., et al.: AutoTutor: a simulation of a human tutor. *Cogn. Syst. Res.* **1**(1), 35–51 (1999)
3. Katz, S., Albacete, P.L.: A tutoring system that simulates the highly interactive nature of human tutoring. *J. Educ. Psychol.* **105**(4), 1126 (2013)

# Social Interaction with Intelligent Tutoring Systems: An Investigation of Power and Related Affect

Katharina Roetzer

Department of Philosophy, University of Vienna, Vienna, Austria  
k.roetzer@univie.ac.at

**Abstract.** Social interaction has been stated as a key mechanism in teaching and learning, including tutoring situations. Humans tend to act socially towards computers, artificial systems, and agents. Intelligent Tutoring Systems (ITS) incorporating a pedagogical agent are of special interest for social interaction studies. Our pilot study focuses on how power and related affect influence and shape social interaction with ITS. We use AutoTutor as framework to study this social interaction. By employing Grounded Theory Methodologies (GTM), this study is expected to provide qualitatively constructed and generalizable theories about power and related affect in the social interaction between learner and ITS.

**Keywords:** Intelligent Tutoring Systems · Conversational agents · Social interaction · Power · Affect · Grounded theory

## 1 Introduction

Humans tend to act socially towards computational systems and agents, as in Intelligent Tutoring Systems (ITS) [1, 2]. *Power* is a pivotal concept in social interaction studies, shaping any kind of social interaction [3]. Power is theoretically and empirically related to *affect*: affective expressions are instigated by gains or losses in power, or used to gain or con power. Interestingly, few studies have investigated power in tutoring scenarios (e.g. [4]). We employ AutoTutor [5] as framework to study how power and related affect influence and shape social interaction between learner an ITS in regard to: (i) social roles (the attributed roles of tutor and learner); (ii) interaction and embedded structure (social structures interactions are embedded in); and (iii) social relations (ongoing social interaction and relationships). AutoTutor is an ITS designed to simulate a human tutor interacting with the learner in natural language; an animated conversational agent generates speech, facial expressions and some gestures, assesses learner contributions, gives feedback, and generates pumps, hints, and assertions.

## 2 Methodology

Our study, carried out in the context of a master thesis in cognitive science at the University of Vienna, Austria (EU), is meant to provide a foundation for further research. Three consecutive tutoring sessions of increasing complexity are under development.

The subject matter is the history of the University of Vienna. The sessions build on each other, focusing on significant historical events, their relation, and historical continuity. The tutoring sessions are designed to feature two different instruction modes: a lecture-like mode, in which the tutor provides information on historic events, and cannot be interrupted by the student; and a repetition-like, dialogue-based mode following each of these “mini-lectures”. Here, the learner answers questions concerning the preceding lectures, while the tutor gives feedback, hints and prompts in an engaging and supportive manner, and provides concluding remarks. Each weekly session should take 20–30 min, and is structured to have an introduction, in which the tutor announces the session’s topic and agenda, followed by 2–4 mini-lectures and respective repetition and questions, and a final conclusion by the tutor. Five subjects, native English speakers aged 20–35, should participate in all three sessions.

Data will be collected through unsystematic exploratory observation of the sessions, with additional semi-structured, focused interviews. Following Grounded Theory Methodologies (GTM) [6], it will be analyzed iteratively after each session. Preliminary results will inform development and adaptation of interview guides. The final GTM analysis will comprise three phases: first, a general analysis step considers all data collected to derive possibly multiple and competing concepts; the data will then be contextualized for power and related affect; finally, the derived concepts will be systematically related to “extant” [6] concepts (such as literature on power and affect). By comparing “data with data”, this procedure should support construction of grounded *generalizable theories*, with systematic consideration of relevant extant concepts in later phases of the analysis. Results should serve as foundation for a planned continued development of standardized means to study social interaction with ITS.

**Acknowledgements.** I would like to thank the University of Memphis, especially the staff of the Institute for Intelligent Systems and the Educational Testing Service, for providing and assisting with AutoTutor/ASAT, as well as my supervisor, Paolo Petta.

## References

1. Nass, C., Moon, Y.: Machines and mindlessness: social responses to computers. *J. Soc. Issues.* **56**, 81–103 (2000)
2. Veletsianos, G., Russell, G.: Pedagogical agents. In: Spector, J.M., David Merrill, M., Elen, J., Bishop, M.J. (eds.) *Handbook of Research on Educational Communications and Technology*, pp. 759–769. Springer, Berlin (2014)
3. Kemper, T.D.E.: *Status, Power and Ritual Interaction. A Relational Reading of Durkheim.* Goffman and Collins. Ashgate Publishing Ltd. (2011)
4. Mascarenhas, S., Prada, R., Paiva, A., Hofstede, G.J.: Social importance dynamics: a model for culturally-adaptive agents. In: Aylett, R., Krenn, B., Pelachaud, C., Shimodaira, H. (eds.) *IWA 2013. LNCS*, vol. 8108, pp. 325–338. Springer, Heidelberg (2013)
5. Graesser, A.C., Li, H., Forsyth, C.: Learning by communicating in natural language with conversational agents. *Curr. Dir. Psychol. Sci.* **23**, 374–380 (2014)
6. Charmaz, K.: *Constructing Grounded Theory. A Practical Guide through Qualitative Analysis.* SAGE Publications Ltd. (2006)



# Efficiency vs. Immersion: Interface Design Trade-offs for an Exploratory Learning Environment

Toby Dragon<sup>1</sup>, Mark Floryan<sup>2</sup>, Grayson Wilkins<sup>1</sup>,  
and Thomas Sparks<sup>2</sup>

<sup>1</sup> Department of Computer Science, Ithaca College,  
953 Danby Rd, Ithaca, NY, USA

{tdragon, gwilkin1}@ithaca.edu

<sup>2</sup> Department of Computer Science,  
University of Virginia, Charlottesville, USA

{mrf8t, tws2xa}@virginia.edu

**Abstract.** In this poster, we present a hypothesis involving inherent trade-offs between a user interface designed to promote immersion and a user interface designed to promote efficiency in the context of an Exploratory Learning Environment (ELE). We consider the entire user interface, but also specifically the interface for automated, intelligent coaching. We present an example system that provides both types of interface, and present our plans for experimentation.

**Keywords:** Immersion · Exploratory Learning Environment · Interface design

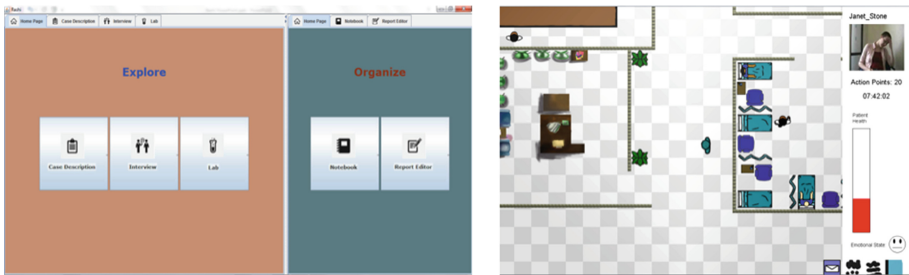
Exploratory Learning Environments (ELEs) naturally promote immersive endogenous experiences [1] in which interesting mechanics (e.g., game mechanics) are intertwined with the activity, and are not simply an independent layer built on top of a tutoring system. This also means that ELEs are inherently complex, which can often lead to floundering, confusion, and/or struggle [2]. We see this as an example of an inherent fact that systems lending themselves to immersive experiences also inherently lend themselves to certain inefficiencies.

Design choices related to user interfaces can increase immersion, but that immersion necessarily implies the existence of features designed to capture engagement, and therefore are not absolutely necessary. On the other hand, interfaces that maximize efficiency remove all unnecessary elements by definition, leaving them potentially uninteresting. We have created two different style interfaces for an ELE focused on inquiry learning in the medical field, Rashi [3]. We present the trade-offs we recognize between immersive and efficient user interfaces, and our plan for experimentation to investigate this phenomenon.

