Lecture Notes in Artificial Intelligence 6170

Edited by R. Goebel, J. Siekmann, and W. Wahlster

Subseries of Lecture Notes in Computer Science

Ashok K. Goel   Mateja Jamnik
N. Hari Narayanan (Eds.)

# Diagrammatic Representation and Inference

6th International Conference, Diagrams 2010
Portland, OR, USA, August 9-11, 2010
Proceedings

Springer

Series Editors

Randy Goebel, University of Alberta, Edmonton, Canada
Jörg Siekmann, University of Saarland, Saarbrücken, Germany
Wolfgang Wahlster, DFKI and University of Saarland, Saarbrücken, Germany

Volume Editors

Ashok K. Goel
School of Interactive Computing, Georgia Institute of Technology
Design & Intelligence Laboratory
85 Fifth Street NW, Atlanta, GA 30308, USA
E-mail: goel@cc.gatech.edu

Mateja Jamnik
University of Cambridge, Computer Laboratory
JJ Thomson Avenue, Cambridge, CB3 0FD, UK
E-mail: Mateja.Jamnik@cl.cam.ac.uk

N. Hari Narayanan
Auburn University, Department of Computer Science & Software Engineering
345 W. Magnolia Street, Auburn, AL 36849-5347, USA
E-mail: naraynh@auburn.edu

Library of Congress Control Number: 2010930707

CR Subject Classification (1998): H.5, H.4, I.2, H.5.2, K.4, F.4.1

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-642-14599-X Springer Berlin Heidelberg New York
ISBN-13 978-3-642-14599-5 Springer Berlin Heidelberg New York



springer.com


Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

# Preface

The 6th International Conference on the Theory and Application of Diagrams – Diagrams 2010 – was held in Portland, USA in August 2010.

Diagrams is an international and interdisciplinary conference series, which continues to present the very best work in all aspects of research on the theory and application of diagrams. Some key questions that researchers are tackling concern gaining an insight into how diagrams are used, how they are represented, which types are available and when it is appropriate to use them. The use of diagrammatic notations is studied for a variety of purposes including communication, cognition, creative thought, computation and problem-solving. Clearly, this must be pursued as an interdisciplinary endeavor, and Diagrams is the only conference series that provides such a united forum for all areas that are concerned with the study of diagrams: for example, architecture, artificial intelligence, cartography, cognitive science, computer science, education, graphic design, history of science, human–computer interaction, linguistics, logic, mathematics, philosophy, psychology, and software modelling. The articles in this volume reflect this variety and interdisciplinarity of the field.

Diagrams 2010 solicited long papers, short papers and posters. We received 67 submissions in all categories, and accepted 14 long papers – 34% acceptance rate, 8 short papers – 33% acceptance rate, and 29 posters – 75% acceptance rate. We deliberately accepted a substantial number of high-quality poster papers from a wide range of fields to contribute to the unique breadth of the Diagrams conference series. Every submission was reviewed by at least three members of the Program Committee, or additional referees who are experts in the relevant topics.

In addition to the paper presentations, we had the privilege to listen to two invited lectures: by Randall Davis on "Understanding Diagrams, and More: The Computer's View," and by Thomas Barkowsky on "Diagrams in the Mind: Visual or Spatial?". The program was strengthened by an interesting, diverse and lively poster session, and two excellent tutorials. For the first time in the history of Diagrams, we organized a Graduate Students Symposium — the aim was to provide young researchers with a forum for the exchange of ideas, networking and feedback on their work from a panel of experienced senior researchers.

Diagrams 2010 was co-located with the 32nd Annual Meeting of the Cognitive Science Society (Cogsci 2010). This co-location provided a lively and stimulating environment, enabling researchers from related communities to exchange ideas and more widely disseminate research results.

We owe thanks to a large number of people for making Diagrams 2010 such a great success. First, we are grateful to the 38 distinguished researchers who served on the Program Committee for their hard work and diligence. We were extremely impressed with the quality of reviews, and in particular of the discussion amongst the referees regarding individual papers with differing reviews. The Easychair conference system facilitated this discussion seamlessly; it also assisted and eased the compilation of these proceedings. We also thank Jim Davies for serving as the Graduate Student Symposium Chair, Stephanie Elzer for serving as the Tutorial Chair, and Unmesh Kurup for organizing special sessions. The conference website was kindly managed by Aidan Delaney, who also designed the conference advertising poster. Jill Russek generously provided assistance in compiling these proceedings. We thank Gem Stapleton for offering much useful advice.

Finally, we acknowledge the generous and substantial financial support from the US National Science Foundation (NSF), which enabled us to provide financial assistance to all PhD students who needed it to attend Diagrams 2010. We also thank Georgia Tech GVU Center for their financial help, and the Cognitive Science Society for partial support of Diagrams 2010 in the form of a Best Student Paper Award. We also acknowledge the help provided by Auburn University with the registration website, and Georgia Tech with the disbursement of student stipends.

May 2010

Mateja Jamnik
Ashok Goel
Hari Narayanan

# Conference Organization

## Conference Chairs

| | |
|---|---|
| Ashok Goel | Georgia Institute of Technology, USA |
| Mateja Jamnik | Cambridge University, UK |
| Hari Narayanan | Auburn University, USA |

## Graduate Symposium Chair

| | |
|---|---|
| Jim Davies | Carleton University, Canada |

## Tutorial Chair

| | |
|---|---|
| Stephanie Elzer | Millersville University, USA |

## Special Session Chair

| | |
|---|---|
| Unmesh Kurup | Rensselaer Polytechnic Institute, USA |

## Website and Technical Support

| | |
|---|---|
| Aidan Delaney | University of Brighton, UK |

## Program Committee

| | |
|---|---|
| Gerard Allwein | Naval Research Laboratory, USA |
| Christine Alvarado | Harvey Mudd College, USA |
| Michael Anderson | University of Hartford, USA |
| Dave Barker-Plummer | Stanford University, USA |
| Alan Blackwell | Cambridge University, UK |
| Dorothea Blostein | Queen's University, Canada |
| Paolo Bottoni | University of Rome, Italy |
| B. Chandrasekaran | Ohio State University, USA |
| Peter Cheng | University of Sussex, UK |
| Phil Cox | Dalhousie University, Canada |
| Richard Cox | University of Sussex, UK |
| Frithjof Dau | University of Wollongong, Australia |
| Max J. Egenhofer | University of Maine, USA |
| Jacques Fleuriot | University of Edinburgh, UK |
| Jean Flower | Autodesk, UK |

| | |
|---|---|
| John Gero | George Mason University, USA |
| Mark D. Gross | Carnegie Mellon University, USA |
| Corin Gurr | University of Reading, UK |
| Mary Hegarty | University of California, Santa Barbara, USA |
| John Howse | University of Brighton, UK |
| Hans Kestler | University of Ulm, Germany |
| John Lee | University of Edinburgh, UK |
| Richard Lowe | Curtin University of Technology, Australia |
| Kim Marriott | Monash University, Australia |
| Bernd Meyer | Monash University, Australia |
| Nathaniel Miller | University of Northern Colorado, USA |
| Mark Minas | Universitaet der Bundeswehr, Germany |
| Nancy Nersessian | Georgia Institute of Technology, USA |
| Luis Pineda | Universidad Nacional Autunoma de Mexico, Mexico |
| Helen Purchase | Glasgow University, UK |
| Peter Rodgers | University of Kent, UK |
| Frank Ruskey | University of Victoria, Canada |
| Atsushi Shimojima | Doshisha University, Japan |
| Sun-Joo Shin | Yale University, USA |
| Gem Stapleton | University of Brighton, UK |
| Nik Swoboda | Universidad Politecnica de Madrid, Spain |
| Susan Trickett | Naval Research Laboratory, USA |
| Barbara Tversky | Stanford University, USA |

## External Reviewers

Peter Chapman
Sanjay Chandrasekharan
Aidan Delaney
Chris Dorr
Alejandro Erickson
Andrew Fish
Ali Hamie
Yohei Kurata
Sonja Maier
Khalegh Mamakani
Nancy Nersessian
Yeonjoo Oh
Petros Papapanagiotou
Sebastian Roennau
Phil Scott
John Taylor
Matej Urbas
Aaron Williams
Sean Wilson
Michael Wybrow

# Table of Contents

## Invited Talks



## Tutorials



## Graduate Student Symposium



## Euler and Venn Diagrams



## Formal Aspects of Diagrams

## Reasoning with Diagrams



## Interacting with Diagrams



## Constructing Diagrams



## Understanding Diagrams and Text

## Posters

## Erratum

# Diagrams in the Mind: Visual or Spatial?

Thomas Barkowsky

University of Bremen, SFB/TR 8 Spatial Cognition,
Enrique-Schmidt-Str. 5, 28359 Bremen, Germany
barkowsky@sfbtr8.uni-bremen.de

Diagrams are known to be powerful forms of knowledge representation both for spatial and non-spatial problems. For human reasoners their power results from their immediate accessibility through visual perception. This talk focuses on the mental side of diagrammatic representations. People coping with visual or spatial tasks use *spatio-analogical* mental representations. These representations are – like external diagrams – characterized by their structural correspondence with the state of affairs they represent. Usually, two forms of mental spatio-analogical representations are distinguished: *spatial mental models* are abstract representations that only focus on specific structural aspects of spatial relationships, whereas *visual mental images* are much more complete and more detailed representations.

However, what are the very differences between spatial mental models and visual mental images? Which conditions determine whether one form or the other is employed in a given reasoning tasks? What are the implications for the problem to be solved and how can the form of mental representation that is used be determined?

In general, for reasons of cognitive economy, representation structures that are easy to build up and easy to maintain are preferred over more complex ones. Thus, people prefer mental models over mental images, whenever the former are sufficient for dealing with a given task. In particular, humans are capable of successively integrating pieces of spatial knowledge into an existing mental representation making it more and more complex. This capability, which currently no technical AI system can satisfactorily simulate, may in a step-by-step manner render a spatial mental model into a mental image. Therefore, it is proposed to view mental models and mental images as the extremes of a continuous dimension rather than as two distinct classes of representations.

On the other hand, in reasoning about a given problem both spatial mental models and visual mental images can be induced in humans depending on how the task is communicated to them. Operating on visual mental images results in eye movements that resemble perceiving a real visual stimulus whereas operating on spatial mental models yields no correspondence of the structure of the mental representation and the eye gaze patterns. Thus, the eye movements of a human reasoner indicate whether mental models or mental images are involved in the reasoning process and also which aspects are in the focus when mental images are involved.

As a consequence, understanding and adopting the characteristics of mental representations on the one hand may help build more powerful and more flexible AI systems. On the other hand, a deeper understanding of mental spatial reasoning enables the design of intelligent interactive human-machine reasoning systems in which both partners efficiently work together in performing on a given task.

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, p. 1, 2010.

# Understanding Diagrams, and More: The Computer's View

Randall Davis

MIT Computer Science and Artificial Intelligence Laboratory

Diagrams are an essential mode of thinking and communication. From the proverbial napkin sketch to office whiteboards to formal CAD drawings, we think and collaborate around sketches.

My group and I at MIT have been working for a decade to enable computers to join the conversation. In this talk I explore what it has taken to get computers to understand sketches and consider what it means for them to understand. In reviewing systems we have built and domains we have explored, I describe several themes that have emerged from our work. First, we have found that natural interaction is knowledge-based. Our ease of interacting with one another, for example, arises in large measure from a substantial body of shared knowledge. What is manifestly true of people appears to be true for computers as well. Second, there is considerable power in using multiple representations. Viewing sketches from the perspective of their spatial, temporal, and conceptual representations, and the interconnections among those representations, offers an effective means of dealing with challenges like noise and ambiguity. Third, diagrams are not enough. Think of all the whiteboard conversations you've had, and consider what got written on the whiteboard. Never mind that things get erased; even with multiple snapshots or a continuous record, what could you understand if all you had afterward was the visual record? This leads to the last theme: putting diagrams in the larger content of human communication as it routinely happens, i.e., communication that is multimodal, conversational, symmetric, and mixed initiative. I explore all of these issues and illustrate them by describing a variety of systems we have constructed.

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, p. 2, 2010.

# Diagrams: A Perspective from Logic

Dave Barker-Plummer

CSLI, Stanford University
dbp@stanford.edu

## 1 Tutorial Overview

The major goals of this two-hour tutorial are to to give an overview of this formal perspective on diagrams, and to introduce and explain the techniques used by logicians to analyze reasoning with diagrammatic representations.

The tutorial will describe the questions asked by logicians when analyzing the properties of diagrammatic representations and the techniques used to reason with them. The three main questions concern: *expressive completeness* — the degree to which diagrams can be used to represent information about a given domain; *soundness of inference* — how we can guarantee the validity of conclusions reached by reasoning with diagrams, and *completeness of inference* — the range of conclusions that can be reached by using those techniques.

The tutorial will proceed by first recapitulating the standard sentence-based approach to modelling reasoning, outline how these techniques can be applied to diagrammatic system, and then give a detailed presentation of the Hyperproof reasoning system. Hyperproof is a formal reasoning system, implemented as a computer application, for *heterogeneous* reasoning with diagrams and sentences. The Hyperproof system demonstrates that techniques from logic can be used to model very naturally a large class of everyday reasoning problems.

While Hyperproof will form the unifying frame of the tutorial, work by many colleagues in the field of diagrammatic reasoning will be discussed. These will be used to bring out both differences and similarities between approaches within the field. Hyperproof has been chosen since the diagrams that the program uses are related in an intuitive way to the situations that the diagrams represent. The Hyperproof system has this in common with everyday diagrammatic representations such as maps and schematics of machinery, for example. The demonstrated techniques are therefore more accessible when presented using Hyperproof than within other diagrammatic systems used to represent more abstract domains.

The tutorial will close with a discussion of a range of formal diagrammatic representations, outlining the ways in which more abstract representations are similar to Hyperproof, more appropriate for their own domains and often more straightforward to analyze formally. We will draw examples *inter alia* from mathematics, in the realm of set theory using Venn diagrams, and puzzle diagrams such as sudoku.

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, p. 3, 2010.

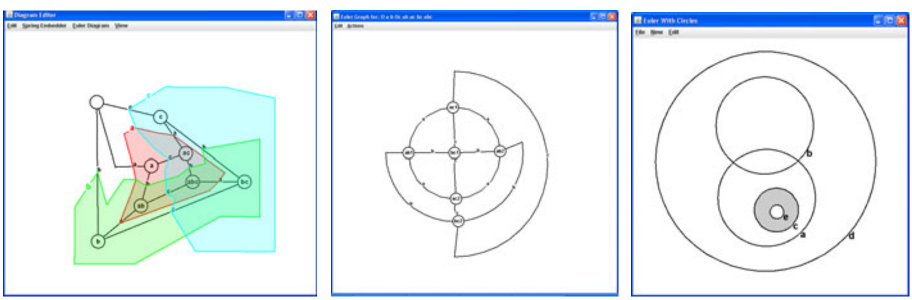
# Drawing Euler Diagrams for Information Visualization

John Howse[1], Peter Rodgers[2], and Gem Stapleton[1]

[1] Visual Modelling Group, University of Brighton, UK
{g.e.stapleton,john.howse}@brighton.ac.uk
[2] University of Kent, Canterbury, UK
p.j.rodgers@kent.ac.uk

Euler diagrams have numerous applications in information visualization as well as logical reasoning. They are typically used for displaying relationships between sets, such as whether one set is a subset of another. In addition, they can represent information about the relative cardinalities of the visualized sets by making the areas of the regions in the diagrams proportional to the set cardinalities. Using visualizations can allow the user to readily make interpretations that are not immediately apparent from the raw data set.

As with other diagram types, the ability to automatically produce an Euler diagram from the raw data would be advantageous. Indeed, the automated generation and layout of diagrams can play a key role in the usability of visual languages. There have been a number of techniques devised for the automated drawing of Euler diagrams, each of which has its own advantages and disadvantages. The techniques can be broadly classified into three categories: dual graph based methods, inductive methods, and methods that use particular geometric shapes. The respective three categories give rise to the diagrams shown below:



Participants will be presented with an overview of different Euler diagram drawing methods, including their strengths and weaknesses. Freely available software tools to support their use will be demonstrated. The tutorial will also discuss their use in information visualization, highlighting a range of areas in which they are helpful. Thus, the tutorial should make attendees more aware of the scope for Euler diagram application and the state-of-the-art tools available for their automated generation.

**Acknowledgements.** Supported by EPSRC grants EP/E011160/1, EP/E010393/1, EP/H012311/1 and EP/H048480/1.

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, p. 4, 2010.

# The Graduate Student Symposium of Diagrams 2010

Jim Davies

Institute of Cognitive Science, Carleton University
1125 Colonel By Drive, Ottawa, Ontario, Canada K1S5B6
jim@jimdavies.org

The Graduate Student Symposium (GSS) is a forum at which Ph.D. students engaged in diagrams research have an opportunity to present their research and interact with established scientists as well as other students.

The GSS has two main goals. First, the GSS provides an environment that is supportive of constructive feedback. Second, it is an opportunity for the students to shine, take the focus, and network with their peers and more senior scholars.

Several papers were submitted only to the GSS. There were papers based on formal analyses (Burton, Howse, Stapleton, & Hamie; Delaney) modeling human understanding (Dickmann; Smuc) and visual language creation for architectural design (Hamadah).

The other presentations were based on students' papers presented in the Diagrams 2010 poster session. Nearly all of these papers involved human experimentation or modeling. Several papers dealt specifically with diagrams in education, others on basic understanding of diagrams, and others on diagram generation. One paper presented work that developed a visual language for mathematical proof.

In addition, two invited speakers gave presentations, one on how to present a paper at a conference, and another with dissertation advice.

A bursary for student attendees to the GSS (totaling $20,000 USD) was generously provided by the National Science Foundation. All students who applied for funding received a bursary to help them pay for attending the conference.

In summary, the Graduate Student Symposium of Diagrams 2010 provides a unique opportunity for feedback to students involved with diagrams research. Further, the topics students choose gives us insight into the future of our field.

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, p. 5, 2010.

# The Efficacy of Euler and Venn Diagrams in Deductive Reasoning: Empirical Findings

Yuri Sato, Koji Mineshima, and Ryo Takemura

Department of Philosophy, Keio University,
2-15-45 Mita, Minato-ku, Tokyo 108-8345, Japan
{sato,minesima,takemura}@abelard.flet.keio.ac.jp

**Abstract.** This paper discusses the cognitive differences between reasoning with Euler diagrams and reasoning with Venn diagrams. We test subjects' performances in syllogism solving in case where these two types of diagrams are used. We conduct an analysis on the role played by the conventional devices of each diagram in reasoning processes. Based on this, we hypothesize that of the two types of diagrams, only Euler diagrams could guide subjects without prior knowledge of their inferential strategies for combining diagrams. To test this hypothesis, subjects in our experiment are only provided with instructions on the meanings of diagrams and required to solve reasoning tasks without any instruction on the solving strategies. Our experimental results support the hypothesis and indicate that Euler diagrams can not only contribute to subjects' correct interpretation of categorical sentences used but also play a crucial role in reasoning processes themselves.

## 1 Introduction

This paper investigates what type of diagrammatic representation is effective in human reasoning and what makes a diagrammatic representation particularly useful for solving a deductive reasoning task. We focus on the use of Euler and Venn diagrams in syllogistic reasoning. These two types of logic diagrams have been intensively studied in formal diagrammatic logic since the 1990s (e.g. Shin, 1994; Hammer, 1995; Howse, Stapleton & Taylor, 2005; for a recent survey, see Stapleton, 2005); however, currently, few empirical researches investigate their cognitive foundations. For example, it is often observed that Venn diagrams are visually less clear and hence harder to handle in actual reasoning than Euler diagrams (for discussions, see e.g., chapter V of Venn, 1881; Hammer & Shin, 1998). However, the cognitive underpinning of such a claim has been seldom investigated experimentally. Thus, it seems fair to say that a serious gap exists between theoretical and empirical researches in this area.

We compare subjects' performances in syllogism solving in cases where diagrammatic representations (Euler diagrams and Venn diagrams) are used and cases where they are not used (i.e., only linguistic/sentential materials are used). Euler and Venn diagrams crucially differ with respect to the conventional devices employed in their representation systems. Both Euler and Venn diagrams

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 6–22, 2010.

used in our experiment adopt a convention of *crossing*, according to which two circles that are indeterminate with respect to their semantic relation are put to partially overlap each other. An important difference is that while the correct manipulation of Euler diagrams exploits the intuitive understanding of topological relations between circles, that of Venn diagrams essentially depends on the understanding of another conventional device, that is, *shading*, and its interaction with partially overlapping circles. In Section 2, we introduce the Euler and Venn representation systems used in our experiment, and explain the role of the conventional devices of each system in more detail.

In Section 3, in view of the role played by the conventional devices of each system in reasoning processes, we hypothesize that of the two types of diagrams, only Euler diagrams can guide subjects without prior knowledge of their inferential strategies for combining diagrams. What we mean by "guide" here is that diagrams not only contribute to a subject's correct representation of the given information but also play a crucial role in reasoning processes themselves. More specifically, the solving processes of reasoning tasks can be replaced by the syntactic manipulation of diagrammatic representations or, in other words, the constructions of diagrammatic proofs. Our hypothesis is that Euler diagrams are *self-guiding* in the sense that the construction of diagrammatic proofs could be automatically triggered even for subjects without explicit prior knowledge of inferential strategies or rules, whereas Venn diagrams are not.

In order to test this hypothesis, subjects in our experiment are only provided with instructions on the meanings of diagrams and are required to solve reasoning tasks without any instruction on how to manipulate diagrams in reasoning processes. In this respect, our set-up differs from the one in the studies of logic teaching methods used in Stenning (1999) and Dobson (1999), where subjects are taught both the meanings and the ways of manipulation of diagrams. Our research should be viewed as a study on the efficacy of external representations in deductive reasoning (e.g., Bauer & Johnson-Laird, 1993, Scaife & Rogers, 1996).

Based on our hypothesis, we predict that the performance of the Euler diagram group would be better than those of the Venn diagram and linguistic groups. In Section 4, we present the details of our experiment and show the results, which confirm our prediction. We also discuss the effects of Euler and Venn diagrams to block some well-known errors in linguistic syllogistic reasoning. In Section 5, we conclude the paper and discuss future research directions.