An efficiency-focused interface (Fig. 1, left) is beneficial in that it is simplistic, conforms to convention, offers quick access, is customizable, and allows for coaching in-situ. In this way, coaching can be both present and non-intrusive. However, the

major drawback of this interface is the large number of tools and options immediately available, which can create a steep learning curve, and potentially a dry or disengaging experience. Finally, the ever-present nature of coaching in this type of interface might serve to highlight the situations where the coach is unhelpful or incorrect in support, which is possible in ELEs where the system needs to interpret open-ended input from users [3]. Each of these factors could contribute to students' boredom and frustration.

Alternatively, an immersion-focused interface can encourage a deeper investment in the experience, at the clear expense of efficiency. The primary mechanical difference is the user's need to manually navigate a hospital to find various tools (Fig. 1, right). The virtual world is beneficial in that it may feel more alive and natural than the previous menu system, making actions feel more meaningful and encouraging immersion. This interface can also potentially limit the learning curve by giving the navigation between tools a more natural metaphor. Finally, automated coaching could appear as an embodied agent (a colleague or mentor), potentially accounting for the fallibility of an automated coach.



**Fig. 1.** The efficiency-focused Rashi interface (left), vs. the immersion-focused interface (right).

The detriments of this interface are that metaphor of physical navigation slows down the process of student work, and inherently disallows working with certain tools simultaneously. Coaching is not easily offered in situ, as you will need to visit your “colleague” for advice. Finally, the design and artwork involved in making a high-quality interface in this style limits the ability to customize and configure the types of tools and scenarios available.

Having a constant underlying system with varying interfaces allows for potentially powerful experimentation, where the interface can be the true independent variable. Having multiple interfaces also allows experimentation with student agency over the selection of the interface. Observation of users' interface preferences over time could demonstrate the benefit of multiple interfaces rather than one or the other.

## References

1. Malone, T.W., Lepper, M.R.: Making learning fun: a taxonomy of intrinsic motivations for learning. In: *Aptitude, Learning, and Instruction*, vol. 3, pp. 223–253 (1987)
2. Kirschner, P.A., Sweller, J., Clark, R.E.: Why minimal guidance during instruction does not work. *Educ. Psychol.* **41**(2), 75–86 (2006)
3. Dragon, T., Park Woolf, B., Marshall, D., Murray, T.: Coaching within a domain independent inquiry environment. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) *ITS 2006*. LNCS, vol. 4053, pp. 144–153. Springer, Heidelberg (2006)

# Dynamic Generation of Dilemma-Based Situations in Virtual Environments

Azzeddine Benabbou, Domitile Lourdeaux, and Dominique Lenne

Sorbonne Universités, Université de Technologie de Compiègne, CNRS,  
Heudiasyc UMR 7253, 60203 Compiègne, France  
{azzeddine.benabbou, domitile.lourdeaux,  
dominique.lenne}@hds.utc.fr

**Abstract.** Training in complex environment is not only a difficult task for the learner but it is also a challenging work for simulation systems. These systems need to generate dynamically relevant situations according to the learner's profile. Our work focuses on the generation of situations with critical dimensions for non-technical skills training. Dilemma is one of these dimensions. In this paper we will present our early approach of dynamic generation of dilemma-based situations.

**Keywords:** Dilemma · Virtual environments · Non-technical skills · Training

## 1 Introduction

MacCoy Critical is a project which aims to study and improve training systems that use simulation and virtual environments. A particular attention is paid to using these systems to train for non-technical skills in critical situations. In these situations, there is not always an ideal way to handle them. Training using virtual environments, in this case, may enable the learners to anticipate these situations, to better understand them and finally to weigh each alternative solution to handling them. In risk assessment, the criticality is a numerical value calculated from several parameters which are often: severity and probability. Besides these parameters, we have identified other critical dimensions by means of field analysis and interviews with instructors. These dimensions are: ambiguity, dilemma, socio-cognitive load, newness and learner's ability. In this paper we will focus on the dilemma generation.

## 2 Generation of Dilemma-Based Situations

### 2.1 Related Work

Some work showed interest in creating dilemma situations for training. We can mention [1, 2]. The main remark about these systems is that they are designed in advance and not generated dynamically. However, we can point out the works of [3] who proposed a user model for a system which automatically generates stories based on dilemmas. The dilemma generation process takes into consideration the relations between characters in

order to identify which type of dilemma to put in place. Our approach is slightly different since our purpose is to generate dynamically situations where there is a conflict of values in general and/or contradictory knowledge (not necessarily involving more than one character) leading to difficult-choice-making situations.

## 2.2 Dilemma Generation

In our approach we propose to classify the dilemma into 3 main categories: (1) Situations where the learner has to perform two contradictory tasks, (2) situations where the learner has to make a choice (opposition of moral values) and finally (3) situations leading to the same negative consequences. In order to generate these situations we propose algorithms used by our orchestration engine on the activity or/and causality models. An output example of the algorithm for category (1) generation would be:

*(task1; Verb: "Brake", preconditions: "Red light is on")*  
*(task2; Verb: "Do not brake", preconditions: "Vehicle aquaplaning")*

The dilemma in this situation is that the driver should stop because he must respect the law, but if he does, he risks losing control of his vehicle and may be disastrous consequences.

As far as category (2) is concerned, instead of looking for contradictory tasks, we search for tasks which produce a conflict between two human values. In the activity model, we can tag the tasks. These tags can be used to specify which human value is concerned if the task is accomplished. The algorithm output in this case would be a pair of tasks tagged with conflictual moral values.

The third type of dilemma is a situation where the choice consequence is always negative and the same. To generate this type, we propose to use the causality model. In this model, a "barrier" may be a human behavior which prevents an event from happening. Using our algorithm, the system will be able to find two events which lead to the same negative event. An output example (more brutal than the previous one) would be:

*("The killer threatens John" → "John is dead"; Barrier: "The mother chooses him")*  
*("The killer threatens Tim" → "Tim is dead"; Barrier: "The mother chooses him")*  
*(( "John is dead" OR "Tim is dead") → "The mother loose her son")*

## References

1. Swartout, W., Gratch, J., Hill, R., Hovy, E., Marsella, S., Rickel, J., Traum, D.: Toward virtual humans. *AI Mag.* **27**(1) (2006)
2. Gratch, J., Marsella, S.: A domain-independent framework for modeling emotion. *J. Cogn. Syst. Res.* **5**(4), 269–306 (2004)
3. Barber, H., Kudenko, D.: A user model for the generation of dilemma-based interactive narratives. In: *Optimizing Player Satisfaction Technical Report. AIIDE 2007*, Stanford, California, USA, June 2007

# **Young Researchers Track**

# An Implementation Architecture for Scenario-Based Simulations

Raja Lala<sup>1</sup>, Johan Jeuring<sup>1,2</sup>, and Jordy van Dortmont<sup>1</sup>

<sup>1</sup> Utrecht University, Utrecht, Netherlands

<sup>2</sup> Faculty of Management Science and Technology,  
Open University of the Netherlands, Heerlen, Netherlands

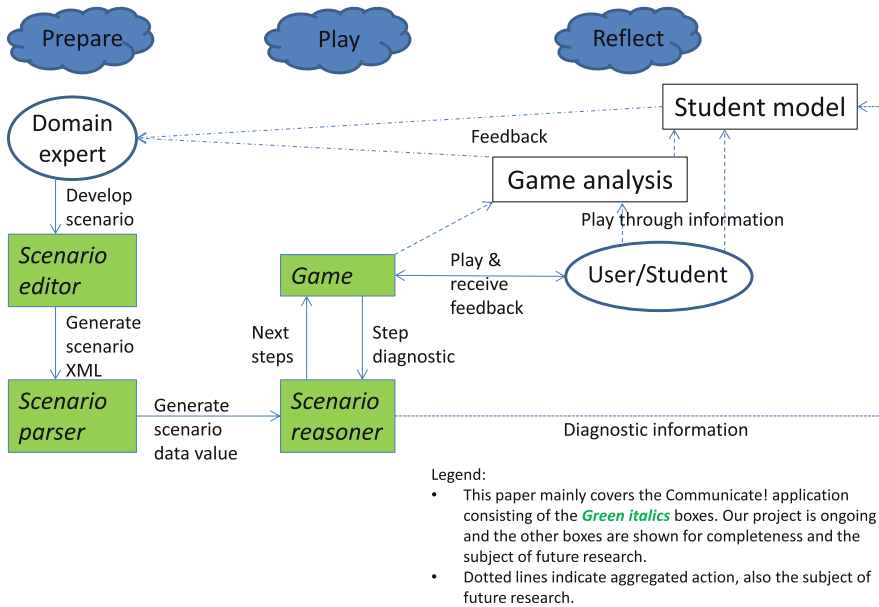
**Abstract.** The past years have witnessed an increased use of applied games for developing and evaluating communication skills. These skills benefit from interpersonal interactions. Providing feedback to students practicing communication skills is difficult in a traditional class setting with one teacher and many students. This logistic challenge may be partly overcome by providing training using a simulation in which a student practices with communication scenarios. A scenario is a description of a series of interactions, where at each step the player is faced with a choice. We have developed a scenario editor that enables teachers to develop scenarios for practicing communication skills. A teacher can develop a scenario without knowledge of the implementation. This paper presents the implementation architecture for such a scenario-based simulation.

Communication skills are best developed in a realistic setting (Realdon et al. 2012). Scripting different ad hoc perspectives is a prerequisite for a narrative structure to reproduce both the flexibility and regularity of communication. A simulation offers an environment for such a realistic situation.

Utrecht University uses a simulation in communication skills courses. Teachers develop communication scenarios in a web-browser based editor and the resulting scenarios are played in the Communicate! application (Jeuring et al. 2015). The simulation is a one-to-one interactive learning environment (Woolf 2010) which provides step-wise feedback to a student. It supports goal-based learning-by-doing (Schank et al. 1993) of communication skills. The simulation has been tested in practice with Psychology, Pharmacy, Medicine & Veterinary medicine students and city council healthcare first-line support employees. Scenario authoring is difficult because a teacher needs to possess pedagogical knowledge, domain understanding and storytelling creativity (Niehaus and Riedl 2011). An important aspect of Communicate! is the de-coupling of scenario development by communication skills experts from the implementation. Thus a domain expert may focus on complex scenario creation.

We distinguish three phases in developing and playing scenarios: prepare, play and reflect. The following figure schematically describes our implementation architecture.

In the Prepare phase a communication expert iteratively develops a scenario in the scenario editor as a directed acyclic graph of steps, and specifies the respective scores and feedback per step. Compared to the GIFT framework (Goldberg et al. 2015) which offers a talking head with a question-answer natural language interface, we focus on scripted communication scenarios.



**Fig. 1.** An implementation architecture of scenario-based simulations

The graph represents the pedagogical communication content knowledge of the expert. It is validated against a schema that describes the structure of scenarios. The scenario parser uses the graph to generate a scenario specific reasoner. At run-time the game interacts with the scenario reasoner, which provides information about the possibilities at each step in the series of interactions. Incremental scores and emotion parameters are fed-back by the reasoner to the game. The game user interface shows a virtual character and an appropriate background location, and uses the game logic to present the game to the user/student.

Usability of authoring environments often comes at the expense of expressiveness (Murray 2003). Our scenario editor tries to combine usability and expressiveness for the domain of communication scenarios. Besides standard sequence, choice, and conditional options, two unique aspects we offer in our scenarios are interleaving (Heeren and Jeuring 2011) and premature endings. Interleaving is particularly useful when students have to perform multiple (sub)tasks, but the order in which these tasks are performed is not important. Premature endings enable a student to skip the following steps in a sequence. Interleaving and premature endings add expressiveness to the editor, and give the author the possibility to obtain a high-level view of a scenario. The editor is implemented in JavaScript and runs in a web-browser, which makes it easily accessible to domain experts.

The Reflect phase is not directly implemented in the Communicate! game, but under development as an independent component that analyses the play-throughs of students and provides insight into student behavior. Effectivity of scenario development, especially using statistical mechanisms like Cronbach’s alpha or RIT (Rasch unit scale) values is also an area for future research.



We compared our editor with four dialogue/scenario editors available in the Unity asset store. These assets range from simple tools without advanced features to advanced tools that need a game-developer to program/simulate the game. One of the primary goals of ITSs is to allow practicing educators to become more involved in their creation (Murray 2003). Communicate! has been well adopted already, and is used by more than twenty teachers/teaching assistants in the above mentioned domains, and played by over a thousand students.

In conclusion, our implementation architecture for communication scenarios allows domain experts to develop scenarios for practicing communication skills without knowledge of the implementation of the simulation.

**Acknowledgements.** This work has been partially funded by the EU H2020 project RAGE (Realising an Applied Gaming Eco-system); <http://www.rageproject.eu/>; Grant agreement No 644187. This paper reflects only the authors' view, the European Commission is not responsible for any use that may be made of the information it contains.

## References

- Goldberg, B., Sottolare, R., Sinatra, A.: Workshop on developing a generalized intelligent framework for tutoring (GIFT): informing design through a community of practice. In: Proceedings AIED 2015. LNCS, vol. 9112, pp. 945–945 (2015)
- Heeren, B., Jeuring, J.: Interleaving strategies. In: Davenport, J.H., Farmer, W.M., Urban, J., Rabe, F. (eds.) MKM 2011 and Calculemus 2011. LNCS, vol. 6824, pp. 196–211. Springer, Heidelberg (2011)
- Jeuring, J., Grosfeld, F., Heeren, B., Hulsbergen, M., IJntema, R., Jonker, V., Mastenbroek, N., van der Smagt, M., Wijmans, F., Wolters, M., van Zeijts, H.: Communicate! — a serious game for communication skills —. In: Conole, G., Klobucar, T., Rensing, C., Konert, J., Lavoué, E. (eds.) EC-TEL 2015. LNCS, vol. 9307, pp. 513–517. Springer, Heidelberg (2015). doi:[10.1007/978-3-319-24258-3\\_49](https://doi.org/10.1007/978-3-319-24258-3_49)
- Murray, T.: An overview of intelligent tutoring system authoring tools: updated analysis of the state of the art. *Authoring Tools for Advanced Technology Learning Environments*, pp. 493–546 (2003)
- Niehaus, J., Li, B., Riedl, M.O.: Automated scenario adaptation in support of intelligent tutoring systems. *Proc. FLAIRS*, **24**, 531–536 (2011)
- Realdon, O., Zurloni, V., Confalonieri, L., Mantovani, F.: Learning communication skills through computer-based interactive simulations. *Emerg. Commun. Stud. New Technol. Pract. Commun.* **9**(c), 276–298 (2012)
- Schank, R.C., Fano, A., Bell, B., Jona, M.: The design of goal-based scenarios. *J. Learn. Sci.* **3** (4), 305–345 (1993)
- Woolf, B.P.: *Building Intelligent Interactive Tutors Student-Centered Strategies for Revolutionizing E-learning*. Morgan Kaufmann (2010)

# A Student-Directed Immersive Intelligent Tutoring System for Language Learning

Jun Seong Choi and Jong H. Park

School of Electronics Engineering,  
Kyungpook National University, Taegu, Korea  
daegulink@naver.com, jhpark@ee.knu.ac.kr

**Abstract.** A self-directed immersive Intelligent Tutoring System is proposed for language learning in pursuit of maximal motivation in students. In this self-directed learning paradigm, the student is given a full freedom in selecting her own courses of learning experience across events (or episodes) unless deviating from the overall pedagogical objective. This high student ownership of learning process is realized with a pedagogy of ‘shepherding’ students instead of didactic teaching or even facilitating. To accommodate an infinite array of immersive learning activities in spite of students’ whimsical choice, a full-blown virtual world is required to provide a corresponding high play affordance. This virtual world is modeled in multitude of layers and dimensions, and is designed to unfold dynamically often coincidentally based on real-world rules often in the forms of emergent situations. These model and design together offer the diversity and realism of situations students can choose to engage themselves in or merely observe. The overall pedagogical objective against this full student autonomy is still fulfilled by separating the learning targets from the episodes and coupling dynamically in situation those targets with the chosen episodes. Beyond the conventional narrative coherency of each episode, a life-long coherency across learning sessions is maintained for each student via historically-contextualized consistent background world inhabited by her corresponding evolving agent.