## 2 Background: Conventional Devices in Euler and Venn Representation Systems

As emphasized in Stenning and Oberlander (1995), diagrams that are beneficial as a tool in deductive reasoning must satisfy two requirements. On one hand, the diagrams used should be simple and concrete enough to express information with their natural and intuitive properties such as geometrical or topological properties. On the other hand, the diagrams used should have some abstraction to deal with the partial information arising in reasoning processes. The dilemma here is that in order to manipulate diagrams with abstraction correctly, users need to learn some arbitrary representational conventions governing the abstraction; however the existence of such conventions often clashes with the first requirement, that is, the naturalness of diagrammatic representations. Accordingly, if diagrams are beneficial in syllogistic reasoning, they must have enough abstraction as well as some natural properties exploitable in the reasoning tasks. An Euler diagrammatic representation system introduced in our previous work (Mineshima, Okada, Sato, and Takemura, 2008), called the EUL system, can serve as such a system.[1] Hence, of the various representation systems based on Euler diagrams, we use the EUL system in our experiment. Here let us briefly explain the EUL system in comparison with another type of Euler diagrammatic representation systems (Gergonne's system) and Venn representation system.

**Gergonne's system.** There is a particular version of Euler representation systems, called Gergonne's representation system (Gergonne, 1817), which is well known in the context of the psychological studies of syllogistic reasoning (e.g., Erickson, 1974).[2] This system has the following features: (i) a diagram consists only of circles; (ii) every minimal region in a diagram represents a non-empty set. As a consequence, in order to represent a single categorical statement in syllogism, one has to use more than one diagram. For example, the statement "Some $A$ are not $B$" is represented by the disjunction of the three diagrams shown in Fig. 1. Similarly, "Some $B$ are $C$" requires four diagrams. This means that when we have a syllogism with these two premises, we have to consider 12 ways of combining diagrams corresponding to the premises (cf. chapter 4 of Johnson-Laird, 1983). Thus, although the diagrams in Gergonne's system have visual clarity in that they solely rest on the topological relationships between circles, they cannot represent partial information (which is essential to the treatment of syllogistic reasoning) in a single diagram, and hence, they are difficult to handle in actual deductive reasoning. We can say that Gergonne's system is natural enough but lacks a conventional mechanism to deal with abstraction.

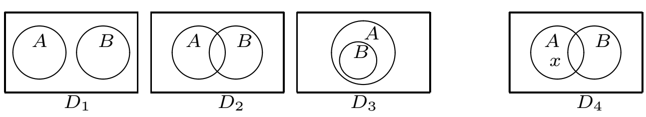



**Fig. 1.** The diagrams corresponding to "Some $A$ are not $B$" in Gergonne's representation system ($D_1$, $D_2$, and $D_3$) and in our EUL representation system ($D_4$)

[1] A formal semantics and a diagrammatic inference system are provided for it. Its completeness is shown in Mineshima, Okada, and Takemura (2009).

[2] The Euler diagrams used in the experiments of Rizzo and Palmonari (2005) and Calvillo, DeLeeuw, and Revlin (2006), where negative results on the efficacy of Euler diagrams were shown, can also be considered to be based on this version of Euler diagrammatic representation system. The Euler diagrams as currently studied in diagrammatic logic are usually the ones based on what we call the convention of crossing, rather than the ones in Gergonne's system (cf. Stapleton, 2005).

**Venn system.** As is well known, Venn (1881) and Peirce (1897) attempted to overcome the difficulty of Gergonne's version of Euler diagrams by removing the existential import from regions. Venn first fixes a so-called "primary diagrams" such as $D_2$ in Fig. 1, where every circle partially overlaps each other. Such diagrams do not convey any specific semantic information. We say that Venn diagrams are subject to the convention of *crossing*, according to which two circles which are indeterminate with respect to their semantic relation are put to partially overlap each other. In Venn diagrams, then, meaningful relations among circles are expressed using a novel device, *shading*, by the stipulation that shaded regions denote empty sets. For example, the statement "All $A$ are $B$" is represented as $D_v$ in Fig. 2 below, which reads as *There is nothing which is A but not B*. In this way, logical relations among terms are represented not simply by topological relations between circles, but by the essential use of shading. Thus, although Venn diagrams are expressive enough (cf. Shin, 1994), some complications are involved in the treatment of syllogistic reasoning. Here we see that the expressiveness of Venn diagrams is obtained at the cost of naturalness.

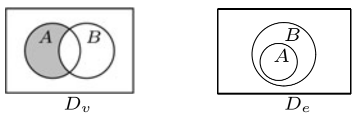



**Fig. 2.** Representations of *All A are B* in Venn diagram ($D_v$) and Euler diagram ($D_e$)

**EUL system.** The EUL system is a simple representation system for Euler diagrams.[3] In contrast to Gergonne's system, the EUL system has the following features: (i) it uses a named *point* "$x$" to indicate the existence of objects; (ii) as in Venn diagrams, it adopts the convention of crossing. Consequently, in the EUL system, a single categorical statement can be represented by just a single diagram. For example, the statement "Some $A$ are not $B$" can be expressed by $D_4$ in Fig. 1. Furthermore, in contrast to Venn diagrams, the EUL system represents categorical statements in terms of the topological relations between circles, that is, inclusion and exclusion relations, without using a conventional device such as shading. It is known that syllogistic reasoning can be characterized just in terms of an inference system based on EUL (cf. Mineshima et al., 2008).

In sum, the EUL system is distinctive in that it avoids the combinatorial complexities inherent in Gergonne's system, and in addition that it dispenses with a new conventional device to express negation, such as shading in Venn diagrams. One common feature of the Venn system and the EUL system is that both rely on the convention of crossing. In what follows, we refer to diagrams in the EUL system simply as Euler diagrams.

[3] By "Euler" diagrams, we mean the diagrams based on topological relations, such as inclusion and exclusion relations, between circles. Thus, both diagrams in Gergonne's system and those in our EUL system are instances of Euler diagrams, whereas Venn diagrams are not.

## 3 Hypothesis on the Efficacy of Euler and Venn Diagrams

The efficacy of diagrams in problem solving has been specified in various ways, in comparison to linguistic representations (e.g., Larkin & Simon 1987). In particular, Shimojima (1996a,b) proposed the notion of *free ride* to account for the efficacy of diagrams in reasoning processes, and it has been influential in the literature.[4] In this section, we consider what consequence it has for the differences between reasoning with Euler diagrams and Venn diagrams. We then present our own hypothesis regarding the inferential efficacy of these two types of diagrams.

As an illustration, let us consider an example of syllogism: *All A are B, No C are B; therefore No C are A*. Fig. 3 shows a solving process of this syllogism using Euler diagrams. Here, the premise *All A are B* is represented by $D_1^e$ and the premise *No C are B* is represented by $D_2^e$. By unifying $D_1^e$ with $D_2^e$, one can obtain diagram $D_3^e$. In this diagram, the exclusion relation holds between circles $A$ and $C$, which corresponds to the valid conclusion *No C are A*. The point here is that by unifying the two premise diagrams, one can *automatically read off* the semantic relation between the circles $A$ and $C$ without any additional operation. Shimojima (1996a,b) calls this the *free ride* property and shows that such an advantage can be seen to exists in other types of diagram use in reasoning and problem solving.

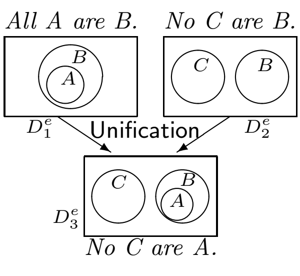



**Fig. 3.** A diagrammatic proof of syllogism *All A are B, No C are B; therefore No C are A* with Euler diagrams

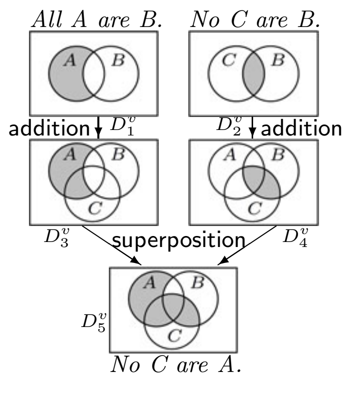



**Fig. 4.** A derivation of syllogism *All A are B, No C are B; therefore No C are A* with Venn diagrams

Fig. 4 illustrates a solving process of the same syllogism as Fig.3 using Venn diagrams. Here, the premise *All A are B* is represented by $D_1^v$, and the premise *No C are B* is represented by $D_2^v$. Venn diagrams have a fixed configuration of circles, and shading plays an essential role in the process of combining the information contained in premises. In the solving process in Fig. 4, circle $C$ is first added with diagrams $D_1^v$ and $D_2^v$ to obtain $D_3^v$ and $D_4^v$, respectively. Next,

[4] See, for example, Gurr, Lee, and Stenning (1998). For earlier related proposals, see, in particular, Sloman (1971) and Barwise and Etchemendy (1991).

by superposing $D^v_3$ with $D^v_4$, one obtains diagram $D^v_5$, from which the conclusion "No $C$ are $A$" can correctly be read off. If this derivation is available to users, they can exploit the free ride property of the reasoning processes in Venn diagrams.

What plays a crucial role in providing free rides in these derivations is the processes of unification and superposition. Generally speaking, deductive reasoning requires tasks to combine information contained in given premises. Such a task of combining information could naturally be replaced by the processes of unification and superposition. In view of the free ride property, then, there seems to be no essential difference in the efficacy between reasoning with Euler diagrams and reasoning with Venn diagrams. Intuitively, however, Venn diagrams seem to be relatively more difficult to manipulate in solving syllogism than Euler diagrams. For users need to add new circles before combining the two premise diagrams by superposition. And such a process of adding a new circle seems to be hard to access and apply, unless users have the prior knowledge of the relevant inference strategy of solving syllogism using Venn diagrams. Those who are ignorant of such a strategy cannot appeal to concrete manipulations of the diagrams, and hence, they seem to draw a conclusion solely based on semantic information which can be obtained from given sentential materials and Venn diagrams.

On the other hand, Euler diagrams seem to be relatively easy to handle for even those users who are not trained to manipulate them in syllogism solving. Note that the essential step in solving processes with Euler diagrams is the unification step, as exemplified in Fig. 3 above. By unifying two Euler diagrams, users could exploit the intuitive understanding of the natural properties of topological relations between circles, such as inclusion and exclusion relations. Here, we expect that users could extract the right strategies to draw a conclusion from Euler diagrams themselves.

Based on this contrast between Euler and Venn diagrams, we hypothesize that of the two types of diagrams, only Euler diagrams can guide subjects without prior knowledge of the strategies for combining diagrams. What we mean by "guide" here is that diagrams not only contribute to a subject's correct representation of given information but also play a crucial role in reasoning processes themselves. More specifically, reasoning processes can be replaced by the syntactic manipulations of diagrams or, in other words, the construction of *diagrammatic proofs*. We will say that diagrammatic representations are *self-guiding* if

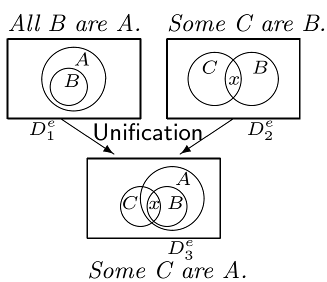



**Fig. 5.** A solving process using the convention of crossing

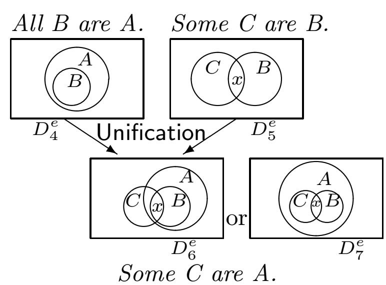



**Fig. 6.** A solving process by enumeration

the construction of diagrammatic proofs are automatically triggered even for subjects without explicit prior knowledge of inferential strategies. Our hypothesis then is that in syllogistic reasoning tasks, Euler diagrams are self-guiding in this sense, whereas Venn diagrams are not.

It should be noted here that the correct manipulation of Euler diagrams sometimes depends on the understanding of the convention of crossing. Fig. 5 indicates an example that requires partially overlapping circles to represent indeterminacy. In this derivation the relationship between circles $A$ and $C$ is not determined by the information contained in the premises. Thus, by the convention of crossing, the circles $A$ and $C$ are put to partially overlap each other in the unified diagram $D^e_3$, from which one can correctly read off the semantic information corresponding to the valid conclusion *Some C are A*. Note here that there is another possible solving process: one could enumerate the possible configurations of unified diagrams and then check whether the conclusion holds in each configuration. Fig. 6 indicates such a process. Although this process is anomalous in view of the convention of crossing, it could derive a correct conclusion. Here, one can read off the same semantic information "Some $C$ are $A$" from $D^e_6$ and $D^e_7$, which yields the correct conclusion.

A similar point applies to the case of syllogisms which have no valid conclusion. For instance, consider the syllogism having premises *All B are A* and *No C are B*.

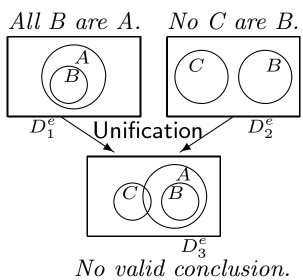



**Fig. 7.** A solving process using the convention of crossing

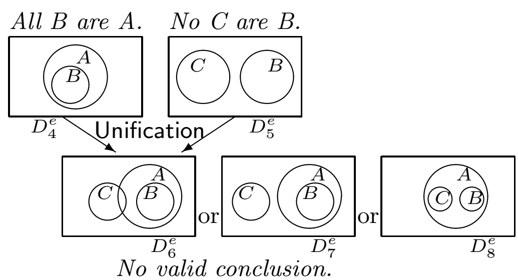



**Fig. 8.** A solving process by enumeration

Fig. 7 shows a solving process based on the convention of crossing. Here, the fact that circle $A$ partially overlaps circle $C$ in the unified diagram $D^e_3$ indicates that no specific semantic relation holds between $A$ and $C$. From this one can conclude that there is no valid conclusion in this syllogism. Fig. 8 indicates a solving process in which all possible configurations of unified diagrams are enumerated, an anomalous process in view of the convention of crossing. Here, there is no relationship between $A$ and $C$ that holds in all the three unified diagrams $D^e_6, D^e_7$, and $D^e_8$. From this, we can also conclude that this syllogism has no particular valid conclusion regarding $A$ and $C$.

These considerations show that, in the case of syllogistic reasoning with Euler diagrams, the manipulation of diagrams in reasoning processes does not essentially depend on the convention of crossing. In contrast, as is seen above, reasoning with Venn diagrams essentially involves processes that depend on the convention of crossing, in particular, the process of adding a new circle (see Fig. 4 above). Without such a process, the task of combining the information contained in given premises cannot be replaced by syntactic manipulations of diagrams. Thus, we can say that users' prior knowledge of the solving strategies would play a crucial role in manipulating Venn diagrams syntactically in reasoning processes.[5]

In the case of linguistic syllogistic reasoning, where subjects are not allowed to use any diagram, it is known that subjects often make some interpretational errors due to the word order, such as the subject-predicate distinction, of a sentential material (cf. Newstead & Griggs, 1983). It is expected that both Euler diagrams and Venn diagrams may help subjects avoid such interpretational errors in linguistic syllogistic reasoning (for discussions, see Stenning, 2002 and Mineshima et al. 2008). Hence, the performances in both Euler and Venn diagrammatic reasoning would be generally better than that in linguistic reasoning.

**Predictions.** Based on the above considerations, we predict that (1) the performance in syllogism solving would be better when subjects use Euler diagrams than when they use Venn diagrams and (2) the performance in syllogism solving would be better when subjects use Euler diagrams or Venn diagrams than when they use only sentential materials.

# 4 Experiment

## 4.1 Method

**Design.** In order to test our hypothesis, we provided the subjects in our experiment only with instructions on the meanings of diagrams and required them to solve reasoning tasks without any instruction on how to manipulate diagrams in syllogism solving. We first conducted a pretest to check whether subjects understood the instructions correctly. The pretest was designed mainly to see whether subjects correctly understand the conventional devices of each system, in particular, the convention of crossing in the Euler and Venn systems and shading in the Venn system. We then compared subjects' performances in syllogism solving in cases where diagrammatic representations (i.e. Euler diagrams and Venn diagrams) are used with the cases where they are not used.

**Participants.** Two hundred and thirty-six undergraduates (mean age 20.13 ± 2.99 SD) in five introductive philosophy classes participated in the experiment. They gave their consent to their cooperate in the experiment, and after the experiment, they were given a small non-monetary reward. The subjects were native speakers of Japanese, and the sentences and instructions were given in Japanese. The subjects were divided into three groups: the linguistic group, the

[5] Although this paper focuses on the processes of *combining* information, it is also interesting to investigate the cognitive properties of a process of *extracting* information from a diagram. Such a process might be regarded as *deletion* steps in diagrammatic reasoning (for some interesting discussion, see Gurr et al., 1998).

Euler group, and the Venn group. The linguistic group consisted of 66 students. Of them, we excluded 21 students: those who left the last more than three questions unanswered (19 students) and those who had participated in our pilot experiments conducted before (2 students). The Euler group consists of 68 students. Of them, we excluded 5 students: those who left the last more than three questions unanswered (3 students) and those who had participated in our pilot experiments conducted before (2 students). The Venn group consists of 102 students. Of them, we excluded 34 students: those who left the last more than three questions unanswered (27 students) and those who had participated in our pilot experiments conducted before (7 students).

**Materials.** The experiment was conducted in the booklet form.

**1) Pretest.** The subjects of the Euler group and Venn group were presented with 10 diagrams listed in Figs. 9 and 10, respectively.

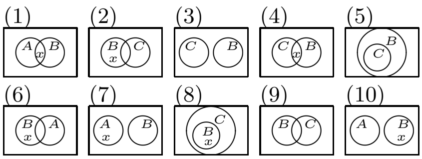


**Fig. 9.** Euler diagrams used in the pretest

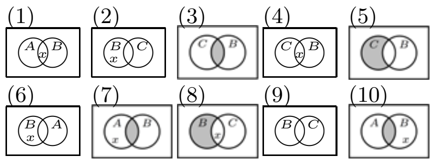


**Fig. 10.** Venn diagrams used in the pretest

The subjects were asked to choose, from a list of five possibilities, the sentences corresponding to a given diagram. Examples are given in Figs. 11 and 12. The answer possibilities were *All-*, *No-*, *Some-*, *Some-not*, and *None of them*. The subject-predicate order of an answer sentence was *AB* or *BC*. The total time given was five minutes. The correct answer to each diagram was: (1) "Some *A* are *B*," (2) "Some *B* are not *C*," (3) "No *B* are *C*," (4) "Some *B* are *C*," (5) "None of them," (6) "None of them," (7) "No *A* are *B*" and "Some *A* are not *B*," (8) "All *B* are *C*" and "Some *B* are *C*," (9) "None of them," (10) "No *A* are *B*," respectively. Before the pretest, the subjects were presented with the examples in Figs. 11 and 12.

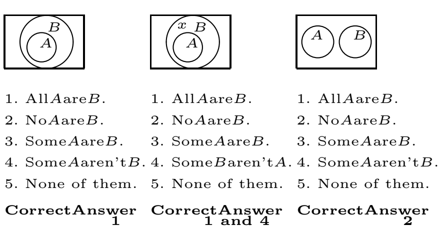



**Fig. 11.** The examples in the pretest of Euler diagrams

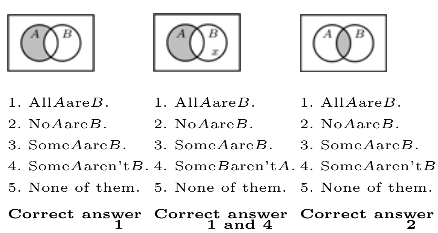
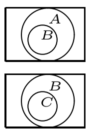



**Fig. 12.** The examples in the pretest of Venn diagrams

**2) Syllogistic reasoning tasks.** Subjects in the Euler group were given syllogisms with Euler diagrams (such as the one in Fig. 13). Subjects in the Venn group were given syllogisms with Venn diagrams (such as the one in Fig. 14), and subjects in linguistic group were given syllogisms without diagrams. We gave 31 syllogisms in total, out of which 14 syllogisms had a valid conclusion and 17 syllogisms had no valid conclusion. The subjects were presented with two premises and were asked to choose, from a list of five possibilities, a sentence corresponding to the valid conclusion. The list consists of *All-*, *No-*, *Some-*, *Some-not*, and *NoValid*. The subject-predicate order of each conclusion was $CA$. The test was a 20-minute power test, and each task was presented in random order (10 patterns were prepared). Before the test, the example in Fig. 13 was presented to subjects in the Euler group, and the one in Fig. 14 to subjects in the Venn group.

All $B$ are $A$.

All $C$ are $B$.



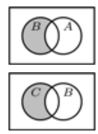

1. All $C$ are $A$.
2. No $C$ are $A$.
3. Some $C$ are $A$.
4. Some $C$ are not $A$.
5. None of them.

**Correct answer: 1**

**Fig. 13.** An example of syllogistic reasoning task of the Euler group

All $B$ are $A$.

All $C$ are $B$.




1. All $C$ are $A$.
2. No $C$ are $A$.
3. Some $C$ are $A$.
4. Some $C$ are not $A$.
5. None of them.

**Correct answer: 1**

**Fig. 14.** An example of syllogistic reasoning task of the Venn group

**Procedure.** All three groups were first given 1 minute 30 seconds to read one page of instructions on the meaning of categorical statements. In addition, the Euler group was given 2 minutes to read two pages of instructions on the meaning of Euler diagrams, and the Venn group was given 2 minutes to read two pages of instructions on the meaning of Venn diagrams. Before the pretest, the Euler and Venn groups were given 1 minute 30 seconds to read two pages of instructions on the pretest. Finally, before the syllogistic reasoning test, all three groups were given 1 minute 30 seconds to read two pages of instructions, in which the subjects were warned to choose only one sentence as answer and not to take a note.[6]

### 4.2 Results

**Pretest.** The accuracy rate of each item in the pretest of Euler diagrams, listed in Fig. 8 was (1) 77.8%, (2) 77.8%, (3) 90.5%, (4) 81.0%, (5) 69.8%, (6) 84.1%, (7) 49.2%, (8) 58.7%, (9) 79.4%, and (10) 82.5%, respectively. One major source of error was the misunderstanding of the convention of crossing; subjects tended to incorrectly select both *Some-* and *Some-not* in (1), (2), (4), (5), (6), and (9).

[6] For more details, see `http://abelard.flet.keio.ac.jp/person/sato/index.html`

This error was observed in 18 students (out of 63 students), who scored less than 8 on the pretest (out of 12). In the following analysis, we exclude these 18 students and refer to the other 45 students as the Euler group. The correlation coefficient between the scores of pretest and of syllogistic reasoning tasks in the total Euler group (N = 63) was substantially positive with 0.606.