## Design of Student-Directed Immersive Intelligent Tutoring System

The student’s self-direction in learning process [1] and immersion into simulated reality [2] each have been claimed to be effective for enhancing learning motivation [3] and epistemic efficacy [4], that is, keeping the student interested and improving retention and applicability of learnt knowledge. In a confluence of both learning paradigms, we propose a student-directed, immersive Intelligent Tutoring System (ITS) for language (or foreign language) learning in pursuit of maximum motivation in the students. In specific designs of computer-based education systems, however, each of the two paradigms is implemented in widely varying degrees. The student’s autonomy in learning process has expanded from didactic teaching to directing [5] up to facilitating [1]. As for immersion into virtual world, various virtual reality techniques enable the students to experience different types and levels of realism [6]. In the continuum

coordinate of these two paradigms, our ITS model is designed to permit a maximum student autonomy by merely ‘shepherding’ the freely- grazing students beyond facilitating, and to immerse the student at least cognitively and preferably spatially [6]. This full student ownership of learning process and genuine immersion into virtual-world situations are realized in terms of three key elements: a full-blown virtual world, dynamic presentation and pedagogical mechanism, and language-specific domain knowledge model.

The degree of immersion is proportional to how realistic the contexts the learning takes place are. This realism of learning contexts is pursued in two aspects, the composition of the virtual world and the situations that unfold in the world. The world composition is required to be comprehensive and sophisticated to serve as a life-long target of experience and situated learning for all types of students (and corresponding agents inhabiting the virtual world) [4]. All the events underlying situations in this virtual world are designed to be not just audio-visually realistic but *causally and dynamically* connected with each other. This dynamic world simulation method can achieve high situation variability [7], still won’t overly sacrifice authoring scalability. Further, those situations each student experiences are designed to be *historically coherent throughout her lifetime* in the virtual world by positioning their associated events in a global spatio-temporal coordinate. To depict its complex nature, the entire virtual world (or Cosmos) is modeled in terms of (primitive cosmic units of) entities and their interrelationships. The Cosmos is layered with the Reality, which in turn is composed of the Physical and Social worlds, and, on top of it, the Conceptual Worlds of its inhabitants or agents.  $\text{Cosmos} \langle \mathbf{R}(\mathbf{t}), \{C_i(\mathbf{t})\} \rangle$ ,  $\mathbf{R} = \langle \mathbf{P}, \mathbf{L} \rangle$  where  $\mathbf{R}$ ,  $\mathbf{P}$  and  $\mathbf{L}$  each denote Reality, Physical World and Social World with  $\langle \rangle$  denoting ‘comprises’;  $i = 1, 2, 3, \dots, \#$  of agents; Agent  $i$ ’s Conceptual World  $C_i(\mathbf{t}) = \langle \mathbf{M}_i, \mathbf{K}_i \rangle$  where  $\mathbf{M}$  and  $\mathbf{K}$  denote its Mind and Knowledge, respectively.

Situations as a collective result of events (or episodes) occurring in the virtual world contextualize experiential learning in our ITS [8]. While all the variations of a storyline are conceived and pre-authored in conventional narrative systems [1, 7], all potential events identified by autonomous agents in our virtual world are *dynamically inter-coupled* across independent (planned or natural) events later in the execution time, forming realistic often *emergent* situations [9]. That is, all those concurrent events are designed to be *coincidentally intertwined under real-world rules* such as general causality, numerous physical laws and social regulations like deontic rules and conventional procedure. This dynamic planning also allows the story author to avoid authoring every possible sequence of events in its entirety.

In the domain module, the knowledge elements are inter-linked according to semantic or syntactic associations into a linguistic network. They can be connected in pairs such as antonymy, in layered clusters such as meronymy, up to in large complex clusters such as phrases in thematic relations of event, etc. along with linguistic near-misses. Each clustered subset of associated elements are presented to the student as close to each other as possible with respect to (intra- or inter-) episodes, situations or sessions, to maximum pedagogical effects similar to Mnemonics Link System’s [10]. As a result, these association clusters can be exploited as a general guideline to modulate learning courses together with the individual student’s performance profile.

In order to fulfill the pedagogical objective against the students' whimsical choice of learning courses or episodes, the presentation and pedagogical mechanism separates the learning targets from the episodes and *couples dynamically* those targets with the chosen episodes with reference to each student's performance profile. To present the learning targets in episodes a student has chosen, an elaborate cross-match is to be established between the virtual world and the corresponding domain knowledge. Like the virtual world, the domain knowledge on the language is also hierarchically organized (as a forest) from the levels of phoneme, word and phrase, up to those of sentence and message (e.g., script, letter.) That is,  $e_i^k = G_k(e_j, j = 1, 2, \dots)$ ,  $e_i^k \in E_1 | E_2 | \dots | E_{k-1}$ , where  $e_i^k \in E_k$ , the linguistic element set in the k-th level;  $G_k()$  denotes the part of grammar related to composing the elements in the k-th level. While matching between individual concepts in the virtual world and their corresponding terms (i.e., words or phrases) is rather obvious, other syntactical or semantic matchings are not straightforward, for instance, 'wish' and 'command' in the world might be matched with those Irrealis moods of optative and imperative, respectively. The learning targets are adjusted (i.e., reselected and often reformulated) according to continuously-changing learning situations similar to learning task [11].

Tracking learning process and personalized feedback becomes all the more important when allowing student full autonomy in learning [11]. Without direct teaching or explicit testing times, our ITS logs in stealth the students' performance revealed in verbally inquiring into (parts of) the virtual world or verbally interacting with other agents in the virtual world. The logged performance data is evaluated to *instantaneously update* the student profile modeled as an overlay on (or an imperfect version of) the domain knowledge. This elaborate student model allows the learning processes to be tailored not just to individual students but to their *instantaneous* states.

## References

1. Figueiredo, R., Brisson, A., Aylett, R., Paiva, A.: Emergent stories facilitated. In: Spierling, U., Szilas, N. (eds.) ICIDS 2008. LNCS, vol. 5334, pp. 218–229. Springer, Heidelberg (2008)
2. Bailenson, J.N., et al.: The use of immersive virtual reality in the learning sciences: digital transformations of teachers, students, and social context. *J. Learn. Sci.* **17**(1), 102–141 (2008)
3. Ryan, R., Deci, E.L.: Intrinsic and extrinsic motivations: classic definitions and new directions. *Contemp. Educ. Psychol.* **25**(1), 54–67 (2000)
4. Brown, J.S., Collins, A., Duguid, P.: Situated cognition and the culture of learning. *Educ. Res.* **18**(1), 32–42 (1989)
5. Riedl, M.O., Stern, A.: Believable agents and intelligent story adaptation for interactive storytelling. In: Göbel, S., Malkewitz, R., Iurgel, I. (eds.) TIDSE 2006. LNCS, vol. 4326, pp. 1–12. Springer, Heidelberg (2006)
6. Björk, S., Holopainen, J.: Patterns in Game Design, pp. 206–206. Charles River Media (2004)
7. Charles, F., et al.: Planning formalisms and authoring in interactive storytelling. In: Proceedings of TIDSE, vol. 3. (2003)

8. Itin, C.M.: Reasserting the philosophy of experiential education as a vehicle for change in the 21st century. *J. Exp. Educ.* **22**(2), 91–98 (1999)
9. Cavazza, M., Charles, F., Mead, S.J.: Emergent situations in interactive storytelling. In: *Proceedings of ACM Symposium on Applied Computing (ACM-SAC)*, Madrid, Spain (2002)
10. Einstein, G., McDaniel, M.: Distinctiveness and the mnemonic benefits of bizarre imagery. In: *Imagery and Related Mnemonic Processes*, pp. 78–102. Springer, New York (1987)
11. Hodhod, R., Cairns, P., Kudenko, D.: Innovative integrated architecture for educational games: challenges and merits. In: *Transactions on Edutainment V*, pp. 1–34. Springer, Heidelberg (2011)

# How to Present Example-Based Support Adaptively in Intelligent Tutoring Systems

Xingliang Chen, Antonija Mitrovic, and Moffat Mathews

Intelligent Computer Tutoring Group,  
University of Canterbury, Christchurch, New Zealand  
xingliang.chen@pg.canterbury.ac.nz,  
{tanja.mitrovic,moffat.mathews}@canterbury.ac.nz

**Abstract.** Previous research investigated the effectiveness of Problem Solving (PS), Worked Examples (WE) and Erroneous Examples (ErrEx) with different types of learners. However, there is still no agreement on what kind of learning support (in terms of different learning activities) should be provided to students in Intelligent Tutoring Systems (ITSs) to optimize learning. A previous study found that alternating worked examples and problem solving (AEP) was the best learning strategy compared with worked examples only or problem solving only in the domain of SQL-Tutor [1]. In our recent study, we found that erroneous examples in addition to worked examples and problem solving aid learning in the same domain. The goal of this PhD project is to investigate how SQL-Tutor could maximize learning by adaptively presenting PS, WE or ErrEx to students based on their models.

**Keywords:** Intelligent tutoring system · Worked examples · Erroneous examples · Assistance · Problem-solving · SQL-Tutor

## 1 Introduction

Intelligent Tutoring Systems (ITS) are one of the most effective learning tools that provide adaptive support in terms of feedback, hints or other types of help to students based on their knowledge and ability. SQL-Tutor [2, 3] is an ITS developed by the Intelligent Computer Tutoring Group (ICTG) at the University of Canterbury, New Zealand. It teaches Structured Query Language (SQL). The main learning activity in SQL-Tutor is problem-solving. The system supports students to solve the problems by providing different types of guidance, consisting of positive and negative feedback. While positive feedback focuses on correct actions, negative feedback focuses on errors.

Several recent studies investigated the effects of learning from worked examples compared to learning from tutored problem solving in ITSs; some of those studies found no difference in learning gain but worked examples (WEs) resulted in shorter learning time [4, 5]. McLaren, Lim and Koedinger [6] reported that students learned more efficiently with worked examples than tutored problem solving (TPS) alone. Contrary to that, in a study conducted in SQL-Tutor, Shareghi Najar and Mitrovic [1] found that students learnt more from TPS than from WEs; furthermore, they found that

the best condition was alternating worked examples with problem solving (AEP), which presented isomorphic pairs of worked examples and tutored problem solving to students.

In contrast, there have not been many studies on the benefits of learning from erroneous examples (ErrEx) with ITSs. Tsovaltzi et al. [7] examined the effect of studying erroneous examples of fraction addition and subtraction in the ITS. They found that metacognitive skill for 6<sup>th</sup> graders improved while studying erroneous examples with interactive help. In addition, erroneous examples with interactive help improved 9<sup>th</sup> and 10<sup>th</sup> graders' problem solving skills and conceptual knowledge. In the study of Booth et al. [8], they found that students who explained correct and incorrect examples significantly improved their post-test performance compared to those who only received WEs in the Algebra I Cognitive Tutor. Additionally, the ErrEx condition and the combined WE/ErrEx condition were beneficial for improving conceptual understanding of algebra, but not for procedural knowledge. In our recent study (not published yet), we demonstrated that an improved instructional strategy, a fixed sequence of worked examples/problems pairs and erroneous examples/problems pairs (WPEP), is beneficial for students with different levels of prior knowledge. In addition, we found that students show better performance on problem solving after they learnt from ErrEx than that from WEs.

As mentioned above, previous studies showed the beneficial effect of adding WEs to tutored problem solving. However, what learning material to provide to different kinds of students within Intelligent Tutoring Systems (ITSs) is still an open question. Therefore, the current ITSs should be enhanced in order to use new adaptive strategies. Additionally, classifying students as novices or advanced students is another important problem to deal with at early stages of learning. We propose to generate machine learning classifiers (using data collected from previously from SQL-Tutor) to predict whether a new student is a novice or an advanced student. In the future, we plan to explore how to maximize learning by adaptively presenting problem solving (PS), worked examples (WE) or Erroneous Examples (ErrEx) to students based on their performance.

## 2 Methodology

Prior studies indicated that different levels of assistance were necessary for students to support their learning effectively [9], and therefore such assistance should be presented adaptively in ITSs. Kalyuga and Sweller [10] developed an adaptive e-learning environment for using worked examples using Cognitive Efficiency (CE) to model students' cognitive load and performance. Shareghi Najar et al. [11] investigated an adaptive strategy which presented learning support based on learners' performance on a previous problem and Cognitive Efficiency. Both studies demonstrated positive outcomes using Cognitive Efficiency as a combined measure for assessing the performance of students.

In terms of this study, specific types of learning materials will be presented to identify students (e.g., novices, advanced students) based on previous performance and cognitive efficiency. For example, if a student was identified as a novice, the system

just presents worked examples, erroneous examples, based on their performance on previous problem and CE. When the student reaches the pre-set score, the system will present tutored problem solving based on their student model. We will design a Web-based pre-test in order to initially classify novices and advanced students. A new version of SQL-Tutor will discover the patterns in student's attempts, and record the violation or satisfaction of the constraints. SQL-Tutor will calculate CE based on such patterns, and therefore different levels of learning material will be presented to them based on their prior level of knowledge and CE. We hypothesize that such adaptive strategy would demonstrate better outcomes in comparison with WPEP reported in our recent study.

## References

1. Shareghi Najar, A., Mitrovic, A.: Examples and tutored problems: how can self-explanation make a difference to learning? In: Lane, H., Yacef, K., Mostow, J., Pavlik, P. (eds.) AIED 2013. LNCS, vol. 7926, pp. 339–348. Springer, Heidelberg (2013)
2. Mitrović, A.: Experiences in implementing constraint-based modeling in SQL-Tutor. In: Goettl, B.P., Halff, H.M., Redfield, C.L., Shute, V.J. (eds.) ITS 1998. LNCS, vol. 1452, pp. 414–423. Springer, Heidelberg (1998)
3. Mitrovic, A.: An intelligent SQL tutor on the web. *Artif. Intell. Educ.* **13**, 173–197 (2003)
4. Schwonke, R., Renkl, A., Krieg, C., Wittwer, J., Alevén, V., Salden, R.: The worked-example effect: not an artefact of lousy control conditions. *Comput. Hum. Behav.* **25**(2), 258–266 (2009)
5. McLaren, B.M., Isotani, S.: When is it best to learn with all worked examples? In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) AIED 2011. LNCS, vol. 6738, pp. 222–229. Springer, Heidelberg (2011)
6. McLaren, B.M., Lim, S.-J., Koedinger, K.R.: When and how often should worked examples be given to students? new results and a summary of the current state of research. In: Proceedings of 30th Annual Conference of the Cognitive Science Society, pp. 2176–2181 (2008)
7. Tsovaltzi, D., McLaren, B.M., Melis, E., Meyer, A.-K.: Erroneous examples: effects on learning fractions in a web-based setting. *Technol. Enhanced Learn.* **4**(3–4), 191–230 (2012)
8. Booth, J.L., Lange, K.E., Koedinger, K.R., Newton, K.J.: Using example problems to improve student learning in algebra: differentiating between correct and incorrect examples. *Learn. Instr.* **25**, 24–34 (2013)
9. Koedinger, K.R., Alevén, V.: Exploring the assistance dilemma in experiments with cognitive tutors. *Educ. Psychol. Rev.* **19**(3), 239–264 (2007)
10. Kalyuga, S., Sweller, J.: Rapid dynamic assessment of expertise to improve the efficiency of adaptive e-learning. *Educ. Tech. Res. Dev.* **53**(3), 83–93 (2005)
11. Najar, A.S., Mitrovic, A., McLaren, B.M.: Adaptive support versus alternating worked examples and tutored problems: which leads to better learning? In: Dimitrova, V., Kufflik, T., Chin, D., Ricci, F., Dolog, P., Houben, G.-J. (eds.) UMAP 2014. LNCS, vol. 8538, pp. 171–182. Springer, Heidelberg (2014)