The accuracy rate of each item in the pretest of Venn diagrams, listed in Fig. 9 was (1) 85.2%, (2) 75.0%, (3) 74.0%, (4) 82.0%, (5) 22.1%, (6) 66.0%, (7) 35.3%, (8) 35.3%, (9) 77.9%, and (10) 66.1%, respectively. One major source of error was also the misunderstanding of crossing (partially overlapping circles); subjects tended to incorrectly select both *Some-* and *Some-not* in (1), (2), (4), (5), (6), and (9). This error was observed in 38 students (out of 68 students) who scored less than 8 on the pretest (out of 12). In the following analysis, we exclude these 38 students and refer to the other 30 students as the Venn group. The correlation coefficient between the scores of pretest and of syllogistic reasoning tasks in the total Venn group (N = 68) was substantially positive with 0.565.

**Syllogistic reasoning tasks.** A one-way analysis of variance (ANOVA) was calculated on the accuracy rates of reasoning tasks in the three groups. The result was significant, $F$ (2, 117) = 52.515, $p < .001$. The average accuracy rates of the total 31 syllogistic reasoning tasks in the three groups are shown in Fig. 15. The accuracy rate of reasoning tasks in the Euler group was higher than that in the linguistic group: 46.7% for the linguistic group and 85.2% for the Euler group ($F$ (1, 88) = 10.247, $p < .001$. in multiple comparison tests by Ryan's procedure, between the linguistic group and the Euler group). The accuracy rate of reasoning tasks in the Euler group was higher than that in the Venn diagrammatic group: 66.5% for the Venn group and 85.2% for the Euler group ($F$ (1, 73) = 4.421, $p < .001$.). The accuracy rate of reasoning tasks in the Venn group was higher than that in the linguistic group: 66.5% for the Venn group and 46.7% for the linguistic group ($F$ (1, 73) = 4.744, $p < .001$).[7]

The results of each syllogistic type are shown in Table 1. Numbers indicate the percentage of total responses to each syllogism. Bold type refers to valid conclusion by the standard of predicate logic. For simplicity, we exclude the conclusions of the so-called "weak" syllogisms (syllogisms whose validity depends on the existential import of subject term) from valid answers.

A significant difference in performance was found in the linguistic group and the diagrammatic groups (Euler and Venn groups). This difference can be

[7] If we include those subjects who failed the pretest (i.e., those who scored less than 8), the results are as follows. The rate for the total Euler group (including those who failed the pretest) was 77.8% and that for the total Venn group (including those who failed the pretest) was 52.9%. Multiple comparison tests yield the following results: (i) There was significant difference between the linguistic group and the total Euler group, $F$ (1, 106) = 7.939, $p < .001$. (ii) There was significant difference between the total Euler group and the total Venn group, $F$ (1, 129) = 7.098, $p < .001$. (iii) There was no significant difference between the total Venn group and the linguistic group, $F$ (1, 112) = 1.604, $p = .10$.

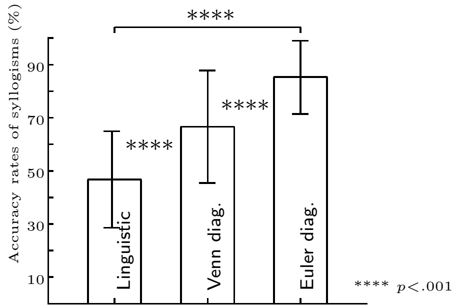



**Fig. 15.** The average accuracy rates of 31 total syllogisms in the linguistic group, the Venn group, and the Euler group (error-bar refers to SD)

**Table 1.** Response distributions for 31 syllogisms in the Linguistic group, Venn group and Euler group (Bold type refers to valid conclusion)

| | | Linguistic group, N=45 | | | | | Venn group, N=30 | | | | | Euler group, N=45 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *code & figure* | *premises 1st, 2nd* | *conclusion** | | | | | *conclusion** | | | | | *conclusion** | | | | |
| | | A | E | I | O | N | A | E | I | O | N | A | E | I | O | N |
| AA2 | $all(A,B); all(C,B)$ | 55.6 | 6.7 | 4.4 | 2.2 | **31.1** | 23.3 | 0.0 | 3.3 | 0.0 | **73.3** | 4.4 | 0.0 | 0.0 | 0.0 | **95.6** |
| AA3 | $all(B,A); all(B,C)$ | 60.0 | 0.0 | 26.7 | 0.0 | **13.3** | 16.6 | 0.0 | 16.6 | 0.0 | **63.3** | 0.0 | 0.0 | 13.3 | 0.0 | **86.7** |
| AA4 | $all(A,B); all(B,C)$ | 60.0 | 0.0 | 28.9 | 0.0 | **11.1** | 23.3 | 3.3 | 16.6 | 3.3 | **53.3** | 8.9 | 0.0 | 11.1 | 2.2 | **77.8** |
| AI1 | $all(B,A); some(C,B)$ | 2.2 | 2.2 | **88.9** | 2.2 | 4.4 | 0.0 | 0.0 | **93.3** | 0.0 | 6.6 | 0.0 | 0.0 | **100.0** | 0.0 | 0.0 |
| AI2 | $all(A,B); some(C,B)$ | 0.0 | 0.0 | 55.6 | 17.8 | **26.7** | 0.0 | 3.3 | 30.0 | 0.0 | **66.6** | 0.0 | 0.0 | 6.7 | 2.2 | **88.9** |
| AI3 | $all(B,A); some(B,C)$ | 4.4 | 2.2 | **80.0** | 6.7 | 6.7 | 0.0 | 0.0 | **80.0** | 6.6 | 13.3 | 0.0 | 0.0 | **84.4** | 0.0 | 15.6 |
| AI4 | $all(A,B); some(B,C)$ | 4.4 | 0.0 | 57.8 | 6.7 | **31.1** | 0.0 | 0.0 | 33.3 | 0.0 | **66.7** | 0.0 | 0.0 | 15.6 | 0.0 | **84.4** |
| IA1 | $some(B,A); all(C,B)$ | 2.2 | 2.2 | 60.0 | 8.9 | **26.7** | 3.3 | 0.0 | 33.3 | 0.0 | **63.3** | 0.0 | 0.0 | 8.9 | 2.2 | **84.4** |
| IA2 | $some(A,B); all(C,B)$ | 6.7 | 0.0 | 51.1 | 11.1 | **31.1** | 3.3 | 3.3 | 23.3 | 0.0 | **70.0** | 0.0 | 0.0 | 13.3 | 2.2 | **84.4** |
| IA3 | $some(B,A); all(B,C)$ | 0.0 | 2.2 | **93.3** | 0.0 | 4.4 | 0.0 | 0.0 | **66.7** | 0.0 | 33.3 | 0.0 | 0.0 | **75.6** | 0.0 | 24.4 |
| IA4 | $some(A,B); all(B,C)$ | 11.1 | 0.0 | **73.3** | 6.7 | 6.7 | 0.0 | 3.3 | **50.0** | 3.3 | 43.3 | 0.0 | 0.0 | **68.9** | 0.0 | 31.1 |
| AE1 | $all(B,A); no(C,B)$ | 0.0 | 64.4 | 0.0 | 6.7 | **26.7** | 0.0 | 33.3 | 3.3 | 3.3 | **60.0** | 0.0 | 6.7 | 0.0 | 4.4 | **88.9** |
| AE2 | $all(A,B); no(C,B)$ | 2.2 | **93.3** | 0.0 | 2.2 | 2.2 | 0.0 | **86.7** | 0.0 | 6.6 | 6.6 | 0.0 | **97.8** | 0.0 | 0.0 | 2.2 |
| AE3 | $all(B,A); no(B,C)$ | 0.0 | 64.4 | 2.2 | 11.1 | **20.0** | 3.3 | 36.6 | 0.0 | 10.0 | **46.6** | 0.0 | 15.6 | 0.0 | 4.4 | **80.0** |
| AE4 | $all(A,B); no(B,C)$ | 0.0 | **77.8** | 4.4 | 6.7 | 11.1 | 0.0 | **80.0** | 0.0 | 6.6 | 10.0 | 0.0 | **95.6** | 0.0 | 2.2 | 2.2 |
| EA1 | $no(B,A); all(C,B)$ | 0.0 | **91.1** | 0.0 | 2.2 | 2.2 | 0.0 | **96.6** | 0.0 | 0.0 | 3.3 | 0.0 | **97.8** | 0.0 | 0.0 | 2.2 |
| EA2 | $no(A,B); all(C,B)$ | 2.2 | **88.9** | 0.0 | 4.4 | 4.4 | 3.3 | **90.0** | 3.3 | 3.3 | 0.0 | 0.0 | **100.0** | 0.0 | 0.0 | 0.0 |
| EA3 | $no(B,A); all(B,C)$ | 0.0 | 62.2 | 0.0 | 20.0 | **17.8** | 0.0 | 53.3 | 0.0 | 10.0 | **36.6** | 0.0 | 11.1 | 0.0 | 4.4 | **84.4** |
| EA4 | $no(A,B); all(B,C)$ | 2.2 | 60.0 | 2.2 | 13.3 | **17.8** | 0.0 | 43.3 | 3.3 | 10.0 | **36.6** | 0.0 | 6.7 | 0.0 | 11.1 | **82.2** |
| AO1 | $all(B,A); some\text{-}not(C,B)$ | 0.0 | 4.4 | 8.9 | 66.7 | **17.8** | 0.0 | 0.0 | 10.0 | 60.0 | **30.0** | 0.0 | 4.4 | 0.0 | 20.0 | **75.6** |
| AO2 | $all(A,B); some\text{-}not(C,B)$ | 0.0 | 4.4 | 4.4 | **75.6** | 15.6 | 0.0 | 3.3 | 6.6 | **66.6** | 23.3 | 0.0 | 0.0 | 2.2 | **91.1** | 6.7 |
| AO3 | $all(B,A); some\text{-}not(B,C)$ | 0.0 | 2.2 | 17.8 | 53.3 | **26.7** | 0.0 | 3.3 | 6.6 | 20.0 | **70.0** | 0.0 | 0.0 | 8.9 | 6.7 | **77.8** |
| AO4 | $all(A,B); some\text{-}not(B,C)$ | 0.0 | 4.4 | 11.1 | 55.6 | **28.9** | 0.0 | 3.3 | 0.0 | 13.3 | **83.3** | 0.0 | 2.2 | 0.0 | 6.7 | **91.1** |
| OA1 | $some\text{-}not(B,A); all(C,B)$ | 2.2 | 2.2 | 4.4 | 66.7 | **24.4** | 0.0 | 0.0 | 6.6 | 30.0 | **63.3** | 0.0 | 4.4 | 0.0 | 13.3 | **82.2** |
| OA2 | $some\text{-}not(A,B); all(C,B)$ | 2.2 | 2.2 | 4.4 | 64.4 | **26.7** | 0.0 | 6.6 | 3.3 | 26.6 | **63.3** | 0.0 | 6.7 | 2.2 | 8.9 | **82.2** |
| OA3 | $some\text{-}not(B,A); all(B,C)$ | 0.0 | 4.4 | 11.1 | **80.0** | 4.4 | 0.0 | 0.0 | 3.3 | **60.0** | 36.6 | 0.0 | 4.4 | 4.4 | **66.7** | 24.4 |
| OA4 | $some\text{-}not(A,B); all(B,C)$ | 0.0 | 11.1 | 20.0 | 42.2 | **26.7** | 3.3 | 3.3 | 6.6 | 16.6 | **66.6** | 0.0 | 2.2 | 4.4 | 2.2 | **91.1** |
| EI1 | $no(B,A); some(C,B)$ | 0.0 | 22.2 | 2.2 | **62.2** | 13.3 | 0.0 | 8.9 | 0.0 | **84.4** | 6.7 | 0.0 | 16.6 | 0.0 | **70.0** | 13.3 |
| EI2 | $no(A,B); some(C,B)$ | 0.0 | 26.7 | 4.4 | **46.7** | 22.2 | 3.3 | 10.0 | 0.0 | **73.3** | 10.0 | 0.0 | 8.9 | 0.0 | **84.4** | 6.7 |
| EI3 | $no(B,A); some(B,C)$ | 0.0 | 20.0 | 2.2 | **53.3** | 17.8 | 0.0 | 20.0 | 0.0 | **63.3** | 13.3 | 0.0 | 4.4 | 0.0 | **84.4** | 11.1 |
| EI4 | $no(A,B); some(B,C)$ | 0.0 | 28.9 | 6.7 | **35.6** | 28.9 | 0.0 | 10.0 | 0.0 | **73.3** | 16.6 | 0.0 | 6.7 | 0.0 | **75.6** | 15.6 |

**Conclusion*, A: $all(C,A)$, E: $no(C,A)$, I: $some(C,A)$, O: $some\text{-}not(C,A)$, N: *no-valid*.

ascribed to two well-known interpretational biases in linguistic syllogistic reasoning: conversion errors and figural effects. Our results indicate that these two types of effects were blocked in both Euler and Venn diagrammatic groups.

**Conversion errors.** It is well known that the categorical sentence "All $A$ are $B$" is sometimes misinterpreted as equivalent to "All $B$ are $A$." Similarly, "Some $A$ are not $B$" is sometimes misinterpreted as equivalent to "Some $B$ are not $A$" (cf. the experimental results in Newstead & Griggs, 1983). As shown by the experiments on *illicit conversion error* in Chapman and Chapman (1959) and Dickstein (1981), such misinterpretations may cause errors in syllogism, in particular those that have no valid conclusions (AA2, AA3, AA4, AI2, AI4, IA1, IA2, AE1, AE3, EA3, EA4, AO1, AO3, AO4, OA1, OA2, and OA4 types in our experiment). For example, in the case of AE1 syllogism, the first premise "All $B$ are $A$" is often misinterpreted as equivalent to "All $A$ are $B$," leading subjects to select the invalid conclusion "No $C$ are $A$." Note that in Euler diagrams, "All $A$ are $B$" and "All $B$ are $A$" correspond to $D_1^e$ and $D_2^e$ of Fig. 16, respectively, and in Venn diagrams, they correspond to $D_1^v$ and $D_2^v$, respectively. Here, one can immediately see that these two diagrams are topologically different, and hence, deliver different information. (Similarly for "Some $A$ are not $B$" and "Some $B$ are not $A$," which are represented in Euler and Venn diagrams in the same way; see $D_3$ and $D_4$ below)

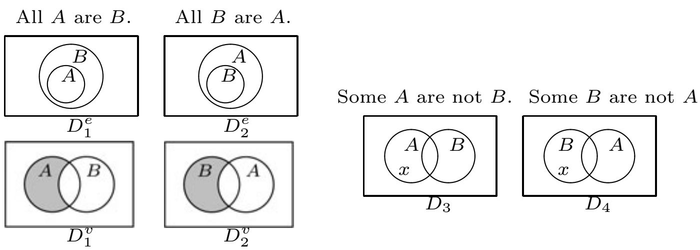



**Fig. 16.** Topologically non-identical pairs in Euler and Venn diagrams

Thus, the use of diagrams seems to block the errors caused by the misinterpretation of categorical sentences. This point is supported by our results: the performance in the class of syllogisms mentioned above (i.e., those that have no valid conclusions) was better when Euler and Venn diagrams were available to subjects. To look more closely, we divide the 17 invalid syllogisms into four groups based on the types of conclusions, which are mistakenly chosen.

**1)** In AA2 $all(A,B); all(C,B)$, AA3 $all(B,A); all(B,C)$ and AA4 $all(A,B); all(B,C)$ syllogisms, 58.5% of the subjects in the linguistic group selected the conclusion "All $C$ are $A$," while the rate reduced to 4.4% in the Euler group and to 21.1% in the Venn group. These data were also subjected to a one-way ANOVA. There was significant difference between the linguistic group and the Euler group, $F$ (1, 88) = 7.712, $p < .001$. There was significant difference between the Venn group and the linguistic group, $F$ (1, 73) = 4.773, $p < .001$. There was significant difference between the Euler group and the Venn group, $F$ (1, 73) = 2.125, at the reduced threshold of $p < .10$.

**2)** In AI2 $all(A,B); some(C,B)$, AI4 $all(A,B); some(B,C)$, IA1 $some(B,A); all(C,B)$ and IA2 $some(A,B); all(C,B)$ syllogisms, 56.1% of the subjects in the linguistic group selected the conclusion "Some $C$ are $A$," while the rate reduced to average 11.1% in the Euler group and 30.0% in the Venn group. There was significant difference between the linguistic and Euler groups, $F$ (1, 88) = 5.862, $p< .001$. There was significant difference between the Venn and linguistic groups, $F$ (1, 73) = 3.042, $p < .01$. There was significant difference between the Euler and Venn groups, $F$ (1, 73) = 2.201, $p < .05$.

**3)** In AE1 $all(B,A); no(C,B)$, AE3 $all(B,A); no(B,C)$, EA3 $no(B,A); all(B,C)$ and EA4 $no(A,B); all(B,C)$ syllogisms, 62.8% of the subjects in the linguistic group selected the conclusion "No $C$ are $A$," while the rate reduced to average 10% in the Euler group and to 37.3% in the Venn group. There was significant difference between the linguistic and Euler groups, $F$ (1, 88) = 7.214, $p < .001$. There was significant difference between the Venn and linguistic groups, $F$ (1, 73) = 2.540, $p < .01$. There was significant difference between the Euler and Venn groups, $F$ (1, 73) = 3.913, $p < .001$.

**4)** In AO1 $all(B,A); some\text{-}not(C,B)$, AO3 $all(B,A); some\text{-}not(B,C)$, AO4 $all(A,B); some\text{-}not(B,C)$, OA1 $some\text{-}not(B,A); all(C,B)$, OA2 $some\text{-}not(A,B); all(C,B)$ and OA4 $some\text{-}not(A,B); all(B,C)$ syllogisms, 58.1% of the subjects in the linguistic group selected the conclusion "Some $C$ are not $A$," while the rate reduced to average 8.5% in the Euler group and 27.2% in the Venn group. There was significant difference between the linguistic and Euler groups, $F$ (1, 88) = 9.835, $p < .001$. There was significant difference between the Venn and linguistic groups, $F$ (1, 73) = 5.485, $p < .001$. There was significant difference between the Euler and Venn groups, $F$ (1, 73) = 3.312, $p < .01$.

**Figural effects.** Because of the strict distinction between subject and predicate in categorical sentences, it is sometimes difficult to understand the logical equivalence between the E-type sentences "No $A$ are $B$" and "No $B$ are $A$" and also between the I-type sentences "Some $A$ are $B$" and "Some $B$ are $A$" (cf. Newstead & Griggs, 1983). Dickstein (1978) reported that such a difficulty appeared most prominently as a difference in the performances between EI1O and EI4O syllogisms, which have the above sentences as premises (EI1O refers to *No B are A, Some C are B; therefore Some C are not A.* EI4O refers to *No A are B, Some B are C; therefore Some C are not A.*). He also pointed out that the difference was a notable example of the *figural effect*.

In Euler and Venn diagrams, E-type and I-type sentences are represented as shown in Fig. 17. Here, it seems to be easy to understand the equivalence of $D_5^e$ and $D_6^e$ (also of $D_5^v$ and $D_6^v$, and of $D_7$ and $D_8$) since they are topologically identical. In fact, comparing EI1O and EI4O syllogisms, there was

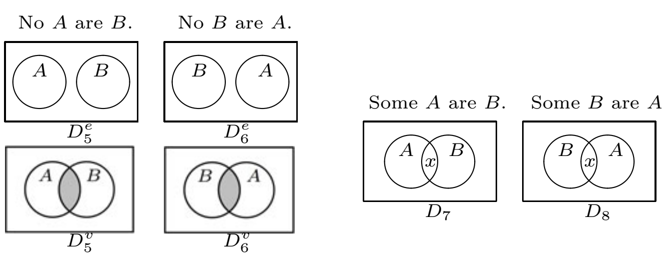



**Fig. 17.** Topologically identical pairs in Euler and Venn diagrams

significant difference between EI1O $no(B, A); some(C, B) : some\text{-}not(C, A)$ and EI4O $no(A, B); some(B, C) : some\text{-}not(C, A)$ in the linguistic group (62.2% for EI1O and 35.6% for EI4O) ($t$ (44) = 11.000, $p < .005$. t-test, within-subjects design). In contrast, there was no significant difference between EI1O and EI4O in the Euler group (84.4% for EI1O and 75.6% for EI4O) ($t$ (44) = 1.622, $p = .100$). Further, there was no significant difference between EI1O and EI4O in the Venn group (70.0% for EI1O and 73.3% for EI4O) ($t$ (29) = 1.000, $p = .100$).

## 5 Discussion

The performance of syllogistic reasoning in the Euler and Venn groups was significantly better than that in the linguistic group. These results present evidence for the claim that our Euler diagrams do help subjects solve syllogisms. However, if we are confined to the comparison between the performance of the Euler group and that of the linguistic group, it might be pointed out that people in the Euler group received much substantial instructions and 10 trials of practice (on diagram interpretation rather than syllogism solving), while people in the linguistic group did not. A potential objection is that this difference in training could have had a major impact on their differences in the performance in syllogism solving. However, such an objection can be avoided if we make a comparison between the Euler group and the Venn group. The latter also received substantial instructions about categorical sentences and trials of practice, but the result was that the performance of the Euler group was significantly better than that of the Venn group. The difference of performance between these two groups can be explained by our hypothesis that Euler diagrams not only contribute to the correct interpretations of categorical sentences but also play a substantial role in the inferential processes of syllogism solving. Euler diagrams themselves could aid subjects to construct diagrammatic proofs and thereby to solve syllogistic reasoning tasks. Given our experimental set-up where subjects were not taught the strategies for combining diagrams in syllogism solving, our empirical findings support the hypothesis that Euler diagrams are distinctive in that they are *self-guiding*, as specified in Section 3. Although many researchers in the field of diagrammatic logic have studied expressive but fairly complex systems derived from Venn diagrams, our empirical findings show that the available systems based on Euler diagrams are important as a support device for human deductive reasoning.

With respect to the performance of the Venn group, a reasonable explanation is that Venn diagrams could contribute to subjects' interpretations of categorical sentences, but could not play a substantial role in reasoning processes themselves. Hence, people in the Venn group would have to rely on inferences based on abstract semantic information extractable from sentences and diagrams rather than concrete syntactic manipulations of diagrams. This explanation also agrees with our results that the performance of the Venn group was significantly better than that of the linguistic group. The results can be explained by considering that Venn diagrams helped subjects' interpretations so that some well-known interpretational errors in syllogisms caused by the word order of a categorical sentence were blocked.