# The Automatic Generation of Knowledge Spaces from Problem Solving Strategies

Ivica Milovanović and Johan Jeuring

Utrecht University,  
Princetonplein 5, 3584 CC Utrecht, Netherlands  
{i.milovanovic,J.T.Jeuring}@uu.nl  
<http://ideas.cs.uu.nl>

**Abstract.** In this paper, we explore theoretical and practical aspects of the automatic generation of knowledge spaces from problem solving strategies. We show how the generated spaces can be used for adapting strategy-based problem solving learning environments (PSLEs).

**Keywords:** Intelligent tutoring systems · Knowledge space theory · Strategies · Student modelling

Intelligent Tutoring Systems (ITSs) can be almost as effective as human tutors in supporting learning problem solving [4]. Most problems are solved incrementally, step by step, by applying a certain problem solving strategy. During the last decade a domain specific language (DSL) for explicitly modelling such strategies has been developed [2]. Rules in this strategy language describe how exercise objects can be transformed. The language defines a number of operators to explicitly model a sequence and a choice of rules or strategies, recursive application etc. Strategies form a hierarchical tree structure and rules are the leaves of those trees. The strategy language has been applied to building a number of intelligent tutoring systems and serious games, and for providing feedback in existing educational environments. We are currently exploring how the structure of different graph-based student models can be automatically generated from a strategy for solving a particular class of problems. In this paper we present theoretical and practical aspects of the automatic generation of one such model, namely a fine-grained learning space, for enabling adaptive learning and assessment in strategy-based problem solving environments.

Knowledge space theory (KST) is a mathematical framework for describing feasible knowledge states of a student [1]. A knowledge domain can be divided into knowledge components, such as skills, competences, exercise items, etc. A knowledge state is a feasible subset of those components. A knowledge space is the set, closed under union, of all the feasible knowledge states. Knowledge space  $\mathbb{S}$  is a learning space if, for each non-empty state  $S \in \mathbb{S}$ , there exists at least one  $c \in S$  for which  $S \setminus \{c\} \in \mathbb{S}$ . Each state in a learning space is fully specified by its two fringes, the inner fringe, containing the most advanced concepts of the state, and the outer fringe, containing the concepts that can be learned next. KST has

been shown to be an excellent framework for both assessment of knowledge and adaptation in a number of PSLEs and serious games [1, 3].

Rules and strategies written in the strategy language by Heeren and Jeuring describe valid sequences of steps for solving a given problem. We show how strategies also divide procedural knowledge of a domain into a set of hierarchical knowledge components which form a knowledge space. We define knowledge of a strategy as the ability of a student to apply it to any exercise object from its domain. In other words, knowing a strategy means knowing at least one of the derivations generated by the strategy, for any exercise object. In the core strategy language, a strategy can be expressed as either the sequence or choice of its sub-strategies, recursive application of a single sub-strategy or as an application of a single sub-strategy to a subexpression. Let  $s$  be a *sequence* of sub-strategies  $s_1, s_2 \dots s_n$ . Then  $s$  is a knowledge component with  $s_1, s_2 \dots s_n$  as its prerequisites and also the inner fringe of the state  $\{s\} \cup \{s_1, s_2 \dots s_n\}$ . Let  $c$  be a *choice* between the sub-strategies  $c_1, c_2 \dots c_n$  and let  $\mathbb{C}$  be a set of all the minimal subsets of  $\{c_1, c_2 \dots c_n\}$  that are sufficient for solving all the objects solvable by  $c$ . Then any of the subsets of  $\mathbb{C}$  is a valid knowledge state and an alternative prerequisite of any strategy for which  $c$  is a prerequisite. Let  $r$  be a *recursive* application of  $s_r$ . Then  $r$  is a knowledge component with  $s_r$  as its prerequisite. Finally, let  $sub$  be an application of  $s_{sub}$  to a subexpression. Then  $sub$  is a knowledge component with  $s_{sub}$  as its prerequisite. The entire knowledge space can be generated by recursively applying the previous definitions to a top level strategy. The generated space is also a learning space. In addition to being used for adaptive assessment, as described in [1], the generated space can be used for adaptive learning. At each state, feedback can be generated at the granularity level of its inner fringe. The outer fringe of each state defines rules and strategies a student is ready to learn. To select the appropriate next exercises, we need to efficiently query available exercise objects. We propose, but have not implemented yet, a data structure equivalent to an inverted index, with a dictionary consisting of derivations and posting lists consisting of suitable starting objects for a given derivation.

## References

1. Falmagne, J.C., Albert, D., Doble, C., Eppstein, D., Hu, X.: Knowledge Spaces: Applications in Education. Springer Publishing Company, Incorporated (2013)
2. Heeren, B., Jeuring, J., Gerdes, A.: Specifying rewrite strategies for interactive exercises. *Math. Comput. Sci.* **3**(3), 349–370 (2010)
3. Reimann, P., Kickmeier-Rust, M., Albert, D.: Problem solving learning environments and assessment: a knowledge space theory approach. *Comput. Educ.* **64**, 183–193 (2013)
4. VanLehn, K.: The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educ. Psychol.* **46**(4), 197–221 (2011)

# Using Multi-channel Data to Assess, Understand, and Support Affect and Metacognition with Intelligent Tutoring Systems

Michelle Taub and Roger Azevedo

Department of Psychology, North Carolina State University, Raleigh, NC, USA  
{mtaub, razeved}@ncsu.edu

Research on self-regulated learning (SRL) has revealed that when students engage in cognitive, affective, metacognitive, and motivational (CAMP) processes, it can positively impact their learning, however, studies have also revealed that students do not typically deploy CAMP processes during learning, and therefore fail to benefit from using these strategies. As such, researchers are developing intelligent tutoring systems (ITSs) that are designed to track, model, and foster the effective use of CAMP SRL processes by providing timely, individualized adaptive scaffolding to assist students with using particular types of cognitive and metacognitive SRL strategies. ITSs have been developed that address affect and metacognition specifically, however there is a limited amount of empirical research using multi-channel data to investigate the impact of how both affect and metacognition impact students' SRL during learning with these ITSs. There is much research that needs to continue to be conducted in the area of affect and metacognition with ITSs, however there are also theoretical, methodological, analytical, and design issues that need to be considered. This paper will address these issues, as well as introduce research questions used to address these issues with two ITSs: MetaTutor and CRYSTAL ISLAND.

The proposed research will address questions, such as: (1) what are the individual and relative contributions of multi-channel data to understanding the influence of affect and metacognition on measures of complex learning and scientific reasoning? (2) What are the key features of affect and metacognitive processes during learning? For example, the duration, fluctuation, dynamics, sequence, etc. of individual affective states and metacognitive processes. In addition, we will also focus on more specific questions, such as: (1) are there significant differences in the proportion of time spent fixating on areas of interest (AOIs) between prior knowledge? This will be based on eye tracking data to determine where students were fixating on the key interface elements (e.g., pedagogical agent modeling emotion regulation strategy) during learning with MetaTutor; or (2) Is there a relationship between concept matrix attempts and proportion of fixations on book content, and does this relationship depend on the proportion of fixations on book concept matrices during the scientific reasoning process

in CRYSTAL ISLAND? In sum, the proposed research aims to capture and analyze multichannel data using multi-level modeling (MLM) from MetaTutor and CRYSTAL ISLAND to (1) build a unifying framework of affect and metacognition, and (2) a blue print for designing ITSs capable of accurately detecting, tracking, modeling, and fostering affect and metacognition.