In our experiment, the subjects in the Venn group were provided with diagrams consisting of *two* circles that correspond to the premises of a given syllogism. Instead, we can also consider a set-up where in the beginning, subjects are provided with Venn diagrams consisting of *three* circles, namely $A, B$, and $C$, as in $D_3^v$ and $D_4^v$ of Fig. 4 in Section 3. In this set-up, the subjects could skip the first steps of adding new circles; the only step needed is to superpose the two premise diagrams (consisting of three circles). Thus it may be predicted that the performance would be improved, although shading and additional circles in premises might complicate people's understanding of diagrams. Indeed, in our pilot experiments, we obtained results confirming this prediction. The performance with Venn diagrams consisting of three circles was better than that with Venn diagrams consisting of two circles and was worse than that with Euler diagrams. For future research, it will be interesting to specify the conditions under which Venn diagrams would be more effective for subjects without substantial training of inferential strategies.

**Acknowledgments.** This project was partially supported by the Centre for Advanced Research on Logic and Sensibility in Keio University and the preparation of the manuscript was further supported by funding from the Japan Society for Promotion of Science to the authors. The authors would like to thank Mitsuhiro Okada, Takashi Iida, and Atsushi Shimojima for the valuable advice on this research and Yosaku Nishiwaki, Noburu Notomi, Tsuyoshi Yokoo, and Ryota Morimoto for their assistance in conducting this experiment.

## References

Barwise, J., Etchemendy, J.: Visual information and valid reasoning. In: Allwein, G., Barwise, J. (eds.) Logical Reasoning with Diagrams, pp. 3–26. Oxford University Press, Oxford (1991)

Bauer, M., Johnson-Laird, P.N.: How diagrams can improve reasoning. Psychology Science 4(6), 372–378 (1993)

Calvillo, P.D., DeLeeuw, K., Revlin, R.: Deduction with Euler Circles: Diagrams That Hurt. In: Barker-Plummer, D., Cox, R., Swoboda, N. (eds.) Diagrams 2006. LNCS (LNAI), vol. 4045, pp. 199–203. Springer, Heidelberg (2006)

Chapman, L., Chapman, J.: Atmosphere effect re-examined. Journal of Experimental Psychology 58(3), 220–226 (1959)

Dickstein, L.S.: The effect of figure on syllogistic reasoning. Memory and Cognition 6, 76–83 (1978)

Dickstein, L.S.: The meaning of conversion in syllogistic reasoning. Bulletin of the Psychonomic Society 18(3), 135–138 (1981)

Dobson, M.: Information enforcement and learning with interactive graphical systems. Learning and Instruction 9, 365–390 (1999)

Erickson, J.R.: A set analysis theory of behavior in formal syllogistic reasoning tasks. In: Solso, R. (ed.) Loyola Symposium on Cognition, vol. 2. Erlbaum, Mahwah (1974)

Gergonne, J.D.: Essai de dialectique rationelle. Annuales de Mathematiques pures et appliqukes 7, 189–228 (1817)
Gurr, C.A., Lee, J., Stenning, K.: Theories of diagrammatic reasoning: distinguishing component problems. Minds and Machines 8, 533–557 (1998)
Hammer, E.: Logic and Visual Information. CSLI Publications, Stanford (1995)
Hammer, E., Shin, S.: Euler's visual logic. History and Philosophy of Logic 19, 1–29 (1998)
Howse, J., Stapleton, G., Taylor, J.: Spider diagrams. LMS Journal of Comrutation and Mathematics 8, 145–194 (2005)
Johnson-Laird, P.N.: Mental Models: Towards a cognitive science of language, inference, and consciousness. Harvard University Press, Cambridge (1983)
Larkin, J., Simon, H.: Why a diagram is (sometimes) worth 10,000 words. Cognitive Science 11, 65–99 (1987)
Mineshima, K., Okada, M., Sato, Y., Takemura, R.: Diagrammatic reasoning system with Euler circles: theory and experiment design. In: Stapleton, G., Howse, J., Lee, J. (eds.) Diagrams 2008. LNCS (LNAI), vol. 5223, pp. 188–205. Springer, Heidelberg (2008)
Mineshima, K., Okada, M., Takemura, R.: A diagrammatic inference system with Euler circles (2009) (submitted to a journal)
Newstead, S., Griggs, R.: Drawing inferences from quantified statements: A study of the square of opposition. Journal of Verbal Learning and Verbal Behavior 22, 535–546 (1983)
Rizzo, A., Palmonari, M.: The mediating role of artifacts in deductive reasoning. In: Poster Presented at the 27th Annual Conference of the Cognitive Science Society (2005)
Scaife, M., Rogers, Y.: External cognition. International Journal of Human - Computer Studies 45, 185–213 (1996)
Shimojima, A.: Operational constraints in diagrammatic reasoning. In: Allwein, G., Barwise, J. (eds.) Logical Reasoning with Diagrams, pp. 27–48. Oxford University Press, Oxford (1996a)
Shimojima, A.: On the Efficacy of Representation. PhD thesis, Indiana University (1996b)
Shin, S.-J.: The Logical Status of Diagrams. Cambridge University Press, Cambridge (1994)
Sloman, A.: Interactions between philosophy and ai: the role of intuition and non-logical reasoning in intelligence. Artificial Intelligence 2, 209–225 (1971)
Stapleton, G.: A survey of reasoning systems based on Euler diagrams. In: Euler Diagrams 2004. ENTCS, vol. 134, pp. 127–151. Elsevier, Amsterdam (2005)
Stenning, K.: The cognitive consequences of modality assignment for educational communication: the picture in logic teaching. Learning and Instruction 9, 391–410 (1999)
Stenning, K.: Seeing Reason: Image and Language in Learning to Think. Oxford University Press, Oxford (2002)
Stenning, K., Oberlander, J.: A cognitive theory of graphical and linguistic reasoning. Cognitive Science 19, 97–140 (1995)

# Drawing Euler Diagrams with Circles

Gem Stapleton[1], Leishi Zhang[2], John Howse[1], and Peter Rodgers[2]

[1] Visual Modelling Group, University of Brighton, UK
{g.e.stapleton,john.howse}@brighton.ac.uk
[2] University of Kent, Canterbury, UK
{l.zhang,p.j.rodgers}@kent.ac.uk

**Abstract.** Euler diagrams are a popular and intuitive visualization tool which are used in a wide variety of application areas, including biological and medical data analysis. As with other data visualization methods, such as graphs, bar charts, or pie charts, the automated generation of an Euler diagram from a suitable data set would be advantageous, removing the burden of manual data analysis and the subsequent task of drawing an appropriate diagram. Various methods have emerged that automatically draw Euler diagrams from abstract descriptions of them. One such method draws some, but not all, abstract descriptions using only circles. We extend that method so that more abstract descriptions can be drawn with circles, allowing sets to be represented by multiple curves. Furthermore, we show how to transform any 'undrawable' abstract description into a drawable one by adding in extra zones. Thus, given any abstract description, our method produces a drawing using only circles. A software implementation of the method is available for download.

## 1 Introduction

It is commonly the case that data can be more easily interpreted using visualizations. One frequently sees, for instance, pie charts used in statistical data analysis and graphs used for representing network data. These visualizations are often automatically produced, allowing the user to readily make interpretations that are not immediately apparent from the raw data set. Sometimes, the raw data are classified into sets and one may be interested in the relationships between the sets, such as whether one set is a subset of another or whether one set contains more elements than another.

For example, the authors of [6] have data concerning health registry enrollees at the world trade centre. Each person in the health registry is classified as being in one or more of three sets: rescue/recovery workers and volunteers; building occupants, passers by, and people in transit; and residents. In order to visualize the distribution of people amongst these three sets, the authors of [6] chose to use an Euler diagram which can be seen in figure 1. A further example, obtained from [16], shows a visualization of five sets of data drawn from a medical domain. The authors of [16] chose to represent one of the sets (Airflow Obstruction Int) using multiple curves. Other areas where Euler diagrams are used for information visualization include crime control [7], computer file organization [4], classification systems [20], education [10], and genetics [12].

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 23–38, 2010.

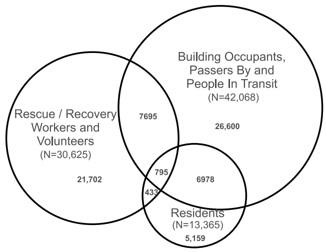



**Fig. 1.** Data visualization

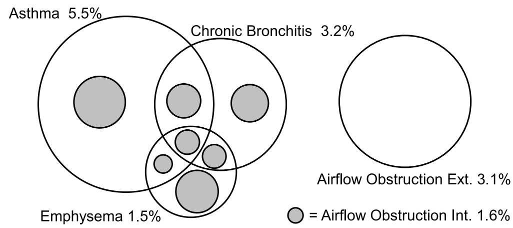



**Fig. 2.** Using multiple circles

As with other diagram types for data visualization, the ability to automatically create Euler diagrams from the data would be advantageous. To date, a range of methods for automatically drawing Euler diagrams have been developed, with most of them starting with an abstract description of the required diagram. The existing methods can be broadly classified into three classes.

**Dual Graph Methods:** With these methods, a so-called dual graph of the required Euler diagram is identified and embedded in the plane. Then the Euler diagram is formed from the dual graph. Methods in this class include the first Euler diagram drawing technique, attributable to Flower and Howse [8]. Others who have developed this class of drawing method include Verroust and Viaud [22], Chow [2], and Simonetto et al. [15]. Recently, Rodgers et al. have developed a general dual graph based method that is capable of drawing a diagram given any abstract description [13]. Some of these methods allow the use of many curves to represent the same set (as in figure 2) to ensure drawability.

**Inductive Methods:** Here, one curve of the required Euler diagram is drawn at a time, building up the diagram as one proceeds. This is a recently devised method, attributable to Stapleton et al. [18], and builds on similar work for Venn diagrams [5,21]. Stapleton et al.'s method is also capable of drawing a diagram given any abstract description and it has advantages over the dual graph based methods in that it readily incorporates user preference for properties that the to-be-drawn diagram is to possess.

**Methods using Particular Shapes:** A large number of methods attempt to draw Euler diagrams using particular geometric shapes, typically circles, because they are aesthetically pleasing. Chow considers drawing diagrams with exactly two circles [2], which is extended to three circles by Chow and Rodgers [3]. The Google Charts API includes facilities to draw Euler diagrams with up to three circles [1] and Wilkinson's method allows any number of circles but it often fails to produce diagrams with the specified abstract description [23]; Wilkinson's diagrams can contain too few zones and, thus, fail to convey the correct semantics. Similarly, Kestler et al. devised a method that draws Euler diagrams with regular polygons but it, too, does not guarantee that the diagrams have the required zones [11]. In previous work, we have devised a method for drawing a particular class of abstract descriptions with circles, which does ensure the correct abstraction is achieved [19]. However, none of these methods is capable of drawing an Euler diagram given an arbitrary abstract description. In part, this is because many abstract descriptions are not drawable with a circles or regular polygons, given the constraints imposed by the authors on the properties that the diagrams are to possess (such as no duplicated curve labels). A distinct advantage of this class of methods is that they can produce aesthetically pleasing diagrams.

In this paper, we take the method of [19] and extend it, so that every abstract description is (essentially) drawable by adding zones and allowing sets to be represented by more than one curve (as in figure 2). Our method takes the abstract description and draws a diagram with circles that contains all required zones, but may contain additional zones; any extra zones are shaded. Section 2 presents necessary background material on Euler diagrams, along with some new concepts that are particular to the work in this paper. Abstract descriptions are defined in section 3 and we provide various definitions of abstract-level concepts. Section 4 describes the class of inductively pierced abstract descriptions developed in [19], on which the results in this paper build. Our drawing method is described in section 5. Section 6 shows some output from the software implementation of the method, alongside diagrams drawn using previously existing methods.

## 2 Euler Diagrams

An Euler diagram is a set of closed curves drawn in $\mathbb{R}^2$. Each curve has a label chosen from some fixed set of labels, $\mathcal{L}$. Our definition of an Euler diagram is consistent with, or a generalization of, those found in the literature, such as in [2,8,17,22]. An **Euler diagram** is a pair, $d = (Curve, l)$, where

1. $Curve$ is a finite set of closed curves in $\mathbb{R}^2$, and
2. $l\colon Curve \to \mathcal{L}$ is a function that returns the label of each curve.

A **minimal region** of $d$ is a connected component of

$$\mathbb{R}^2 - \bigcup_{c \in Curve} image(c)$$

where $image(c)$ is the set of points in $\mathbb{R}^2$ to which $c$ maps. We define the set of curves in a diagram with some specified label, $\lambda$, to be a **contour** with label $\lambda$. The diagram $d_1$ in figure 3 has four contours, but five curves. A point, $p$, is inside a contour precisely when the number of the contour's curves that $p$ is is inside is odd. Another important concept is that of a **zone**, which is a set of minimal regions that can be described as being inside certain contours (possibly none) and outside the rest of the contours. The diagram $d_1$ in figure 3 has 11 zones, each of which is a minimal region.

There are a collection of properties that it is desirable for Euler diagrams to possess, since they are often thought to correlate with the ease with which the diagrams can be interpreted. The most commonly considered properties are:

1. **Unique Labels**: no curve label is used more than once.
2. **Simplicity**: all curves are simple (have no self-intersections).
3. **No Concurrency**: the curves intersect at a discrete set of points (i.e. no curves run along each other in a concurrent fashion).
4. **Only Crossings**: whenever two curves intersect, they cross.
5. **No 3-points**: there are no 3-points of intersection between the curves (i.e. any point in the plane is passed through at most 3 times by the curves).
6. **Connected Zones**: each zone consists of exactly one minimal region.

A diagram, $d$, possessing all of these properties is **completely wellformed**. Neither diagram in figure 3 is completely wellformed, since both use the curve label $R$ twice and, thus, in each diagram the set $R$ is represented by more than one curve. Now, $d$ is **completely wellformed up to labelling** if it possesses all properties except, perhaps, the unique labels property. If all of the curves in $d$ are circles then $d$ is **drawn with circles**. Our drawing method only produces diagrams drawn with circles that are completely wellformed up to labelling.

Further concepts that we need concern the topological adjacency of zones and 'clusters' of topologically adjacent zones. We define these concepts only for diagrams that are completely wellformed up to labelling, since this is sufficient for our purposes. In particular, in such diagrams we know that two zones which are topologically adjacent are separated by a single curve. For example, in figure 3, the zones $z_2$ and $z_3$ are topologically adjacent in $d_1$, separated by the leftmost curve labelled $R$; when this curve is removed, $z_2$ and $z_3$ form a minimal region. The zones $z_6$ and $z_{11}$ are not topologically adjacent and neither are $z_2$ and $z_4$.

Let $z_1$ and $z_2$ be zones in $d = (Curve, l)$. If there exists a curve, $c$, in $Curve$ such that $z_1$ and $z_2$ form a minimal region in the diagram $(Curve - \{c\}, l - \{(c, l(c))\})$ then $z_1$ and $z_2$ are **topologically adjacent** in $d$ **separated** by $c$. Regarding our drawing problem, we could choose to draw a circle that splits two adjacent zones and which intersects their separating curve. We call topologically adjacent zones $z_1$ and $z_2$ a **cluster** given $c$. We also define a cluster comprising four zones. Let $c_1$ and $c_2$ be distinct curves in $d$, that intersect at some point $p$. The four zones in the immediate neighbourhood of $p$ (since we are assuming wellformedness up to labelling, precisely four such zones exist) form a **cluster** given $c_1$, $c_2$ and $p$, denoted $C(c_1, c_2, p)$. In figure 3, the zones $z_3$, $z_4$, $z_6$ and $z_7$ form a cluster given $Q$ and $S$ (blurring the distinction between the curves and their labels). Given a cluster of four zones, we can draw a circle around the point $p$ that splits all and only these zones.

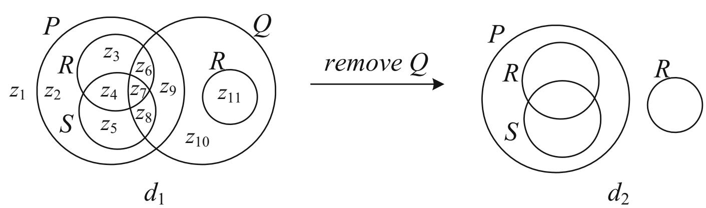



**Fig. 3.** Euler diagram concepts

## 3 Abstract Descriptions

As is typical Euler diagram drawing methods, we start with an abstract description of the required diagram. This description tells us which zones are to be present. An **abstract description**, $D$, is a pair, $(L, Z)$, where

1. $L$ is a finite subset of $\mathcal{L}$ (i.e. all of the labels in $D$ are chosen from the set $\mathcal{L}$) and we define $L(D) = L$,
2. $Z \subseteq \mathbb{P}L$ such that $\emptyset \in Z$ and for each $\lambda \in L$ there is a zone, $z$, in $Z$ where $\lambda \in z$ and we define $Z(D) = Z$.

The abstract description, $D$, of $d_2$ in figure 3 has labels $\{P, R, S\}$ and zones $\{\emptyset, \{P\}, \{R\}, \{P, R\}, \{P, S\}, \{P, R, S\}\}$; we say that $d_2$ is a *drawing* of $D$. We will sometimes abuse notation, omitting the label set and writing the zone set as, for instance, $\{P,\ R,\ PR,\ PS,\ PRS\}$.

It is not possible to identify whether two zones will necessarily be topologically adjacent when presented only with an abstract description. However, we can observe that, in a diagram that does not possess any concurrency, two zones that are topologically adjacent have abstractions that differ by a single curve label. For example, the topologically adjacent zones $z_2$ and $z_3$ in figure 3 have abstractions $\{P\}$ and $\{P, R\}$ which differ by $R$, the label of their separating curve. We use this observation to define an abstract notion of a cluster. Let $z$ be an abstract zone (i.e. a finite set of labels) and let $\Lambda \subseteq \mathcal{L}$ be a set of labels disjoint from $z$. The set $\{z \cup \Lambda_i : \Lambda_i \subseteq \Lambda\}$ is a **$\Lambda$-cluster** for $z$, denoted $\mathcal{C}(z, \Lambda)$. The cluster $\mathcal{C}(\{P, R\}, \{Q, S\}, d_1)$ is the cluster $\{PR,\ PQR,\ PRS,\ PQRS\}$ and corresponds to the cluster $\{z_3, z_4, z_6, z_7\}$ in $d_1$, in figure 3. In general, a set of zones in a diagram that form a cluster will have abstractions that form a cluster. However, a set of zones may have abstractions that form a cluster but need not themselves be a cluster in the drawn diagram. For example, $z_6$ and $z_{11}$, figure 3, do not form a cluster but their abstractions, $\{R, Q\}$ and $\{P, R, Q\}$, are a cluster.

Further abstract level concepts are useful to us. Our drawing method first draws curves that are not contained by any other curves and 'works inwards' drawing contained curves later in the process. We can identify at the abstract level whether a contour, $C_1$, is to be contained by another, $C_2$, and, as such, in any drawing $C_2$'s curves will each be contained by at least one of $C_1$'s curves. We are also interested in which abstract zones are contained by which curve labels.

Let $D = (L, Z)$ be an abstract description and let $\lambda_1$ and $\lambda_2$ be distinct curve labels in $L$. If $\lambda_1 \in z$ and $z \in Z$ then we say $\lambda_1$ **contains** $z$ in $D$ with the set of such zones denoted $Z_c(\lambda_1)$. If $Z_c(\lambda_1) \subset Z_c(\lambda_2)$ then $\lambda_2$ **contains** $\lambda_1$ in $D$. The set of curves that contain $\lambda_1$ in $D$ is denoted $L^c(\lambda_1)$. In the abstract description (given above) for $d_2$ of figure 3, the curve label $P$ contains the curve label $S$ but not the curve label $R$. This reflects the fact that, in $d_2$, the contour labelled $P$ does not contain the contour labelled $R$.

We need an operation to remove curve labels from abstraction descriptions. Given an abstract description, $D = (L, Z)$, and $\lambda \in L$, we define $D - \lambda$ to be $D - \lambda = (L - \{\lambda\}, \{z - \{\lambda\} : z \in Z\})$. The abstract description for $d_1$ in figure 3 becomes the abstract description for $d_2$ on the removal of $Q$. A **decomposition** of $D$ is a sequence, $dec(D) = (D_0, D_1, ..., D_n)$ where each $D_{i-1}$ $(0 < i \leq n)$ is obtained from $D_i$ by the removal of some label, $\lambda_i$, from $D_i$ (so, $D_{i-1} = D_i - \lambda_i$) and $D_n = D$. If $D_0$ contains no labels then $dec(D)$ is a **total decomposition**.

## 4 Inductively Pierced Descriptions

A class of abstract descriptions that can be drawn with circles in a completely wellformed manner can be built by successively adding *piercing curves*. Figure 4 shows a sequence of diagrams where, at each stage, the curve added is a piercing curve. This section summarizes results in [19] and adds a new concept of an inductively pierced diagram. The following definition is generalized from [19].

**Definition 1.** *Let $D = (L, Z)$ be an abstract description. Let $\lambda_1, \lambda_2, ..., \lambda_{n+1} \in L$ be distinct curve labels. Then $\lambda_{n+1}$ is an **n-piercing** of $\lambda_1$, ..., $\lambda_n$ in $D$ if there exists a zone, $z$, such that*

1. $\lambda_i \notin z$ *for each* $i \leq n+1$
2. $Z_c(\lambda_{n+1}) = \mathcal{C}(z \cup \{\lambda_{n+1}\}, \{\lambda_1, ..., \lambda_n\})$, *and*
3. $\mathcal{C}(z, \{\lambda_1, ..., \lambda_n\}) \subseteq Z$.

*The zone $z$ is said to **identify** $\lambda_{n+1}$ as a piercing.*

In figure 4, the curve $S$ is a 1-piercing of $R$ in $d_4$. If an abstract description can be built by successively adding 0-piercing, 1-piercing, or 2-piercing curves then, usually, it can be drawn with circles in a completely wellformed manner. However, there are occasions when this is not possible. For example, in figure 5, we may want to add a curve, $T$, to $d_3$ that is a 2-piercing of $P$ and $Q$. However, it is not possible to do so using a circle whilst maintaining wellformedness. Thus, the definition of an inductively pierced description, which allows only 0, 1, or 2-piercings, restricts the ways in which 2-piercings can arise.