# AMNESIA, a Dynamic Environment for Progressive Assessment of Cognitive Functions

Asma Ben Khedher and Claude Frasson

Département d'informatique et de recherche opérationnelle,  
Université de Montréal, 2920 Chemin de la Tour, Montréal H3T-1J4, Canada  
{benkheda, frasson}@iro.umontreal.ca

The study of the cognitive processes as memory and problem solving has been a motivating research domain. In this context, we designed a virtual environment AMNESIA where a series of cognitive tasks are provided to a participant starting from simple memory exercises to logic tests, until clinical reasoning (medical cases' resolution). The environment was implemented with Unity, a development platform for creating 3D games and the designed medical cases were inspired from the hypothetico-deductive clinical reasoning. The user should make early hypotheses and confirms or refutes them using additional clinical data. Then, he makes a final diagnosis with a prescription of the adequate treatment.

While the user interacts with the environment, we record his gaze data and extract different eye movements' metrics. We focus in our work on the scanpath metric since it allows us to trace the reasoning process followed by a user, in order to assess the correctness of his reasoning. In addition, we measure the brain activity through two mental states, namely engagement and mental workload.

Figure 1 shows the reasoning environment in which we identify the scanpath of the user's eye movements that represents his reasoning process (the red circles). We can also visualize the distribution of the user's attention that allows us to identify if the

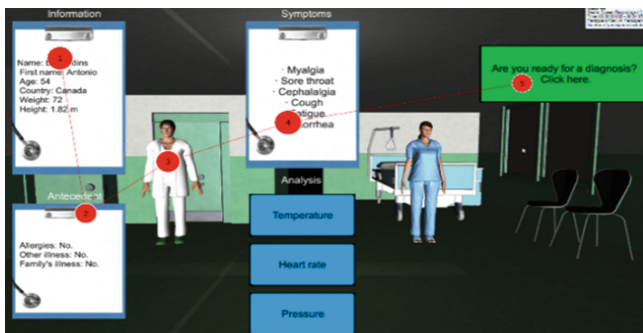


Fig. 1. Example of a scanpath

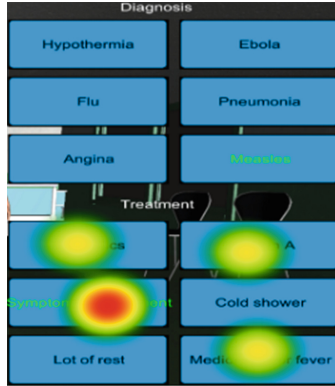


Fig. 2. Heat map visualization

student focused on a specific diagnosis or treatment (Fig. 2). Bigger red zones indicate higher focus. The first obtained results revealed that eye movements are reliable indicators of cognitive processes' assessment.

Our future work in this context is to run experiments with novice medical students and integrate EEG measures with the eye tracking.

# Comparisons of Different Types of Feedback of Linear Equation Aide (LEA): A Mobile Assisted Learning Application on Linear Equations

Rex P. Bringula, Jan Sepli De Leon, Bernadette Anne Pascual,  
Kharl John Rayala, Kevin Sendino, and Marc Rodin Ligas

College of Computer Studies and Systems,  
University of the East, Manila, Philippines  
rex\_bringula@yahoo.com, marc\_ligas@yahoo.com,  
deleonjansepli@gmail.com, pascualbeaa@gmail.com,  
kharljohnrayala@gmail.com,  
kevin\_sendino@ymail.com

**Abstract.** A feedback is intended to show students what to do, or point out some faults in the student's solution to correct. Giving a feedback to a student is the key to an effective learning of a student and it may have an impact to the student learning achievements [2]. In this study, four forms of feedback will be implemented. The first form of feedback is through giving hints [1]. Hints are clues in correcting a wrong move in the solution. The second feedback is corrective, that is, the software directly gives the correct step in a wrong move. The third feedback is given when all solutions are entered by the learner and the software will indicate which step of the solution is incorrect. The last feedback will only indicate whether the final answer is correct or not. This study will determine which of the four different feedbacks will have significant effect on solving simple linear equations. The software will be implemented in one high school in Manila. One class section of Grade 7 students will be assigned to each type of feedbacks. Participants will use the software for the span of three days. Pretest and posttest will be administered before and after the intervention period, respectively. Analysis of variance will be employed to determine significance of the findings. Conclusions, recommendations, and future research directions will be presented based on the findings.

**Keywords:** Feedback · Hint · Intelligent tutoring system

## References

1. Bringula, R.P., Alvarez, J.N.S., Evangelista, M.A.C., So, R.B., Gatus, M.M.: Technical description of equation sensei: a mobile-assisted learning application in mathematics. In: Sampson, D.G., Huang, R., Hwang, G.-J., Liu, T.-C., Chen, N.-S., Kinshuk, Tsai, C.-C. (eds.) IEEE 15th International Conference on Advanced Learning Technologies (ICALT 2015), pp. 105–107. IEEE Computer Society (2015)
2. Hattie, J., Timperley, H.: The power of feedback. *Rev. Educ. Res.* **77**, 81–112 (2007)

## Author Index

- Abdel Razek, Mohammed 500  
Acharya, Sabita 227  
Addimando, Loredana 471  
Adewoyin, Oluwabunmi 286  
Aksit, Osman 165  
Albacete, Patricia 519  
Al-Hamoud, Abdullah 491  
Aleven, Vincent 90, 396, 409, 485  
Alkurtass, Issa 491  
Allessio, Danielle 314, 474  
Alotaibi, Shaikhah 455  
Alsanie, Waleed 491  
Alshammari, Mohammad 498  
AlZoubi, Omar 389  
Anane, Rachid 498  
Andrade, Fábio Goulart 509  
Andrade, Fernando R.H. 176  
Andres, Juan Miguel L. 234  
Andres, Juliana Ma. Alexandra L. 234  
Antonucci, Alessandro 471  
Araya, Roberto 286  
Arroyo, Ivon 314, 474, 488  
Asensio-Pérez, Juan I. 439  
Azevedo, Roger 34, 197, 240, 273, 368, 543
- Barbalios, Nikolaos 111, 327  
Barnes, Jayden 430  
Barnes, Tiffany 347, 504  
Barrón-Estrada, María Lucía 453  
Beacham, Nigel 517  
Beck, Joseph E. 234  
Benabbou, Azzeddine 526  
Benlamine, Mohamed S. 494  
Bernard, Jason 334  
Bier, Norman 327  
Biswas, Gautam 187  
Blaum, Dylan 341  
Bonesana, Claudio 471  
Boscolo, Clelia 307  
Bote-Lorenzo, Miguel L. 439  
Bouchet, François 368  
Boyer, Kristy Elizabeth 154
- Brajković, Emil 469  
Bringul, Rex P. 547  
Brisson, Janie 506, 511  
Britt, M. Anne 341  
Bull, Susan 307  
Burleson, Winslow 314, 474  
Busetto, Alberto-Giovanni 79  
Butler, Eric 320
- Carlotto, Talvany 301  
Carneiro, Gustavo 430  
Carvalho da Silva, Júlia Marques 509  
Cassell, Justine 423  
Chaabouni, Mariem 458  
Chachoua, Soraya 467  
Chang, Ting-Wen 334  
Chaouachi, Maher 494  
Chen, Xingliang 13, 538  
Chi, Min 208, 504  
Choi, Jun Seong 534  
Choquet, Christophe 458  
Corbett, Albert 133  
Cristea, Alexandra I. 294
- De Leon, Jan Sepli 547  
De Medio, Carlo 375  
Demetriadis, Stavros 260, 280  
Demi, Sandra 409  
Dennis, Matt 517  
Di Eugenio, Barbara 227, 389  
Dragon, Toby 523  
Dufresne, Aude 494
- Eagle, Michael 133  
Estraillier, Pascal 467, 480
- Fabic, Geela 447  
Fenza, Giuseppe 144  
Floryan, Mark 523  
Fossati, Davide 227, 389  
Fox, Armando 122