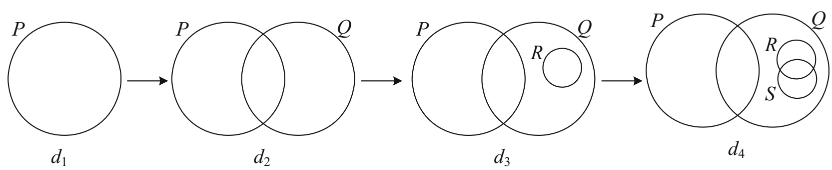



**Fig. 4.** An inductively pierced diagram

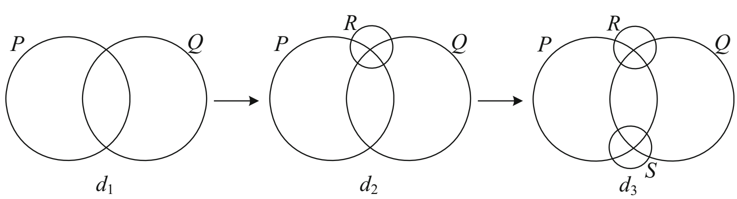



**Fig. 5.** Adding three 2-piercing curves

**Definition 2.** *Let $\mathcal{C}_1 = \mathcal{C}(z, \{\lambda_1, \lambda_2\})$ and $\mathcal{C}_2 = \mathcal{C}(z \cup \{\lambda_3\}, \{\lambda_1, \lambda_2\})$ be clusters. Let $D = (L, Z)$ be an abstract description. If $C_1 \cup C_2 \subseteq Z$ then $\lambda_3$ is* ***outside-associated*** *with $C_2$ in $D$ and is* ***inside-associated*** *with $C_1$ in $D$.*

**Definition 3.** *Let $D = (L, Z)$ be an abstract description. Then $D$ is* ***inductively pierced*** *if either*

1. *$D = (\emptyset, \{\emptyset\})$, or*
2. *$D$ has a 0-piercing, $\lambda$, such that $D - \lambda$ is inductively pierced, or*
3. *$D$ has a 1-piercing, $\lambda$, such that $D - \lambda$ is inductively pierced, or*
4. *$D$ has a 2-piercing, $\lambda_3$, of $\lambda_1$ and $\lambda_2$ identified by $z$, and either*
   (a) *no other curve label, $\lambda_4$, in $D$ is outside-associated with the cluster $\mathcal{C}(z, \{\lambda_1, \lambda_2\})$ or*
   (b) *exactly one other curve label, $\lambda_4$, in $D$ is outside-associated with the cluster $\mathcal{C}(z, \{\lambda_1, \lambda_2\})$ and we have either*
      i. *$L^c(\lambda_3) = L^c(\lambda_4) = L^c(\lambda_1)$ or*
      ii. *$L^c(\lambda_3) = L^c(\lambda_4) = L^c(\lambda_2)$.*

   *and $D - \lambda_3$ is inductively pierced.*

All of the diagrams in figures 4 and 5 have inductively pierced descriptions whereas the diagram $d_1$ in figure 3 does not.

**Definition 4.** *A diagram, $d$, is* ***inductively pierced*** *if either $d$ contains no curves or the following hold:*

1. *$d$ is drawn entirely with circles,*
2. *$d$ is completely wellformed,*
3. *given any pair of abstract zones, $z_1$ and $z_2$, in $d$'s abstraction, $D$, if the symmetric difference of $z_1$ and $z_2$ contains exactly one label, $\lambda$, then in $d$ the zones with abstractions $z_1$ and $z_2$ are topologically adjacent, separated by the curve labelled $\lambda$, and*
4. *there is a circle, $c$, whose label is an $i$-piercing ($i \leq 2$) in the abstraction, $D$, of $d$, and the diagram obtained from $d$ by removing $c$ is inductively pierced.*

The diagrams in figures 4 and 5 are inductively pierced. However, the diagram $d_2$ in figure 3 has an inductively pierced abstract description but $d_2$ itself is not inductively pierced; it can be redrawn in an inductively pierced manner.

**Theorem 1.** *Let $D$ be an inductively pierced abstract description. Then there exists an inductively pierced drawing, $d$, of $D$. Moreover such a $d$ can be drawn in polynomial time, [19].*

Presented in [19] is a detailed algorithm to draw $d$ given $D$, as in theorem 1.

## 5 Drawing with Circles

We will now demonstrate how to turn an arbitrary abstract description into another abstract description that can be drawn in an inductively pierced manner, except that it may have duplicated curve labels. A diagram is **inductively pierced up to curve relabelling** if there exists a relabelling of its curves so that the curve labels are unique and the resulting diagram is inductively pierced. The diagram $d_2$ in figure 3 is inductively pierced up to curve relabelling. In addition, $d_1$ is also inductively pierced up to curve relabelling but, unlike $d_2$, its abstract description is not inductively pierced.

It is helpful to summarize the initial stages our drawing process. We take an abstract description, $D$, and find a total decomposition, $dec(D) = (D_0, ..., D_n)$ of $D$. At least one of the $D_i$s is an inductively pierced subdescription of $D_n$ (for instance, $D_0$ is inductively pierced). We can draw such a $D_i$, yielding $d_i$, using the methods of [19] which draws $D_i$ by adding an appropriate circle to the drawing of $D_{i-1}$. Once we reach the first $D_j$ which is not inductively pierced, we start to draw contours consisting of more than one circle. We will address how to choose sensibly a decomposition and how to add the remaining contours to $d_{j-1}$ in order to obtain $d$. We point the reader to subsection 5.4, which includes a comprehensive illustration of our drawing method.

### 5.1 Choosing a Decomposition

There are choices about the order in which the curve labels are removed when producing a decomposition of an abstract description and we prioritize removing curve labels that do not contain other curve labels; this choice will be discussed below.

**Definition 5.** *Let $D = (L, Z)$ be an abstract description that contains curve label $\lambda$. We say that $\lambda$ is* ***minimal*** *if $\lambda$ does not contain any curve labels in $D$.*

In figure 6, $d_1$'s abstract description has minimal curve labels $R$, $S$ and $T$, whereas for $d_2$ the minimal labels are $R$, $U$ and $V$. Trivially, every abstract description, $D$ (with $L(D) \neq \emptyset$), contains at least one minimal curve label and, moreover, every piercing curve is minimal. When producing a decomposition, our method removes a minimal curve label at each step. This ensures that, when we draw the diagram (the process for which is described later), if curve label $\lambda_1$ is contained by curve label $\lambda_2$ then the contour, $c_1$, for $\lambda_1$ will be drawn inside the contour, $c_2$, for $\lambda_2$. This nicely reflects the semantics of the diagram: if $\lambda_1$ represents a proper subset of $\lambda_2$ then $c_1$ will be contained by $c_2$.

**Definition 6.** *Let $D = (L, Z)$ be an abstract description. To produce a* ***chosen total decomposition*** *of $D$ carry out the following steps:*

1. *Set $i = n$, where $|L(D)| = n$ and define $D = D_i$ and $dec_i(D) = (D)$.*
2. *Identify a minimal curve label, $\lambda$, in $D$.*
3. *Remove $\lambda$ from $D_i$ to give $D_{i-1}$.*
4. *Form $dec_{i-1}(D)$ by copying $dec_i(D)$ and placing $D_{i-1}$ at the beginning.*
5. *If $i > 1$ decrease $i$ by 1 and return to step 2. Otherwise $dec_i$ is a chosen total decomposition.*

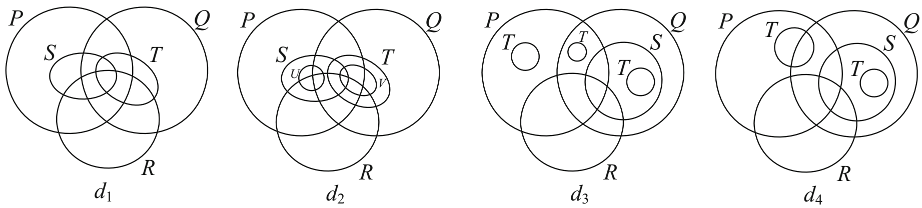



**Fig. 6.** Choosing a decomposition

In figure 6, we could remove the curve labels in the following order to produce a chosen total decomposition of the abstract description for $d_2$: $U \to V \to S \to T \to R \to P \to Q$; here we obtain an inductively pierced abstract description on the removal of $S$. An alternative order is $V \to T \to U \to S \to R \to Q \to P$.

### 5.2 Transforming Decompositions

We would like to be able to visualize abstract description, $D$, using only circles (which are aesthetically pleasing) at the expense of duplicating curve labels. If $D$ is an arbitrary abstract description this is, unfortunately, not necessarily possible. However, it is always possible to add zones to $D$ and realize an abstract description that is drawable in this manner. Here, we show how to add sufficient zones to $D$ to ensure drawability, given a chosen total decomposition, $dec(D) = (D_0, ..., D_n)$.

We observe that, when removing $\lambda_i$ from $D_{i+1}$ to obtain $D_i$, the zone set $Z(D_i)$ can be expressed as $Z(D_i) = in_i \cup out_i$, where

1. $in_i = \{z \in Z(D_i) : z \cup \{\lambda_i\} \in Z(D_{i+1})\}$, and
2. $out_i = \{z \in Z(D_i) : z \in Z(D_{i+1})\}$.

We say that the zone sets $in_i$ and $out_i$ are defined by $D_i$ and $D_{i+1}$. If $\lambda_i$ is a piercing curve label then $in_i \subseteq out_i$, since $\lambda_i$ 'splits' all of the zones through which it passes (if a piece of a zone is inside $\lambda_i$ then a piece is also outside $\lambda_i$). consider a zone, $z$, that is in $in_i$ but not in $out_i$. Then $z$ is not split by $\lambda_i$ and $z \notin Z(D_{i+1})$; transforming $D_{i+1}$ by adding $z$ to $Z(D_{i+1})$ will result in $z$ being split by $\lambda_i$ and being added to $out_i$. We transform $dec(D)$ into a new sequence of abstract descriptions that ensure all zones passed through are split on the addition of $\lambda_i$. This transformation process is defined below.

The addition of these zones removes any need for concurrency in the drawings. For instance, suppose we wish to add a contour labelled $U$ to $d_4$ in figure 6, so that the zone $\{P\}$ is contained by $U$ and all other zones are outside $U$. Then the new curve would need to run along the boundary of the zone $\{P\}$ and, therefore, be (partially) concurrent with the curves $P$, $R$, and $T$. Altering this curve addition so that the zone $\{P\}$ is instead split by $U$ allows us to draw $U$ as a circle inside the zone $\{P\}$, and the 'extra' zone will be shaded.

**Definition 7.** *Given a chosen, total decomposition, $dec(D) = (D_0, ..., D_n)$, transform $dec(D)$ into a* ***splitting super-decomposition****, $dec(D') = (D'_0, ..., D'_n)$, associated with $D$ as follows:*

1. *$D_0$ remains unchanged, that is $D_0 = D'_0$.*
2. *$D_{i+1} = (L_{i+1}, Z_{i+1})$ is replaced by $D'_{i+1} = (L_{i+1}, Z'_{i+1})$ where*

$$Z'_{i+1} = Z_{i+1} \cup \bigcup_{j \leq i} in_j$$

*where $in_j$ is as defined above, given $D_j$ and $D_{j+1}$.*

Given a splitting super-decomposition associated with $D$, we know that if $D_i$ is inductively pierced then $D'_i = D_i$.

**Theorem 2.** *A splitting super-decomposition, $dec(D') = (D'_0, ..., D'_n)$, associated with $D$ is a total decomposition of $D'_n$.*

Our problem is now to find a drawing of $D'_n$ rather than $D_n$. We note that $D'_n$ has a superset of $D_n$'s zones and we will use shading, as is typical in the literature, to indicate that the extra zones are not required (semantically, the extra zones represent the empty set).

### 5.3 Contour Identification and the Drawing Process

Given a splitting super-decomposition, $dec(D') = (D'_0, ..., D'_n)$, we are in a position to start drawing our diagram. First, we identify $D'_i$ in $dec(D')$ such that $D'_i$ is inductively pierced but $D'_{i+1}$ is not inductively pierced. We draw $D'_i$, using the methods of [19], yielding an inductively pierced drawing of $D'_i$. The manner in which we add the remaining curves using partitions (described below) also shows how $D'_i$ is drawn; in the inductively pierced case, there is one 'valid partition' that includes all zones in $in'_j$ which gives rise to one circle.

Suppose, without loss of generality, that we have obtained a drawing, $d'_j$, of $D'_j$, where $j \geq i$, that is inductively pierced up to curve relabelling (so it is drawn with circles). It is then sufficient to describe how to add a contour, labelled $\lambda_j$, to $d'_j$ in order to obtain such a drawing, $d'_{j+1}$, of $D'_{j+1}$. This will justify that $D'_n$ has a drawing that is inductively pierced up to curve relabelling.

Consider the sets $in'_j$ and $out'_j$ which describe, at the abstract level, how to add $\lambda_j$ to $d'_j$: the zones in $in_j$ are to be split by curves labelled $\lambda_j$ whereas those in $out_j$ are to be completely outside curves labelled $\lambda_j$. Trivially, we can draw one circle inside each zone of $d'_j$ whose abstraction is in $in'_j$ to obtain $d'_{j+1}$; label each such circle $\lambda_j$. See figure 6, where the contour $T$ has been drawn in this manner in $d_3$ given the set $in = \{P,\ PQ,\ QS\}$.

**Theorem 3.** *Let $dec(D) = (D_0, ..., D_n)$ be a decomposition with splitting super-decomposition $dec(D') = (D'_0, ..., D'_n)$. Then $dec(D')$ has a drawing, $d$, that is inductively pierced up to curve relabelling.*

Of course, the justification of the above theorem (drawing one circle in each split zone) may very well give rise to contours consisting of more curves than is absolutely necessary, as in $d_3$ of figure 6. We seek methods of choosing how to draw each contour using fewer curves. Consider the drawing, $d'_j$, of $D'_j$. We know that each zone in $in'_j$ is to be split by the to-be-added contour. We partition $in'_j$ into sets of zones, according to whether they are topologically adjacent or form a cluster in $d'_j$. The sets in the partition will each give rise to a circle labelled $\lambda_j$ in $d'_{j+1}$. In $d_3$ of figure 6, the zones $P$ and $PQ$ form a cluster, so $in = \{P,\ PQ,\ QS\}$ can be partitioned into two sets: $\{\{P,\ PQ\},\ \{QS\}\}$. Using this partition, we draw $d_4$ in figure 6 rather than $d_3$.

**Definition 8.** *A partition of $in'_j$ is **valid** given $d'_j$ if each set, $S$, in the partition ensures the following:*

1. *$S$ is a cluster that contains 1, 2 or 4 zones,*
2. *if $|S| = 2$ then the zones in $d'_j$ whose abstractions are in $S$ are topologically adjacent given a curve whose label is in the symmetric difference of the zones in $S$, and*
3. *if $|S| = 4$ then there exists a pair of curves, $c_1$ and $c_2$, that intersect at some point $p$ in $d'_j$ such that the zones in $d'_j$ whose abstractions are in $S$ form a cluster given $c_1$, $c_2$ and $p$.*

Each set, $S$, in a valid partition gives rise to a circle in $d'_{j+1}$:

1. if $|S| = 1$ then draw a circle inside the zone whose abstraction is in $S$,
2. if $|S| = 2$ then draw a circle that intersects $c$ (as described in 2 above), and no other curves, and that splits all and only the zones whose abstractions are in $S$, and
3. if $|S| = 4$ then draw a circle around $p$ (as described in 3 above) that intersects $c_1$ and $c_2$, and no other curves, and that splits all and only the zones whose abstractions are in $S$.

There are often many valid partitions of $in'_j$ and we may want to use heuristics to guide us towards a good choice. One heuristic is to minimize the number of sets in the partition, since each set will give rise to a circle in the drawn diagram. In figure 2, the contour consisting of multiple curves would arise from a valid partition with the largest number of sets.

### 5.4 Illustrating the Drawing Method

We now demonstrate the drawing method via a worked example, starting with $D = \{\emptyset, P, PQ, R, PR, QR, PQR, PS, PQS, PRS, PQRS, QS\}$. Since there are four curve labels, as the first step in producing a chosen total decomposition, we define $D = D_4$. Next, we identify $S$ as a minimal curve label and remove $S$ to give $D_3 = \{\emptyset, P, PQ, R, PR, QR, PQR, Q\}$. Similarly, we identify $R$, then $Q$, then $P$ as minimal, giving $dec(D) = (D_0, D_1, D_2, D_3, D_4)$ as a chosen decomposition of $D$, where $D_2 = \{\emptyset, P, PQ, Q\}$, $D_1 = \{\emptyset, P\}$, and $D_0 = \{\emptyset\}$. The table summarizes $in_i$ and $out_i$ at each step, and gives $Z'_i$ (the zone sets of the abstract descriptions in the splitting super-decomposition):

| $D_i$ | $in_i$ | $out_i$ | $Z'_i$ |
|---|---|---|---|
| $D_0$ | $\{\emptyset\}$ | $\{\emptyset\}$ | $Z(D_0)$ |
| $D_1$ | $\{\emptyset, P\}$ | $\{\emptyset, P\}$ | $Z(D_1)$ |
| $D_2$ | $\{\emptyset, P, PQ, Q\}$ | $\{\emptyset, P, PQ, Q\}$ | $Z(D_2)$ |
| $D_3$ | $\{P, PQ, PR, PQR, Q\}$ | $\{\emptyset, P, PQ, R, PR, QR, PQR\}$ | $Z(D_3)$ |
| $D_4$ | – | – | $Z(D_4) \cup \{Q\}$ |

Thus, the splitting super-decomposition is $dec(D') = (D'_0, D'_1, D'_2, D'_3, D'_4)$ where $D_i = D'_i$ for $i \leq 3$ and $D'_4$ has zone set $Z(D_4) \cup \{Q\}$. We note that $D'_3$ is an abstract description of Venn-3, the Venn diagram with three curves, and is drawn by our method as $d'_3$ in figure 7. To $d'_3$ we wish to add a contour labelled $S$; note that $in'_3 = \{P, PQ, PR, PQR, Q\}$ and $out'_3 = \{\emptyset, P, PQ, R, PR, QR, PQR, Q\}$. Given $d'_3$, $\{\{P, PQ, PR, PQR\}, \{Q\}\}$ is a valid partition of $in'_3$. Using this partition, we obtain $d'_4$ where the zone with abstraction $\{Q\}$ is shaded, since $\{Q\}$ is in $D'_4$ but not in $D_4$.

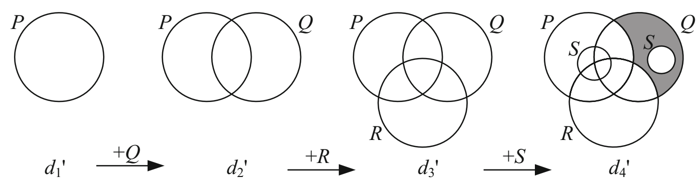



**Fig. 7.** Illustrating the drawing method

Our drawing method ensures some properties are possessed by the drawn diagrams, in addition to being completely well-formed up to labelling and consisting only of circles. Ideally, we want to minimize the number of shaded zones and the number of curves of which each contour consists. In particular, we note:

(1) Choosing to remove minimal labels ensures that if one contour, $C_1$, represents a proper subset of another contour, $C_2$, then all of $C_1$'s curves are drawn inside curves of $C_2$ thus ensuring 'enclosure' corresponds to 'subset'.
(2) Minimal curve labels contain fewer zones than the curve labels that contain them. Since we remove only minimal curve labels, it is likely that each contour consists of fewer curves when we draw the diagram. The intuitive justification for this that $in_i$ will have smaller cardinality when removing $C_2$ than when removing $C_1$, where $C_1$ contains $C_2$ (a smaller $in_i$ will have fewer partitions).
(3) The manner in which we transform decompositions ensures that a minimal number of shaded zones are present in the drawn diagram, given the original decomposition.
(4) Moreover, creating a chosen decomposition by removing minimal curve labels at each step is likely to mean that fewer zones will need to be added when producing a splitting super-decomposition since $in_i$ is small.

To illustrate, drawing the abstraction $\{\emptyset, ab, ac, b\}$ yields the lefthand diagram in figure 9 by first drawing the curve $a$, then $b$ and finally $c$; the order of curve label removal to create a chosen decomposition would, therefore, be given by $c \rightarrow b \rightarrow a$. However, we could have produced a different decomposition by not removing the minimal curve label $c$ before $a$. For instance, the (not chosen) decomposition arising from removing curve labels in the order $a \rightarrow c \rightarrow b$ would have resulted in the diagram $d_1$ in figure 8 where contour $c$ is not contained by contour $a$, relating to (1) above. The diagram $d_1$ also demonstrates (2), since the contour $a$ consists of two curves whereas it only consists of one curve in figure 9.

Point (3) should be self-evident: each circle we add splits all the zones through which it passes and we add exactly the zones required so that splitting occurs. Finally, for point(4), $d_2$ in figure 8 was drawn from abstract description $\{\emptyset, ab, ac\}$ and a chosen decomposition given by curve removal order $c \rightarrow b \rightarrow a$. A (not chosen) decomposition arising from removing $a \rightarrow b \rightarrow c$ ($a$ is removed first, but is not minimal) results in $d_3$, which contains more shaded zones.

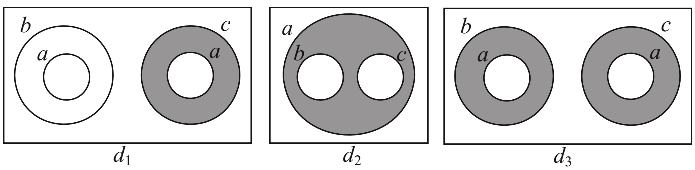



**Fig. 8.** Alternative choices

## 6 Implementation and Comparison with Other Methods

We have implemented our drawing method and the software is available for download; see www.eulerdiagrams.com. Examples drawn using our software are shown in figure 9. The lefthand diagram was drawn from abstraction $\{\emptyset, ab, ac, b\}$; when entering the abstract description into the tool, the $\emptyset$ zone is not entered and the commas are omitted. The other two diagrams were drawn from abstractions $\{\emptyset, a, ab, ac, b, bd, ef\}$ and $\{\emptyset, ab, abc, ac, ae, b, bc, bd, c, cd, d\}$ respectively, where the contour $d$ comprises two curves in the latter case. In all cases, the shaded zones were not present in the abstract description. Layout improvements are certainly possible, particularly with respect to the location of the curve labels relative to the curves and the areas of the zones. We plan to investigate the use of force directed algorithms to improve the layout.

We now include some examples of output from other implemented drawing methods, permitting their aesthetic qualities to be contrasted with the diagrams drawn using our software. Figure 10 shows an illustration of the output using the software of Flower and Howse [8], which presents techniques to draw completely wellformed diagrams, but the associated software only supports drawing up to 4 curves. The techniques of Flower and Howse [8] were extended in [9] to enhance the layout; the result of the layout improvements applied to the lefthand diagram in figure 11 can be seen on the right.