- Frasson, Claude 382, 494, 500, 545  
 Furukubo, Kazutoshi 464
- Gal, Ya'akov (Kobi) 443  
 Gale, William 430  
 Gasparetti, Fabio 375  
 Gauch, Brian 187  
 Ghali, Ramla 382  
 Ghezala, Henda Ben 458  
 Ginon, Blandine 307  
 Gluz, João C. 441  
 González-Hernández, Francisco 453  
 Graf, Sabine 334  
 Grafsgaard, Joseph F. 154, 197, 273  
 Green, Nick 227, 389  
 Gross, Markus 79  
 Grubišić, Ani 469  
 Guetta, Naor 443
- Harley, Jason M. 368  
 Harsley, Rachel 227, 389  
 Hastings, Peter 341  
 Hayashi, Yugo 254  
 Hayashi, Yusuke 464, 478  
 Hendley, Robert J. 498  
 Hirashima, Tsukasa 464, 478  
 Ho, Hoang Nam 480  
 Hosseini, Roya 327  
 Hughes, Simon 341  
 Huse, Nico 69
- Ikeda, Mitsuru 354  
 Iksal, Sébastien 502  
 Isotani, Seiji 176
- Jacovina, Matthew E. 59, 476  
 Jaques, Patrícia A. 301  
 Jeurig, Johan 531, 541  
 Johnson, Amy M. 476  
 Johnson, Matthew D. 307  
 Johnson, W. Lewis 514  
 Jordan, Pamela 519
- Kacem, Ahmed Hadj 221  
 Käser, Tanja 79  
 Katz, Sandra 519  
 Kautzmann, Tiago Roberto 301  
 Kelly, Craig 247  
 Kenfack, Clauvica 506, 511
- Khedher, Asma Ben 545  
 Kissok, Pamela 506, 511  
 Klingler, Severin 79  
 Koedinger, Kenneth R. 111  
 Kohn, Juliane 79  
 Kumar, Amruth N. 101, 416  
 Kumar, Rohit 48
- Lala, Raja 531  
 Laroussi, Mona 458  
 Le, Nguyen-Thinh 69  
 Lee, Hope 430  
 Lenne, Dominique 526  
 Lester, James C. 154, 165, 240  
 Ligas, Marc Rodin 547  
 Limongelli, Carla 361, 375, 461  
 Lin, Chen 208, 504  
 Lin, Shuqiong 327  
 Liu, Yuting 488  
 Liu, Zhongxiu 347  
 Lombardi, Matteo 361, 461  
 Long, Yanjin 90  
 Lourdeaux, Domitile 526
- Maass, Jaclyn K. 247  
 MacLaren, Benjamin 133  
 Madaio, Michael A. 423  
 Maeda, Kazushige 464  
 Magnisalis, Ioannis 280  
 Malki, Jamal 467  
 Mangili, Francesca 471  
 Marani, Alessandro 361, 461  
 Martin, Seth A. 197, 273  
 Martínez-Monés, Alejandra 439  
 Masthoff, Judith 517  
 Mathews, Moffat 13, 538  
 Matsuda, Noboru 111, 327  
 May, Madeth 502  
 Mazidi, Karen 23  
 McLaren, Bruce M. 133  
 McNamara, Danielle S. 59, 476  
 Millar, Garrett C. 197, 240, 273  
 Milovanović, Ivica 541  
 Min, Wookhee 165  
 Mitchell, Aaron 133  
 Mitrovic, Antonija 13, 447, 538  
 Miwa, Kazuhisa 3  
 Mizoguchi, Riichiro 176, 354  
 Mondragon, Aydée Liza 402  
 Mostafavi, Behrooz 347, 504

- Mott, Bradford W. 165  
 Mudrick, Nicholas V. 197, 240, 273  
 Muldner, Kasia 314, 474  
 Muñoz-Cristóbal, Juan A. 439
- Neshatian, Kourosh 447  
 Nkambou, Roger 402, 506, 511  
 Nowakowski, Samuel 480  
 Nyamen Tato, Ange Adrienne 511
- Oakden-Rayner, Luke 430  
 Ogan, Amy 423  
 Okpo, Juliet 517  
 Olsen, Jennifer K. 485  
 Omheni, Nizar 221  
 Orciuoli, Francesco 144  
 Ortega-Arranz, Alejandro 439  
 Ouellet, Sébastien 382
- Palaoag, Thelma D. 234  
 Papadopoulos, Pantelis M. 280  
 Papoušek, Jan 267  
 Park, Jong H. 534  
 Pascual, Bernadette Anne 547  
 Passerino, Liliana M. 441  
 Pavlik Jr., Philip I. 247  
 Pelánek, Radek 267, 451  
 Piau-Toffolon, Claudine 458  
 Poirier, Pierre 402  
 Popescu, Elvira 334  
 Popescu, Octav 409  
 Popović, Zoran 320
- Rabah, Mourad 480  
 Ramamurthy, Anya 111  
 Rayala, Kharl John 547  
 Reyes-García, Carlos A. 453  
 Řihák, Jiří 451  
 Ringenberg, Michael 409  
 Robert, Serge 506, 511  
 Roetzer, Katharina 521  
 Rodrigo, Ma. Mercedes T. 234, 449  
 Rosito, Maurício Covolan 509  
 Rowe, Jonathan 240  
 Roy Choudhury, Rohan 122  
 Roy, Matthew 48
- Rubegni, Elisa 471  
 Rummel, Nikol 485
- Šarić, Ines 469  
 Schultz, Sarah E. 314, 474, 488  
 Schwarz, Gustavo 441  
 Sciarrone, Filippo 361, 375  
 Segal, Avi 443  
 Sendino, Kevin 547  
 Sewall, Jonathan 396, 409  
 Shani, Guy 443  
 Shen, Shitian 504  
 Shi, Lei 294  
 Shibayama, Kazuya 3  
 Smith, Andy 165  
 Smith, Kirsten 517  
 Snow, Erica L. 59  
 Solenthaler, Barbara 79  
 Sparks, Thomas 523  
 Stamper, John 133  
 Stanislav, Vít 267  
 Stankov, Slavomir 469  
 Stylianides, Gabriel J. 111  
 Suleman, Raja M. 354  
 Supianto, Ahmad Afif 478  
 Sutner, Klaus 327
- Taboul, Amir 443  
 Tanner Jackson, G. 59, 476  
 Tarau, Paul 23  
 Taub, Michelle 34, 197, 240, 273, 543  
 Tegos, Stergios 260  
 Temperini, Marco 361, 375  
 Terai, Hitoshi 3  
 Tighe, Elizabeth L. 476  
 Tomaš, Suzana 469  
 Torlak, Emina 320  
 Tsiatsos, Thrasylvoulos 260
- Vail, Alexandria K. 154  
 van Dortmund, Jordy 531  
 van Velsen, Martin 327, 409  
 Vasa, Hardik 327  
 Vasić, Daniel 469  
 Vassileva, Julita 286, 455  
 Vea, Larry A. 449  
 Volarić, Tomislav 469  
 von Aster, Michael 79

Wagner, Angela 133  
Wallace, Patricia 341  
Wang, Ning 514  
Weerasinghe, Amali 430  
Wiebe, Eric N. 154, 165  
Wilkins, Grayson 523  
Wixon, Naomi 314, 474, 488  
Woolf, Beverly 314, 474

Yamamoto, Sho 464  
Yin, Hezheng 122  
  
Zaffalon, Marco 471  
Zatarain-Cabada, Ramón 453  
Zhao, Zhengzheng 111  
Žitko, Branko 469