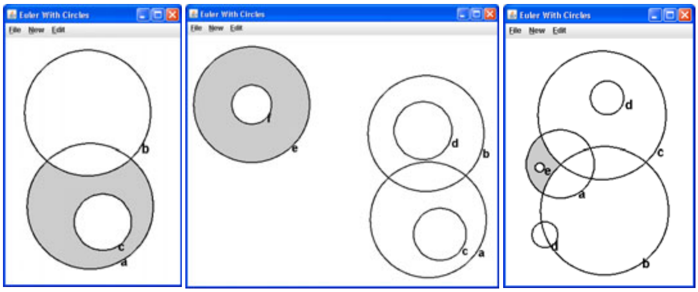



**Fig. 9.** Output from our software

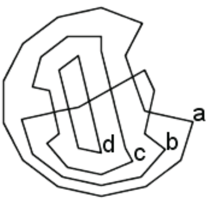



**Fig. 10.** Generation using [8]

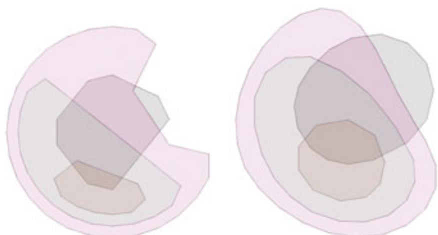


**Fig. 11.** Using layout improvement [9]

Further extensions to the methods of [8] allow the drawing of abstract descriptions that need not have a completely wellformed embedding. This was done in [13], where techniques to allow any abstract description to be drawn were developed; output from the software of [13] is in figure 12. An alternative method is developed by Simonetto and Auber [14], which is implemented in [15]. Output can be seen in figure 13, where the labels have been manually added post drawing; we thank Paolo Simonetto for this image. Most recently, an inductive generation method has been developed [18], which draws Euler diagrams by adding one curve at a time; see figure 14 for an example of the software output.

**Fig. 12.** Generation using [13]

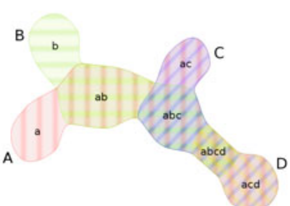



**Fig. 13.** Generation using [15]

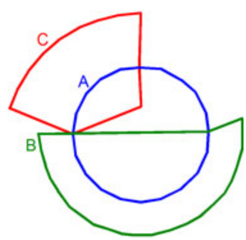



**Fig. 14.** Generation using [18]

A different method was developed by Chow [2], that relies on the intersection between all curves in the to-be-generated Euler diagram being present. We do not have access to Chow's implementation, so we refer the reader to `http://apollo.cs.uvic.ca/euler/DrawEuler/index.html` for images of automatically drawn diagrams.

## 7 Conclusion

We have presented a technique that draws Euler diagrams that are completely wellformed up to labelling. The drawings use only circles as curves, which are aesthetically desirable; many manually drawn Euler diagrams employ circles which demonstrates their popularity. This is the first implemented method that can draw any abstract description using circles. Our drawings may include extra zones but we mark them as such by shading them gray. The method also takes into account aesthetic considerations as discussed in section 5.4.

Along with layout improvements, future work will involve giving more consideration as to how to choose valid partitions, since the choice of partition can impact the quality of the drawn diagram. Moreover, the zones we added to produce a splitting super-decomposition removed the need for concurrency in the diagram. We could add further zones that reduce the number of duplicate curve labels required. For instance, three zones, $z_1$, $z_2$ and $z_3$, in $in_i$ may have a valid partition $\{\{z_1, z_2\}, \{z_3\}\}$, meaning we use two circles when adding $\lambda_i$. We might be able to add a fourth zone, $z_4$, to $in_i$ where $\{\{z_1, z_2, z_3, z_4\}\}$ is a valid partition for which we are able to add a single 2-piercing curve. Finding a balance between the number of curves of which a contour consists and the number of 'extra' zones in order to obtain an effective diagram will be an interesting challenge.

**Acknowledgements.** This research is supported by EPSRC grants EP/E011160/1, EP/E010393/1, EP/H012311/1 and EP/H048480/1. We thank John Taylor for comments on this research and Jean Flower for help with software development.

## References

1. Google Charts API, `http://code.google.com/apis/chart/` (accessed August 2009)
2. Chow, S.: Generating and Drawing Area-Proportional Euler and Venn Diagrams. PhD thesis, University of Victoria (2007)
3. Chow, S., Rodgers, P.: Constructing area-proportional Venn and Euler diagrams with three circles. In: Euler Diagrams 2005 (2005)
4. DeChiara, R., Erra, U., Scarano, V.: VennFS: A Venn diagram file manager. In: Information Visualisation, pp. 120–126. IEEE, Los Alamitos (2003)
5. Edwards, A.: Venn diagrams for many sets. New Scientist 7, 51–56 (1989)
6. Farfel, M., et al.: An overview of 9/11 experiences and respiratory and mental health conditions among world trade center health registry enrollees. Journal of Urban Health 85(6), 880–909 (2008)

7. Farrell, G., Sousa, W.: Repeat victimization and hot spots: The overlap and its implication for crime control and problem-oriented policing. Crime Prevention Studies 12, 221–240 (2001)
8. Flower, J., Howse, J.: Generating Euler diagrams. In: Hegarty, M., Meyer, B., Narayanan, N.H. (eds.) Diagrams 2002. LNCS (LNAI), vol. 2317, pp. 61–75. Springer, Heidelberg (2002)
9. Flower, J., Rodgers, P., Mutton, P.: Layout metrics for Euler diagrams. In: Information Visualisation, pp. 272–280. IEEE, Los Alamitos (2003)
10. Ip, E.: Visualizing multiple regression. Journal of Statistics Education 9(1) (2001)
11. Kestler, H., Muller, A., Kraus, J., Buchholz, M., Gress, T., Liu, H., Kane, D., Zeeberg, B., Weinstein, J.: Vennmaster: Area-proportional Euler diagrams for functional GO analysis of microarrays. BMC Bioinformatics 9(67) (2008)
12. Kestler, H., Muller, A., Liu, H., Kane, D., Zeeberg, B., Weinstein, J.: Euler diagrams for visualizing annotated gene expression data. In: Euler Diagrams 2005 (2005)
13. Rodgers, P., Zhang, L., Fish, A.: General Euler diagram generation. In: Stapleton, G., Howse, J., Lee, J. (eds.) Diagrams 2008. LNCS (LNAI), vol. 5223, pp. 13–27. Springer, Heidelberg (2008)
14. Simonetto, P., Auber, D.: An heuristic for the construction of intersection graphs. In: Information Visualisation. IEEE, Los Alamitos (2009)
15. Simonetto, P., Auber, D., Archambault, D.: Fully automatic visualisation of overlapping sets. Computer Graphics Forum 28(3) (2009)
16. Soriano, J., Davis, K., Coleman, B., Visick, G., Mannino, D., Pride, N.: The proportional Venn diagram of obstructive lung disease. Chest 124, 474–481 (2003)
17. Stapleton, G., Rodgers, P., Howse, J., Taylor, J.: Properties of Euler diagrams. In: Layout of Software Engineering Diagrams, EASST, pp. 2–16 (2007)
18. Stapleton, G., Rodgers, P., Howse, J., Zhang, L.: Inductively generating Euler diagrams. IEEE Trans. on Visualization and Computer Graphics (2009) (accepted)
19. Stapleton, G., Zhang, L., Howse, J., Rodgers, P.: Drawing Euler diagrams with circles: The theory of piercings. IEEE Trans. on Visualisation and Computer Graphics (2010) (accepted)
20. Thièvre, J., Viaud, M., Verroust-Blondet, A.: Using Euler diagrams in traditional library environments. In: Euler Diagrams 2004. ENTCS, vol. 134, pp. 189–202 (2005)
21. Venn, J.: On the diagrammatic and mechanical representation of propositions and reasonings. The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science (1880)
22. Verroust, A., Viaud, M.-L.: Ensuring the drawability of Euler diagrams for up to eight sets. In: Blackwell, A.F., Marriott, K., Shimojima, A. (eds.) Diagrams 2004. LNCS (LNAI), vol. 2980, pp. 128–141. Springer, Heidelberg (2004)
23. Wilkinson, L.: VennEuler package for R (October 2009)

# Coloured Euler Diagrams: A Tool for Visualizing Dynamic Systems and Structured Information

Paolo Bottoni[1] and Andrew Fish[2,⋆]

[1] Department of Computer Science, "Sapienza" University of Rome, Italy
bottoni@di.uniroma1.it
[2] School of Computing, Mathematical and Information Sciences, University of Brighton, UK
Andrew.Fish@brighton.ac.uk

**Abstract.** Euler diagrams are a popular method for visualising sets and their relationships, exploited for resource management and visual logic specification, for example. We add the notion of colouring, provide a formal description of the extended system and demonstrate how coloured Euler diagrams provide adequate visualisations for concepts in the new bio-inspired model of Reaction Systems and for polyarchies, visualising multiple intersecting hierarchies.

## 1 Introduction

Venn and Euler diagrams were first introduced as an aid in syllogistic reasoning, but in recent times they have been utilised in various application domains, e.g. to represent genetic set relations [8], for file system management [3], or to represent the size of library database query results [15]. They can be viewed as a logic system in their own right, and incorporated into heterogeneous reasoning systems [14], or used to search for minimal proofs [13]. The various definitions of Venn and Euler diagrams (or Euler-like diagrams) in the literature slightly differ in syntax and semantics. We propose to extend the definition of an Euler diagram system to admit a notion of colouring, to which application-dependent semantics can be associated, in contrast with usage of colour as a secondary notation to highlight some information, without including it in the formal system.

Extending the formal system in this manner enables the precise use of these diagrams as visualisations incorporating information within this previously unutilised graphical dimension rather than via some other means that may lead to an increase in complexity of the notation or to more cluttered diagrams. We emphasize, however, that we use the term colouring in a mathematical sense of an assignation of numbers (with distinct numbers corresponding to distinct colours in the traditional sense), and whilst this notion maps in the natural manner to the use of actual colours on the diagram, this more abstract concept could in fact be represented graphically by different means if the situation required it. For instance, for colour-blind users, one could utilise an alternative graphical (or

⋆ Partially funded by UK EPSRC EP/E011160: Visualisation with Euler Diagrams.

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 39–53, 2010.

concrete) representation making use of varying degrees of "dashned-ness" rather than colours such as red, green and blue, whilst utilising the same abstract model (and the same terminology of colouring at the abstract level). Moreover, we exploit vectors of colours, so that different forms of colouring can be used to visualise different aspects of the system (e.g. border color for identifying a database source, border thickness for expressing degree of relevance, and texture for identifying a keyword, in the presentation of the results of an OR query from federated databases, searching for documents presenting certain keywords).

The extension of the formal Euler diagram system to incorporate colouring has merit in that it is of theoretical interest to investigate ways of incorporating information within the diagrammatic notation, whether we are considering their use for simple data visualisation or perhaps more complicated uses such as the basis of diagrammatic logics. Their utility in the case of representing information that changes (e.g. set-based data that changes over time or diagrammatic logic statements that differ by diagrammatic logical inference rule application) is an avenue in which their usage may well bring user benefits, utilising colouring to indicate important changes between diagrams. The formalisation facilitates the use of the colouring concept within automated software systems.

We demonstrate the utility of coloured Euler diagrams as a representation system for both dynamical and static aspects within two different application domains. Firstly, we illustrate their usage for visualising notions such as modules along sequences of events, from Reaction Systems [4], a new bio-inspired model of computing where transformations rewrite the complete state of the system (so there is no resource counting). Secondly, we indicate the relationship of coloured Euler diagrams with hierarchies and with polyarchies [10], a representation of multiple intersecting hierarchies, utilised for categorisation in several domains.

We give intuition and motivations for Coloured Euler Diagrams (CEDs) in Section 2, and provide terminology and background definitions in Section 3 and a formal definition in Section 4. Section 5 gives an overview of Reaction Systems (*rs*) and discusses the visualisation of some *rs* features. In Section 6, we demonstrate their use in the domain of categorisation, comparing with visualisations such as Polyarchies which represent multiple intersecting hierarchies. Conclusions in Section 7 indicate other application areas for future investigation.

## 2 Motivation

Euler diagrams are a method of visually depicting a family of sets and their relationships. However, in many application domains, it is necessary to simultaneously present multiple families of sets. As an example, in an information system on the organisation of a multinational enterprise, one may categorise the personnel according to several dimensions, e.g. role, position in the organisational hierarchy, place of work. While each of these categories can be presented using Euler curves, there is no salient visual difference to support projection onto one of the categories (i.e. to distinguish the category type of the curves).

We explore here the visualisation of polyarchies, multiple overlapping hierarchies, each viewed as a family of sets. We propose to colour sets with a distinct colour for each hierarchy, with shared nodes coloured with a blend of the colours. In this way, one can exploit the spatial features of Euler diagrams, without having to resort to the use of multiple curve labels to indicate multiple categories.

Similarly, for dynamic structures, one wants to follow the individual evolution of families of phenomena, while maintaining a representation of the evolution of their relations. For example, for Reaction Systems discussed in this paper, interesting evolutions are those of states and of particular subsets, called modules.

Independent of the domain, two set-theoretic relationships related to this visualisation problem are those of *embedding* and *separation*, as defined in [4]. Informally, given a family of sets $\mathcal{L}$, a sub-family $\mathcal{F}$ is *embedded* in another sub-family $\mathcal{G}$ if all the component sets of $\mathcal{F}$ are contained in the intersection of the sets in $\mathcal{G}$. $\mathcal{F}$ is *separated* from $\mathcal{G}$ if it is embedded in $\mathcal{G}$ and there is a set $Y \in \mathcal{L}$ such that $Y$ is contained in the intersection of all sets in $\mathcal{G}$, and the union of all sets in $\mathcal{F}$ is contained in $Y$. Providing a visual distinction between the two families may assist readers in assessing whether such relations exist.

In all of these cases, traditional visualisation through Euler diagrams does not distinguish between the families within a single diagram (unless one imposes extra labelling conventions). In our proposal, the incorporation of colouring to the Euler diagram system enables the indication of membership of a family via the use of colour, which in turn enables the representation of the set based relationships within the different families of sets as well as between these families. Furthermore, when considering dynamic information (for example variations over time in the composition of families of sets), using colouring provides a method of linking together the families of sets within a sequence of diagrams.

## 3 Set Systems and Colouring

We first recall standard notation for sets and set systems, and then define a general notion of coloured sets, subsets and set systems; the colourings can then be specialised according to the particular domain of application. We also define a method of deriving a colouring for subsets $S$ from a colouring of sets $X$.

Let $X$ be a set; then $2^X$ is the power set of $X$. A *set system* on $X$ is a pair $(X, S)$ where $S \subset 2^X$ is a set of subsets of $X$ (see [2] for instance). Let $\Delta \subset \mathbb{Z}^+$ denote the set of prime numbers and $DIV(x)$ the set of prime integer divisors of $x$ for $x \in \mathbb{Z}$. We assume that $\emptyset \in S$ for all set systems; this corresponds to the requirement that the "outside zone" is present in the diagrams, so the term set system becomes synonymous with the abstract diagram from [6].

**Definition 1.** *Let $X = \{x_1, \dots, x_n\}$ be a finite set, and let $S = \{s_1, \dots, s_k\} \subset 2^X$. Then an $X$-colouring is a function $c : X \to \mathbb{Z}$, and an $S$-colouring is a function $c' : S \to \mathbb{Z}$. Let $c$ be an $X$-colouring. Then the* natural extension to subsets *of $c$ is $c' : 2^X \to \mathbb{Z}$, given by $c'(\Sigma) = \prod_{i \in I} c(x_i)$, where $\emptyset \neq \Sigma = \{x_i : i \in I\} \subset 2^X$, for some finite index set $I$, and $c'(\emptyset) = 0$. A* prime colouring *on $X$ is an $X$-colouring such that $c(X) \subset \Delta$ and an* injective colouring *of $X$ is a colouring $c$ which is injective.*

Taking the natural extension to subsets as a product of set colours, and using distinct prime numbers for the original colourings ensures unique colourings for subsets. Colourings of subsets are *reducible* if they can be derived as products.

**Lemma 1.** *The natural extension to subsets of any injective prime colouring on* $X$ *is an injective colouring on* $2^X$.

**Definition 2.** *Let* $c$ *be an* $X$*-colouring and* $c'$ *an* $S$*-colouring. Then* $c$ extends to $c'$ *(or* $c'$ reduces to $c$*) if* $c'$ *is a restriction of the natural extension to subsets of* $c$*. Let* $c'$ *be an* $S$*-colouring. Then* $c'$ *is* reducible *if there exists an* $X$*-colouring* $c$ *such that* $c'$ *reduces to* $c$.

**Lemma 2.** *Let* $c'$ *be an* $S$*-colouring on a set* $X$*, such that* $c'(\emptyset) = 0$*, and* $\forall \Sigma \in S$*, if* $\mid \Sigma \mid = k$ *then* $c'(\Sigma)$ *is a product of* $k$ *distinct primes. Then, any injective function* $div : X \to \Delta$*, s.t.* $\forall \Sigma \in S, x_i \in \Sigma \Leftrightarrow div(x_i) \in DIV(c'(\Sigma))$ *is a prime injective* $X$*-colouring that extends to* $c'$.

*Proof (sketch).* The "if and only if condition" tells us that every subset (member of $S$) colouring is a prime number which divides the colouring of a set (member of $X$) if the set is a member of the subset. So every prime number that is used to colour a set that is a member of a subset must also divide the colouring of that subset. Since a subset is coloured by exactly $k$ distinct primes if it has $k$ members, the subset colour is precisely the product of the $k$ primes that colour the corresponding $k$ sets.

The previous two Lemmas will allow us to derive various colourings and provide some means of consistency checking later on. To allow for more than one colouring associated to a set system, we will use a vector of colouring functions.

**Definition 3.** *Let* $(X, S)$ *be a set system. A* colouring *of* $(X, S)$ *is a vector* $K = (c_1, \ldots, c_k)$ *of functions, with* $k \geq 1$*, where each* $c_i$ *is either an* $X$*-colouring or an* $S$*-colouring for* $i \in \{1, \ldots, k\}$*. A set system* $(X, S)$ *with a colouring* $K$ *is called a* coloured set system*, denoted by* $(X, S, K)$.

*Example 1.* Let $X = \{E, F, G, H\}$, $S = \{\emptyset, \{E\}, \{F\}, \{G\}, \{F, G\}, \{G, H\}\}$, let $c_1 : X \to \mathbb{Z}$ be defined by $c_1(E) = c_1(F) = c_1(G) = 0$ and $c(H) = 1$, and let $c_2 : S \to \mathbb{Z}$ be defined by $c_2(\emptyset) = c_2(\{G, H\}) = 0$, $c_2(\{E\}) = 1, c_2(\{F\}) = 2, c_2(\{G\}) = 3$ and $c_2(\{F, G\}) = 5$. Then $c_1$ is an $X$-colouring function, $c_2$ is an $S$-colouring function and $(X, S, K)$ is a coloured set system, where $K = (c_1, c_2)$.

## 4 Coloured Euler Diagrams

We provide a basic definition of Euler diagrams, in a manner similar to other works (although we specify the definition in topological terms in order to ensure precision), and introduce the new concept of Coloured Euler Diagrams.

Commonly, somewhat restrictive set-ups are adopted, typically consisting of the well-formed diagrams of [6], where a decision procedure is provided to indicate if there is a well-formed concrete diagram realising an abstract diagram and if so to produce a drawing of it. The well-formedness conditions are concrete level constraints imposed on the system, with the intention of reducing human comprehension errors, and so there is validity in trying to preserve them; the conditions for [6] are: curves are simple (i.e. no self-intersection); curves only intersect at a finite number of points (i.e. no concurrent line segments); with at most two curves meeting at any particular point and such that the curves cross transversely at that point (i.e. curves that meet must really cross); no two curves have the same label; and no region of the plane that is inside a set of curves and outside the remaining set of curve is disconnected (no split zones).

However, since we wish to consider diagrams that are general enough to allow any set system to be represented (i.e. any abstract model has a visualisation), we cannot use uniquely labelled simple closed curves, as in [6]. Instead, we use unions of simple closed curves (or more precisely, the region bounded by them) to represent the sets of a set system, with labels determining the association. We use the term *contour* for a set of curves with a label in common, agreeing with the usage in [6] when restricting to well-formed Euler diagrams.

In general, any set-up can utilise the proposed conceptual extension of colouring. For example, in [2] some well-formedness conditions are relaxed and the term Euler-like diagrams is adopted for variations such as allowing "holes". In [16] it is shown that all abstract diagrams for at most eight sets are drawable using such Euler-like diagrams. Recently, a methodology for generating general Euler diagrams has been developed [11], ensuring the production of a diagram, with a heuristic approach to repairing possible breaks in well-formedness.

Let $C$ be a simple closed curve in the plane. Then $\overline{C}$ denotes the closed region of the plane bounded by $C$ (homeomorphic to a disc), $int(\overline{C})$ the interior of $\overline{C}$, and $ext(\overline{C})$ the exterior of $\overline{C}$. We say that curves $C_1, \ldots, C_n$ are *closure-disjoint* if $\overline{C_1} \cap \ldots \cap \overline{C_n} = \emptyset$; and $C_2$ is *closure-contained in* $C_1$ if $\overline{C_1} \cap \overline{C_2} = \overline{C_2}$.

**Definition 4.** *Let $\Lambda$ be a countable alphabet of labels. Let $\mathcal{C} = \{C_1, \ldots, C_n\}$ be a family of curves in the plane with a finite number of points of intersection and $l : \mathcal{C} \to 2^{\Lambda}$ a function assigning a set of labels to each curve in $\mathcal{C}$. For $\lambda \in \Lambda$, if the set $C'_\lambda = \{C_i \in \mathcal{C} \mid \lambda \in l(C_i)\} \subset \mathcal{C}$, of the curves with $\lambda$ as a label, is non-empty then it is called the* contour *of $\lambda$. Let $\mathcal{C}' = \{C'_\lambda \mid \lambda \in \Lambda\}$ be the set of contours in $\mathcal{C}$. Let $L(C'_\lambda) = \bigcup_{C_i \in C'_\lambda} l(C_i)$, the set of labels for the curves in $C'_\lambda$, and $L(\mathcal{C}') = \bigcup_{C'_\lambda \in \mathcal{C}'} L(C'_\lambda)$. If $\{C_i \mid C_i \in C'_\lambda\}$ is closure-disjoint, for each $\lambda \in \Lambda$, then we say that $d = (\mathcal{C}, l)$ is an* Euler diagram*. The maximal connected sets of points of the plane with the curves in $\mathcal{C}$ removed are called the* minimal regions *of $d$, denoted $R(d)$. Let $\mathcal{Y} \subset \mathcal{C}'$ be a set of contours. Let $z$ be the region of the plane inside $\mathcal{Y}$ but outside $\mathcal{C}' \setminus \mathcal{Y}$. If $z$ is nonempty then it is called a* zone*:*

$$z = \bigcap_{c \in \mathcal{Y}} int\,(\overline{c}) \;\cap \bigcap_{c \in \mathcal{C}' \setminus \mathcal{Y}} ext\,(\overline{c}) \neq \emptyset.$$

*The set of zones of $d$ is denoted by $Z = Z(d)$.*

In terms of the interpretation of the diagrams, as usual we have that if two regions determined by two contours do not overlap then the sets represented are

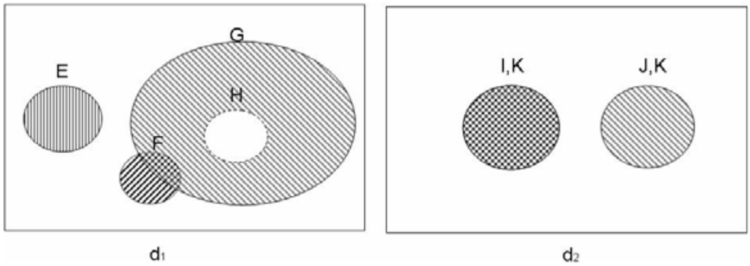



**Fig. 1.** Examples of coloured Euler diagrams. $d_1$ demonstrates both curve and region colouring; $d_2$ demonstrates the use of multiple labels and curves for set representation.

disjoint, whilst if the region of one contour $E_1$ is wholly contained in the region of another contour $E_2$ then we have the subset relationship $E_1 \subset E_2$.

*Example 2.* Figure 1 shows two (coloured) Euler diagrams $d_1$ and $d_2$, where we have drawn a (rectangular) bounding box around each diagram and labelled the diagrams in order to distinguish where one diagram ends and the other begins. In $d_1$ we have four curves, each with one label ($E$, $F$, $G$, and $H$), which are all therefore contours; in fact these curves determine a well-formed Euler diagram of [6]. There are six zones in the diagram, described by their set of containing curve labels: $\emptyset$, $\{E\}$, $\{F\}$, $\{G\}$, $\{F, G\}$, and $\{G, H\}$. This diagram realises the set system of Example 1. In terms of set-theoretic relationships we infer: $H \subset G$, $E \cap F = \emptyset$, $E \cap G = \emptyset$, and $F \cap H = \emptyset$. Diagram $d_2$, with two curves and three contours, realises the set system $(X, S)$, with $X = \{I, J, K\}$ and $S = \{\emptyset, \{I, K\}, \{J, K\}\}$. In this complex example $X$ is a set of sets and we label curves with the names of the sets, so that they have a label in common. The contours for $I$ and $J$ coincide with the curves with that label in their label set. Applying a standard interpretation of Euler diagrams, we have that the two regions contained by the two curves represent the sets $I \cap K$ and $J \cap K$. As missing zones represent the empty set, we derive $(I \cap K) \cap (J \cap K) = \emptyset$, and so the diagram supports the inference $I \cup J = K$. One can observe the trade-off between the complexity of interpretation of these diagrams with multiple curves and labels and the ability to realise any set system.

With regard to the extension to colouring we demonstrate two realisations of colouring functions in Example 2: colouring the curves themselves, or utilising textures for diagram regions. For overlapping regions, one could choose independent textures, or some form of meshing of textures from the basic regions, giving a visual indicator of which regions are overlapping, similar to that in [12]. The formal definition of an instantiation function from the abstract colouring functions to such graphical attributes is omitted here for simplicity. Before we give the definition of colouring, we provide some consistency conditions: firstly, if any two curves with the same label set have the same colour, and there is a colouring of contours that induces this colouring on curves then we have curve-contour colour consistency; secondly, if a colouring of minimal regions can be induced from a colouring on zones which in turn can be induced by a colouring of contours then we have region-contour consistency.

**Definition 5.** *Let $\mathcal{C}$ be a set of labelled curves, forming a set of contours $\mathcal{C}'$ and let $c$ be a colouring on $\mathcal{C}$ s.t. $\forall C_i, C_j \in \mathcal{C}$, we have $l(C_i) = l(C_j) \Rightarrow c(C_i) = c(C_j)$. Then $l(\mathcal{C}) \subset 2^{L(\mathcal{C}')}$ and $c$ defines a colouring $c_1 : l(\mathcal{C}) \to \mathbb{Z}$. If $c_1$ is reducible to a $L(\mathcal{C}')$-colouring then we have* curve-contour colour consistency*. Let $R$ be the set of minimal regions defined by $\mathcal{C}$, and suppose that $c''$ is a colouring of $R$. If $c''$ satisfies the properties that: (i) every minimal region of any zone $z$ has the same image under $c''$; (ii) the $Z$-colouring determined by $c''$ is reducible to a $\mathcal{C}'$ colouring, then we have* region-contour colour consistency*.*

*Remark 1.* If all contours have unique single labels, then a 1-1 correspondence exists between contour labels and contours, and since every curve with the same label set has the same colour, also between curve label sets and curves. In this case colourings of $L(\mathcal{C}')$ and $\mathcal{C}'$ are equivalent, as well as those of $l(\mathcal{C})$ and of $\mathcal{C}$.

**Definition 6.** *Let $d = (\mathcal{C}, l)$ be an Euler diagram and $R$ its set of minimal regions. A function $c : \mathcal{C} \to \mathbb{Z}$ is called a $\mathcal{C}$-colouring function and a function $c : R \to \mathbb{Z}$ is called an $R$-colouring function. A* colouring *of $d = (\mathcal{C}, l)$ is a vector $K = (c_1, \ldots, c_k)$ with $k \geq 1$, and each $c_i$ either a $\mathcal{C}$-colouring function or an $R$-colouring function. An Euler diagram $d = (\mathcal{C}, l)$ with a colouring $K$ is called a* coloured Euler diagram *(abbreviated CED), denoted $(\mathcal{C}, l, K)$, if each component of $K$ satisfies either curve-contour colour consistency or region-contour colour consistency, according to its domain being $\mathcal{C}$ or $R$, respectively.*

The consistency relationships ensure that we have colouring functions on the set of contours and zones of the diagram, which therefore give rise to colouring functions on the underlying set system that the diagram is representing.

*Example 3.* From Figure 1, in terms of colouring, $d_1$ has a 2-component colouring vector $K = (c_1, c_2)$, the same as for the set system in Example 1, where $c_1$ is realised by border colouring (with correspondences $0 \longleftrightarrow$ **solid**, and $1 \longleftrightarrow$ **dashed**) and $c_2$ is realised by region texturing. The diagram $d_2$, with two curves and three contours, is equipped with a 1-component $R$-colouring function $c_1 : R \to \mathbb{Z}$ such that $c_1(\emptyset) = 0$, $c_1(\{I, K\}) = 1$, and $c_1(\{J, K\}) = 2$, realised using textures.

We relate the notions of coloured set system and concrete coloured Euler diagram, and show that any coloured set system can be realised by a concrete coloured Euler diagram. Diagram $d_2$ from Example 2 shows an example of an application of the proof strategy. Notice that consistency amongst colouring on curves and contours is incorporated into the definition of CEDs, whilst consistency between sets and contours is incorporated in the definition of realisation.

**Definition 7.** *Let $(X, S)$ be a set system. Then an Euler diagram $d = (\mathcal{C}, l)$ is a* realisation *of $(X, S)$ if there is a bijection $b : X \leftrightarrow \mathcal{C}'$ which induces a bijection $b_s : S \leftrightarrow Z(d)$. Also, $(X, S)$ is called the* abstraction *of $d = (\mathcal{C}, l)$. Let $(d = \mathcal{C}, l, K')$ be a CED and $(X, S, K)$ a coloured set system, such that $(\mathcal{C}, l)$ is a* *realisation of* $(X, S)$. *Then we say that* $K'$ realises the colouring $K$ *of* $(X, S)$, *if* $\exists k \in \mathbb{Z}^+, \mid K \mid = \mid K' \mid = k$ *and the bijection* $b$ *respects the colouring vectors:* $\forall i \in \{1, \ldots, k\}$ $c_i(x) = y \Leftrightarrow c'_i(b(x)) = y$, *for* $x \in X$, *and* $c'_i(s) = y \Leftrightarrow c_i(b_s(s)) = y$, *for* $s \in S$ *(according to* $c_i$*'s being a contour or a region colouring). We call* $(\mathcal{C}, l, K')$ *a* realisation *of* $(X, S, K)$ *and* $(X, S, K)$ *the* abstraction *of* $(\mathcal{C}, l, K')$.

In order to realise coloured set systems, and to reduce the amount of information needed to define a consistent colouring, we show how to induce colourings on curves and regions from contours in a diagram by: colouring curves with the product of the colours of the contours the curve belongs to; inducing region colouring from contour colouring via the zone colouring obtained as a natural extension to subsets.

**Definition 8.** *Let* $d = (\mathcal{C}, l)$ *be an Euler diagram and let* $c : \mathcal{C}'(d) \to \mathbb{Z}$ *be a colouring of the contour set of* $d$. *Then the* induced curve colouring $c_c : \mathcal{C}(d) \to \mathbb{Z}$ *is given by:* $c_c(C) = \prod_{\{C' \in \mathcal{C}'(d) | C \in C'\}} c(C')$ *for* $C \in \mathcal{C}(d)$, *whilst the* induced region colouring $c_r : \mathcal{R}(d) \to \mathbb{Z}$ *is obtained from the extension of* $c$ *to zones by setting every minimal region's colour to be the colour of the zone containing it.*

*Remark 2.* In the following proof, and later on, we make use of the important notion of induced colourings: a set colouring (i.e an $X$-colouring) gives rise to a colouring of contours from which we can induce a colouring of curves, or in fact a colouring of zones, and hence minimal regions. If all of the colourings of a diagram are induced from contour colourings, then this ensures the consistency of all relations required for the diagram to be a CED; i.e. an Euler diagram $d = (\mathcal{C}, l)$ together with a vector of induced colouring functions is a CED.

**Theorem 1**

*(i) Let* $d = (\mathcal{C}, l)$ *be an Euler diagram realising a set system* $(X, S)$ *and let* $c$ *be an* $X$*-colouring. Then* $c$ *gives rise to a colouring of contours, via the bijection of the realisation, which induces a curve or region colouring of* $d$.

*(ii)Any coloured set system* $(X, S, K)$ *can be realised as a CED* $d = (\mathcal{C}, l, K)$.

*Proof.* The first part follows from definitions. For the second part, we first show that any set system can be realised. For every $x \in S \setminus \emptyset$, take one curve[1] labelled by $x$. This yields a Euler diagram $d = (\mathcal{C}, l)$ with the correct zone set. Map the $S$-colourings of $(X, S)$ to $R$-colourings of $(\mathcal{C}, l)$, using the $Z$-colouring determined by $S$ and the same colour for the set of minimal regions comprising a zone. Map the $X$-colourings to colourings of $\mathcal{C}'$, the contours of $d$, updating the colourings of curves with multiple labels (i.e. take the induced colouring on curves).

For families of sets depicted in a sequence of diagrams, as a sequence of contours, we define a contour sequence, tracking a set through a sequence of diagrams, and a colouring respects this contour sequence if it is consistent across the sequence.

[1] The construction is uniform whether $S$ is a set of distinct elements, or a set of sets. In the first case we consider each element as a singleton containing exactly it.

Extending to a family of these sequences gives rise to the $\mathcal{Y}^*$-respectful sequences, covering a sequence of diagrams if every contour appears in the $\mathcal{Y}^*$-respectful sequence.

**Definition 9.** *Let $\mathcal{D} = (d_1, \ldots, d_n)$, with $d_h = (\mathcal{C}_h, l_h)$, be a sequence of Euler diagrams. For each $d_h$, $\mathcal{C}'_h$ is its contour set and $(X_h, S_h)$ its set system abstraction. Let $\mathcal{Y} = (Y_1, \ldots, Y_n)$ be a sequence of elements*[2]*, with $Y_i \in X_i$, for $i \in \{1, \ldots, n\}$. Then $C'(\mathcal{Y})$, the* contour sequence *of $\mathcal{Y}$, is the sequence $(C'_{Y_1}, \ldots, C'_{Y_n})$ of contours in $\mathcal{D}$ s.t. $C'_{Y_i}$ corresponds to $Y_i$ under the bijection of the realisation. Let $\mathcal{C}'(\mathcal{Y})$ denote the set of contours in the sequence. Let each $d_h$ be coloured by a colouring $K^h$. If $C'_{Y_j}, C'_{Y_k} \in \mathcal{C}'(\mathcal{Y}) \Rightarrow K^j(C'_{Y_j}) = K^k(C'_{Y_k})$, then we say that the colourings* respect the contour-sequence *$C'(\mathcal{Y})$. Let $\mathcal{Y}^* = \{\mathcal{Y}_1, \ldots, \mathcal{Y}_k\}$ be a collection of $k$ contour sequences. If the colourings respect the contour sequence of $\mathcal{Y}_i$ for all $i \in \{1, \ldots, k\}$, then we say that the sequence of CEDs $\mathcal{D}' = (d'_1, \ldots, d'_n)$, with $d'_h = (\mathcal{C}_h, l_h, K^h)$, is a* $\mathcal{Y}^*$-respectful CED sequence. *If $\bigcup_{j=1}^{k} \mathcal{C}'(\mathcal{Y}_j) = \bigcup_{i=1}^{n} \mathcal{C}'(d_i)$ then we say that $\mathcal{Y}^*$* covers *$\mathcal{D}$.*

**Theorem 2.** *Let $\mathcal{D} = (d_1, \ldots, d_n)$, with $d_h = (\mathcal{C}_h, l_h)$, be a sequence of Euler diagrams, such that $(X_h, S_h)$ is the set system abstraction of $d_h$. Let $\mathcal{F}^* = \{\mathcal{F}_1, \ldots, \mathcal{F}_k\}$ be a collection of families of sets, such that each $\mathcal{F}_i$ defines a contour sequence of $\mathcal{D}$, and $\mathcal{F}^*$ covers $\mathcal{D}$. Then there is a colouring function that makes $\mathcal{D} = (d_1, \ldots, d_n)$ into a* $\mathcal{F}^*$-respectful CED sequence.

*Proof.* Assign a unique prime colour to each family $\mathcal{F}_i$, thereby assigning a colour to the contour sequence for that family. Then take the colourings in each diagram $d_h$ of the CED sequence to be those induced from the contour colourings. This yields an $\mathcal{F}^*$-respectful CED sequence, as required.

## 5 Representing Reaction Systems with CEDs

In this Section we propose the use of CEDs to represent some significant notions from the recently proposed computational model of Reaction Systems [5,4]. A Reaction System (*rs*) is an ordered pair $\mathcal{A} = (X, A)$, where $X$ is a finite (background) set, and $A$ is a finite set of reactions of the form $a = (R_a, I_a, P_a)$, s.t. $R_a, I_a, P_a \subseteq X$ for each $a \in A$. $R_a$ is called the set of reactants, $I_a$ of inhibitors, and $P_a$ of products. A reaction $a$ is enabled in a state $T$ if $R_a \subseteq T$ and $I_a \cap T = \emptyset$. If $a$ is enabled in $T$, then $P_a$ is produced as the new state, replacing $T$, denoted by $res_{\mathcal{A}}(T) = P_a$, or $T \xrightarrow{\mathcal{A}} P_a$; otherwise the result of applying $a$ to $T$ is the empty set. Note that $a$ can be enabled only if $R_a \cap I_a = \emptyset$.

Figure 2 shows an informal graphical representation of a set of reaction rules on background set $X = \{1, 2, 3, 4, 5\}$, as a respectful CED sequence for the family $\{R, I, P\}$, augmented with information of set membership. Here we provide a legend on the left of the figures indicating the colour coding.

[2] Note that the $X_h$ are the set of elements of the set system, but these elements $Y_i$ are themselves sets in the applications and we consider them as such here. Of course, there is a natural map from elements to singleton sets that can be utilised if required.

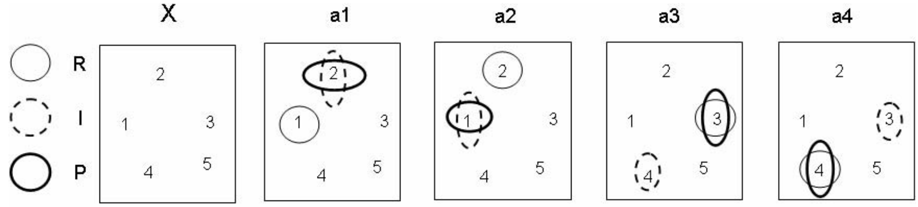



**Fig. 2.** Four reaction rules $a_1, \ldots, a_4$ on a background set $X$. For example, as $R_{a_1} = \{1\}, I_{a_1} = \{2\}$ and $P_{a_1} = \{2\}$, in a state containing 1 but not 2, rule $a_1$ produces 2.

These notions are immediately extended to sets: given a set $A$, $en_A(T) \subset A$ is the set of reactions from $A$ enabled in $T$. The effect of applying $A$ to $T$ is then the cumulative effect $res_{\mathcal{A}}(T) = \bigcup_{a \in en_A(T)} P_a$. For simplicity, we consider here only self-sustaining processes, i.e. without any contribution from the environment. Hence, we identify the evolution of the process with the sequence of states produced by the *rs*. A process $\pi$ is characterised by a finite sequence $sts(\pi) = (W_0, \ldots, W_n)$, s.t. for $0 < i \leq n$, $W_i \subseteq X, W_i = res_{\mathcal{A}}(W_{i-1})$.

An *extended reaction system* (or *ers*) is an ordered pair $\mathcal{E} = (\mathcal{A}, R)$, where $\mathcal{A} = (X, A)$ is an *rs*, and $R \subset 2^X \times 2^X$. $R$ is a restriction relation s.t. in the sequence $(W_0, \ldots, W_n)$ each pair $(W_i, W_{i+1}) \in R$. $\mathcal{E}$ is also denoted $(X, A, R)$.

Let $\mathcal{E} = (X, A, R)$ be an *ers*, $\mathcal{W} = (W_0, \ldots, W_n)$ a finite sequence of states for some process in $\mathcal{E}$, and $\omega = (Q_i, \ldots, Q_j)$ a sequence of sets s.t. $Q_i \subset W_i, \ldots, Q_j \subset W_j$ and $Q_i = res_{\mathcal{A}}(Q_{i-1})$, for $0 < i \leq j \leq n$ (see Figure 3). Under some technical conditions (see [4]), $\omega$ is called an *event* and each $Q_i$ a *module* of $\omega$ in $W_i$. A *snapshot* $\mathcal{S}_k$ of $\omega$ at step $k$ is the set of modules of $W_k$.

Respectful CED sequences with induced colourings provide an immediate representation of modules. We state here the required properties of the sequences, and claim that explicit constructions exist (not shown here due to lack of space), using prime injective colourings and the notion of inducing to generate them.

**Definition 10.** *Let $\omega = (Q_i, \ldots, Q_j)$ be an event of $\mathcal{E}$ with $\mathcal{W}$ the corresponding sequence of states. Let $\mathcal{D} = (d_0, \ldots, d_{j-i})$ be a finite sequence of CEDs, s.t. each $d_h$, with $0 \leq h \leq j - i$, is a realisation of $(\{W_{i+h}, Q_{i+h}\}, T, K^h)$ for some colouring $K^h$, $T \subset 2^{\{W_{i+h}, Q_{i+h}\}}$, and the curves for the contours $Q'_{i+h}$ are closure-contained in the curves for the contours $W'_{i+h}$, with curves identified if the corresponding sets are equal. We call $\mathcal{D}$ a* representation *of $\omega$ if the two resulting contour sequences $\mathcal{W}', \mathcal{Q}' \in \mathcal{C}'(\mathcal{D})$ are such that $\mathcal{Y}^* = \{\mathcal{W}', \mathcal{Q}'\}$ covers $\mathcal{D}$, and $\mathcal{D}$ is a $\mathcal{Y}^*$-respectful CED sequence.*

*Remark 3.* Here, and in the following, when constructing CED sequences, the curve and region colourings are induced from the contour colourings. So, if some of the $Q_i$ and $W_i$ are equal (so their curves coincide) the curve colouring used is obtained from the contour colourings (taking the product of the colours of the contours). We also use different types of colouring (curve versus region) to indicate the different types of families (states versus modules).

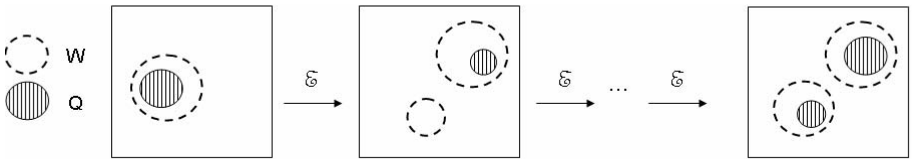



**Fig. 3.** A graphical representation of states ($W_i$) and modules ($Q_i$) using CEDs; here each subsequent diagram represents a single evolution of the system

Events can merge in one state and must remain merged thereafter. Let $\omega$ and $\omega^1$ be two events of $\mathcal{E}$. We say that $\omega$ *merges* $\omega^1$ in $W_i$ if $Q_i$ is a module of $\omega$ in $W_i$, $Q_i^1$ a module of $\omega^1$ in $W_i$ and $Q_i \subseteq Q_i^1$. Then $\omega$ and $\omega^1$ remain merged as both processes evolve through $W_{i+1}, W_{i+2}, \ldots, W_j$, i.e. $Q_k \subset Q_k^1$ for all $k \in \{i, \ldots, j\}$. We provide a CED characterisation of merging as follows, from which the representation in Figure 4 derives.

**Definition 11.** *Let $\omega$, $\mathcal{D}$, $\mathcal{W}$, $\mathcal{Q}$, $\mathcal{W}'$, $\mathcal{Q}'$ be as in Definition 10 and $\omega^1$ another event on the same $\mathcal{W}$, with a corresponding family of modules $\mathcal{Q}^1$. Then we say that a sequence of diagrams $\mathcal{D}^2$ is a* representation for the merging *of $\omega$ and $\omega^1$ iff: 1) there exists a sequence of diagrams $\mathcal{D}^1$ which is a representation of $\omega^1$ s.t. the families of contours $\mathcal{W}'$ and $\mathcal{W}'^1$ coincide in $\mathcal{D}$ and $\mathcal{D}^1$, with distinct colourings for contours in $\mathcal{Q}'$ and $\mathcal{Q}'^1$ and same colourings for $\mathcal{W}$. 2) There exists an operation $*$ which combines pairs of the diagrams, in sequential order, preserving contours and contour colourings, and which merges curves which represent the same set, s.t. $\mathcal{D}^2 = \mathcal{D} * \mathcal{D}^1$. 3) $\mathcal{D}^2$ is a respectful sequence of diagrams for $\mathcal{Y}^* = \{\mathcal{W}', \mathcal{Q}', \mathcal{Q}'^1\}$. 4) $\mathcal{Y}^*$ covers $\mathcal{D}^2$.*

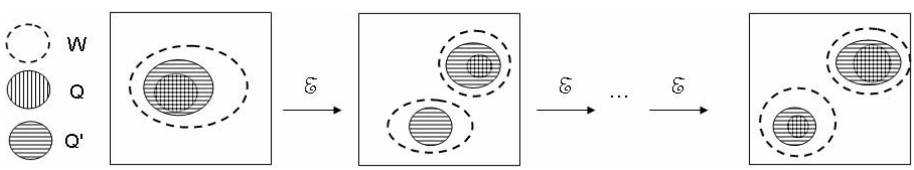



**Fig. 4.** A representation of the merging of events showing two sequences of modules $Q_i$ and $Q_i'$ with the same set of states $W_i$

A further refinement considers a set $Q$ which is a module for different states $W_k, W_l^1$ from two state sequences $\tau$ and $\tau^1$. If $W_k \subseteq W_l^1$, then the successor of $Q$ in $\tau^1$ (i.e. the module obtained from $Q$ by the application of the set of reactions) is a subset of the successor of $Q$ in $\tau$. We define a colouring scheme for combining diagrams which represent different state sequences in the same *ers*.

**Definition 12.** *Let $\omega$, $\omega_1$ be two events from sequences of states $\mathcal{W}$ and $\mathcal{W}^1$, with families of modules $\mathcal{Q}$ and $\mathcal{Q}^1$. A sequence $\mathcal{D}^2$ is a* representation for common events *iff: 1) There exist two sequences of CEDs $\mathcal{D}$ and $\mathcal{D}^1$ which are* *representations for $\omega$ and $\omega_1$, respectively. 2) All colourings for contours in $\mathcal{Q}'$, $\mathcal{Q}'^1$, $\mathcal{W}'$, $\mathcal{W}'^1$ are distinct. 3) $\mathcal{D}^2 = \mathcal{D} * \mathcal{D}^1$, with $*$ as in Definition 11. 4) $\mathcal{D}^2$ is a respectful CED sequence for $\mathcal{Y}^* = \{\mathcal{W}', \mathcal{W}'^1, \mathcal{Q}', \mathcal{Q}'^1\}$. 5) $\mathcal{Y}^*$ covers $\mathcal{D}^2$*

In Figure 5, curve colouring (dashed versus non-dashed) distinguishes the $W_i$ states and texture (horizontal vs. vertical lines) distinguishes modules. The colour for the common module is different from those of the states and of the other modules, and is shown by combining the colourings for the different events.

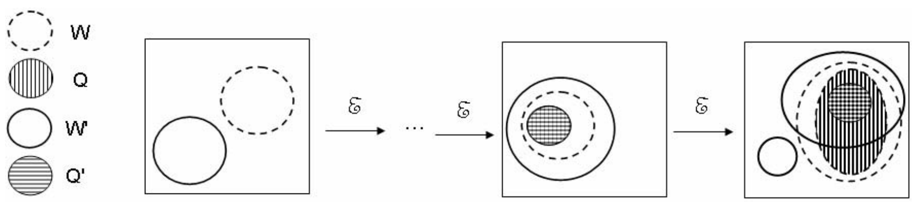



**Fig. 5.** Representing common events: $Q$ and $Q'$ are modules for $W$ and $W'$ resp.

In [4], it is shown that each snapshot in a sequence of snapshots is a partial order, with top and bottom elements. A fundamental property of snapshots is that, given two consecutive snapshots $\mathcal{S}_k, \mathcal{S}_{k+1}$ and $\mathcal{F}, \mathcal{G}$ two families of subsets in $\mathcal{S}_k$, if $\mathcal{F}$ is embedded in $\mathcal{G}$, and each module in $\mathcal{F}$ and $\mathcal{G}$ has a successor in $\mathcal{S}_{k+1}$, then $\mathcal{F}'$ is separated from $\mathcal{G}'$ in $\mathcal{S}_{k+1}$, as shown in Figure 6.

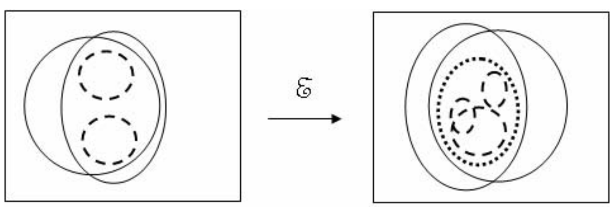



**Fig. 6.** From embedding to separation. Left: $S_k$ with $\mathcal{F}$, dashed, embedded in G, bold. Right: $S_{k+1}$ with $\mathcal{F}$ separated from $\mathcal{G}$ by the dotted curve.

## 6 Information Structure Visualisations

We describe some information structure visualisations and indicate how CEDs could be used in their place. A basic structure is a hierarchy, modeled as a rooted, directed tree. Viewing how a single hierarchy changes over time is an important task in some domains (e.g. the TimeTube visualization uses Disk Trees to represent the evolution of the web ecology [1]). Placing hierarchies next to each other and connecting common nodes with lines was a method used to visualise taxonomies in [7]. Multiple intersecting hierarchies that share common nodes have been called Polyarchies [10]. Visualization techniques include the visual pivot, which shows how hierarchies relate to each other in the context of various entities: the specified entities are highlighted, the new hierarchy appears and rotates into place and the old hierarchy rotates out of place and fades away, leaving the new hierarchy, with the user being shown the transition. A stacked link view is also adopted, showing multiple hierarchies with lines linking the common nodes (an example is shown on the left of Figure 7). This enables a similar animation for viewing several related hierarchies, which preserves the previous view. In [9], Polyarchies are viewed as edge coloured multigraphs, effectively considering them as the union of rooted trees, each corresponding to a single hierarchy. CEDs can be used to represent hierarchies: a stacked link view can be represented as a sequence of CEDs with colour indicating commonality arising from the common nodes of the hierarchies; this then extends to a visualisation using a CED sequence instead of a Polyarchy in stacked link view.

**Definition 13.** *Let $T = (V, E, r)$ be a tree, with vertex set $V$ and directed edge set $E$, rooted at $r \in V$. $E^t$ is the transitive closure of $E$. Let $d_T$ be an Euler diagram with contour set $\mathcal{C}'$ and one curve for each contour, s.t. there is a bijection $b : V \to \mathcal{C}'$ for which $(v_1, v_2) \in E^t$ iff the curve for $b(v_1)$ is contained in the interior of the curve for $b(v_2)$. Then $d_T$, together with any $C$-colouring function $c$ assigning the same value to all curves, is a* coloured Euler diagram of $T$.

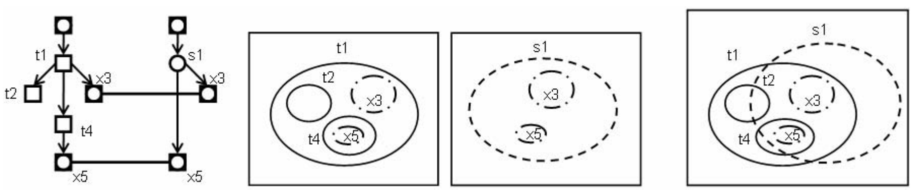



**Fig. 7.** Left: A polyarchy, consisting of two hierarchies $T_1$ and $T_2$ joined by lines indicating nodes in common (i.e. this is a stacked link view of $T_1$ and $T_2$). Middle: A pair of coloured Euler diagrams $d_1$ and $d_2$, each one representing the corresponding hierarchy with the colour (dotted curves) indicating the nodes in common. Right: A single combined diagram $d_3$ representing the polyarchy.

**Definition 14.** *Let $T_1$ and $T_2$ be rooted trees with a set of vertices $V = V(T_1) \cap V(T_2)$ in common. Then $G(T_1, T_2)$, the* stacked link view *of $T_1$ and $T_2$, is the partially directed graph $T_1 \sqcup T_2$ together with an undirected edge set between the pairs of vertices in $T_1$ and $T_2$ in common. A* visualisation of the stacked link view *of $T_1$ and $T_2$ is a CED sequence $d_1, d_2$ such that $d_1$ and $d_2$ are coloured Euler diagrams for $T_1$ and $T_2$ respectively, using one same colour $c$ for the nodes in $V$, and not using $c$ for any node in the symmetrical difference $(V(T_1) \cup V(T_2)) \setminus V$.*

**Lemma 3.** *Let $d_{T_1}$ and $d_{T_2}$ be the CEDs of $T_1$ and $T_2$ respectively, with the $T_i$ as in Definition 14 Let nodes in $T_1$ be coloured with $a_1$ and those in $T_2$ with $a_2$, $a_1$ and $a_2$ being distinct prime integers. Let $d_1$ be $d_{T_1}$ with colouring altered so* *that the colour of curves of vertices in* $V$ *is* $a_1 * a_2$ *whilst those not in* $V$ *are coloured* $a_1$*; similarly,* $d_2$ *is* $d_{T_2}$ *except that the colour of curves of vertices in* $V$ *is* $a_1 * a_2$*. Then the coloured CED sequence* $d_G = (d_1, d_2)$ *of* $G(T_1, T_2)$ *is a visualisation of the stacked link view of* $T_1$ *and* $T_2$*.*

*Example 4.* Figure 7 shows a polyarchy consisting of two hierarchies $T_1$ and $T_2$ joined by lines indicating shared nodes. We have augmented the usual representation of trees with a form of colouring, using shapes [square or circle] to represent nodes from each hierarchy, and an overlapping square and circle to represent nodes in common. The respectful CED sequence which is the visualisation of the stacked link view of $T_1$ and $T_2$ is shown in the middle of the figure as $d_1$ and $d_2$. The right hand side of Figure 7 shows $d_3$, an example of the combination of the two diagrams $d_1$ and $d_2$, which offers a potential opportunity to reduce the length of diagram sequences, when it is possible.

**Theorem 3.** *Let* $P$ *be any polyarchy consisting of multiple trees in a stacked link view. Then there is a coloured Euler diagram sequence visualising* $P$*.*

*Proof (sketch).* Let $T_1, \ldots, T_n$ denote the $n$ hierarchies of the polyarchy. For each $T_i$ define an ED as usual. Assign a unique prime colour $p_i$ to each $T_i$. For a node $n$ common to exactly the hierarchies $T_{n_1}, \ldots, T_{n_j}$, assign to $n$ the colour $p_{n_1} * \ldots * p_{n_j}$. Then each node of any $T_i$ corresponds to a contour of the CED, and if the node is shared amongst hierarchies, the corresponding contours in each diagram have the same colour.

## 7 Conclusion

We have extended the notion of Euler diagrams to incorporate colourings, providing extra dimensions for representing various domain features, and demonstrated their applicability as visualisations of concepts in two different domains: Reaction Systems and Information Structure visualisations. We draw representations of coloured Euler diagrams where the colouring functions are instantiated by graphical attributes such as actual colour, texture, dashedness, etc, and the choice of such instantiations can be important in terms of human perception and understanding, but they have a common encoding in a mathematical sense.

Colouring functions can be applied to alternative Euler based diagram systems. Colouring is applied to set systems and diagrams and we have shown that any coloured set system can be realised as a coloured Euler diagram using the set-up in this paper. Also, we have presented methods to induce colourings of curves and regions from a given specification of colourings on sets. One extra benefit of this is to reduce the amount of information required for such colouring specifications which may prove useful in implemented systems.

Possible directions for future work include: integration of colouring in logical systems, either to assist human readers in comprehension of logical steps involving conjunction of two diagrams, or to allow representation of attributes; extending the representation devised for polyarchies to representation of hypergraphs; representation of clusters, highlighting their changes over time along a sequence of diagrams. On the theoretical side, we plan to extend the proposal with notions of diagram combination and matching.

## References


1. Chi, E.H., Pitkow, J., Mackinlay, J., Pirolli, P., Gossweiler, R., Card, S.K.: Visualizing the evolution of web ecologies. In: Proc. CHI 1998, pp. 400–407. ACM Press/Addison-Wesley (1998)
2. Chow, S.C.: Generating and Drawing Area-Proportional Euler and Venn Diagrams. PhD thesis, University of Victoria (2007)
3. De Chiara, R., Erra, U., Scarano, V.: VennFS: A Venn diagram file manager. In: Proc. InfoVis 2003, pp. 120–126. IEEE Computer Society Press, Los Alamitos (2003)
4. Ehrenfeucht, A., Rozenberg, G.: Events and modules in reaction systems. Theoretical Computer Science 376(1-2), 3–16 (2007)
5. Ehrenfeucht, A., Rozenberg, G.: Reaction systems. Fundamenta Informaticae 75(1-4), 263–280 (2007)
6. Flower, J., Fish, A., Howse, J.: Euler diagram generation. Journal of Visual Languages and Computing 19, 675–694 (2008)
7. Graham, M., Kennedy, J.: Extending taxonomic visualisation to incorporate synonymy and structural markers. Inform. Visualiz. 4(3), 206–223 (2005)
8. Kestler, H., Muller, A., Gress, T., Buchholz, M.: Generalized Venn diagrams: A new method for visualizing complex genetic set relations. Journal of Bioinformatics 21(8), 1592–1595 (2005)
9. McGuffin, M.J., Schraefel, M.C.: A comparison of hyperstructures: Zzstructures, mspaces, and polyarchies. In: Proc. 15th ACM Hypertext and Hypermedia, pp. 153–162. ACM Press, New York (2004)
10. Robertson, G., Cameron, K., Czerwinski, M., Robbins, D.: Polyarchy visualization: Visualizing multiple intersecting hierarchies. In: Proc. ACM SIGCHI, pp. 423–430. ACM Press, New York (2002)
11. Rodgers, P., Zhang, L., Fish, A.: General euler diagram generation. In: Stapleton, G., Howse, J., Lee, J. (eds.) Diagrams 2008. LNCS (LNAI), vol. 5223, pp. 13–27. Springer, Heidelberg (2008)
12. Simonetto, P., Auber, D., Archambaul, D.: Fully automatic visualisation of overlapping sets. Computer Graphics Forum 28, 967–974 (2009)
13. Stapleton, G., Masthoff, J., Flower, J., Fish, A., Southern, J.: Automated theorem proving in Euler diagrams systems. J. of Autom. Reas. 39(4), 431–470 (2007)
14. Swoboda, N., Allwein, G.: Using DAG transformations to verify Euler/Venn homogeneous and Euler/Venn FOL heterogeneous rules of inference. Software and System Modeling 3(2), 136–149 (2004)
15. Thièvre, J., Viaud, M., Verroust-Blondet, A.: Using euler diagrams in traditional library environments. In: Proc. Euler Diagrams 2004. ENTCS, vol. 134, pp. 189–202 (2005)
16. Verroust, A., Viaud, M.-L.: Ensuring the drawability of Euler diagrams for up to eight sets. In: Blackwell, A.F., Marriott, K., Shimojima, A. (eds.) Diagrams 2004. LNCS (LNAI), vol. 2980, pp. 128–141. Springer, Heidelberg (2004)

# Drawing Area-Proportional Venn-3 Diagrams with Convex Polygons

Peter Rodgers[1], Jean Flower[2], Gem Stapleton[3], and John Howse[3]

[1] University of Kent, Canterbury, UK
p.j.rodgers@kent.ac.uk
[2] Autodesk
[3] Visual Modelling Group, University of Brighton, UK
{g.e.stapleton,john.howse}@brighton.ac.uk

**Abstract.** Area-proportional Venn diagrams are a popular way of visualizing the relationships between data sets, where the set intersections have a specified numerical value. In these diagrams, the areas of the regions are in proportion to the given values. Venn-3, the Venn diagram consisting of three intersecting curves, has been used in many applications, including marketing, ecology and medicine. Whilst circles are widely used to draw such diagrams, most area specifications cannot be drawn in this way and, so, should only be used where an approximate solution is acceptable. However, placing different restrictions on the shape of curves may result in usable diagrams that have an exact solution, that is, where the areas of the regions are exactly in proportion to the represented data. In this paper, we explore the use of convex shapes for drawing exact area proportional Venn-3 diagrams. Convex curves reduce the visual complexity of the diagram and, as most desirable shapes (such as circles, ovals and rectangles) are convex, the work described here may lead to further drawing methods with these shapes. We describe methods for constructing convex diagrams with polygons that have four or five sides and derive results concerning which area specifications can be drawn with them. This work improves the state-of-the-art by extending the set of area specifications that can be drawn in a convex manner. We also show how, when a specification cannot be drawn in a convex manner, a non-convex drawing can be generated.

## 1 Introduction

Area-proportional Venn diagrams, where the areas of the regions formed from curves are equal to an area specification, are widely used when visualizing numerical data [6,7,8]. An example of such a diagram is shown in Figure 1, adapted from [10]. It shows the intersections of physician-diagnosed asthma, chronic bronchitis, and emphysema within patients who have obstructive lung disease. Here, the areas only approximate the required values, which are given in the regions of the diagram, as an exact solution does not exist. In fact, when using three circles to represent an area specification there is almost certainly no exact area-proportional Venn-3 diagram [3].

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 54–68, 2010.

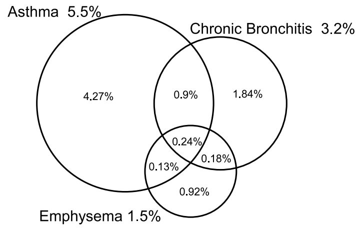



**Fig. 1.** An approximate area-proportional Venn-3 diagram

There have been various efforts towards developing drawing methods for area-proportional Venn-3 diagrams but their limitations mean that the vast majority of area specifications cannot be drawn automatically, even for this seemingly simple case. Chow and Rodgers devised a method for producing approximate Venn-3 diagrams using only circles [3], and the Google Charts API includes facilities for drawing approximate area-proportional Euler diagrams with at most three circles, including Venn-3 [1]. The approximations are often insufficiently accurate to be helpful and can be misleading. Recent work, which is limited and not practically applicable, establishes when an area specification can be drawn by a restricted class of symmetric, convex Venn-3 diagrams [9]. Extending to non-symmetric cases is essential to ensure practical applicability.

More generally, other automated area-proportional drawing methods include that by Chow, which draws so-called monotone diagrams (including Venn-3) [5]; usability problems arise because the curves are typically non-convex. It is unknown which area specifications are drawn with convex curves by Chow's method, although we note that attaining convexity was not a primary concern. Other work, by Chow and Ruskey [4], produced rectilinear diagrams, further studied in [2]. These are diagrams drawn with curves whose segments can only be horizontal or vertical, hence have a series of 90 degree bends. However, in general, rectilinear layouts of Venn-3 require non-convex curves. Diagrams drawn with convex curves are more likely to possess desirable aesthetic qualities, with reduced visual complexity, and in addition, the most desirable shapes (such as circles, ovals and rectangles) are convex.

This paper analyzes area-proportional Venn-3 diagrams drawn with convex polygons, with four main contributions: (a) a classification of some area specifications that can be drawn using convex polygons, (b) construction methods to draw a convex diagram, where the area specification has been identified as drawable in this manner, (c) a method to draw the area specification when these methods fail, thus ensuring that every area specification can be drawn, and (d) a freely available software tool for drawing these diagrams; see *www.cs.kent.ac.uk/people/staff/pjr/ConvexVenn3/diagrams2010.html*. Section 2 gives preliminary definitions, of Venn-3 diagrams and related concepts, as well as presenting some required linear algebra concepts. In Section 3, we define several classes of Venn-3 diagrams that allow us to investigate which area specifications can be drawn using convex curves. We also demonstrate (analytical) methods to draw such diagrams. Finally, Section 4 describes our software implementation, including details of drawing methods that rely on numerical approaches when our analytical drawing methods cannot be applied.

## 2 Venn Diagrams and Shears of the Plane

The labels in Venn diagrams are drawn from a set, $\mathcal{L}$. Given a closed curve, $c$, the set of points interior to $c$ will be denoted $int(c)$. Similarly, the set of those points exterior to $c$ is denoted $ext(c)$.

**Definition 1.** *A* ***Venn diagram****, $d = (Curve, l)$, is a pair where*

1. *Curve is a finite set of simple, closed curves in $\mathbb{R}^2$,*
2. *$l\colon Curve \to \mathcal{L}$ is an injective labelling function that assigns a label to each curve, and*
3. *for each subset, $C$, of Curve, the set*

$$\bigcap_{c \in C} int(c) \cap \bigcap_{c \in Curve - C} ext(c)$$

*is non-empty and forms a simply connected component in $\mathbb{R}^2$ called a* ***minimal region*** *and is* ***described*** *by $\{l(c) : c \in C\}$.*

*If $|Curve| = n$ then $d$ is a* ***Venn-n diagram****. If at most two curves intersect at any given point then $d$ is* ***simple****. If each curve in $d$ is convex then $d$ is* ***convex****.*

We focus on the construction of simple, convex Venn-3 using only polygons (recall, a polygon is a closed curve). Therefore, we assume, without loss of generality, that $\mathcal{L} = \{A, B, C\}$. Thus, the set $R = \mathbb{P}\{A, B, C\}$ is the set of all minimal region descriptions for Venn-3. Given a Venn-3 diagram, we will identify minimal regions in the diagram by their descriptions. For instance, the minimal region inside all 3 curves is properly described by $\{A, B, C\}$, but we will abuse notation and write this as $ABC$. A minimal region inside curves labelled $A$ and $B$ but outside $C$ will, therefore, be identified by $AB$. Furthermore, we will often blur the distinction between the minimal region and its description, simply writing $ABC$ to mean the region inside $A$, $B$ and $C$. We call the minimal region $ABC$ the **triple intersection**, $AB$, $AC$ and $BC$ are **double intersections**, and $A$, $B$, and $C$ are **single intersections**. A **region** in $d$ is a (not necessarily connected) component of $\mathbb{R}^2$ that is a union of minimal regions. The region formed by the union of $ABC$ and $AB$ is, therefore, described by $\{ABC, AB\}$. The region $\{AB, AC, BC, ABC\}$ is called the **core** of $d$ [2].

In order to provide a construction of an area-proportional Venn-3 diagram, we need to start with a specification of the required areas. Our definition of an area specification allows areas to be zero or negative; allowing for non-positive areas is for convenience later in the paper when we present methods for drawing Venn-3 diagrams.

**Definition 2.** *An* ***area specification*** *is a function, area*: $R \to \mathbb{R}$, *that assigns an area to each minimal region description. Given a Venn-3 diagram,* $d = (Curve, l)$, *if, for each* $r \in R$, *the area of the minimal region,* $mr$, *described by* $r$, *is* $area(r)$ *(that is,* $area(r) = area(mr)$*) then* $d$ ***represents*** $area\colon R \to \mathbb{R}$.

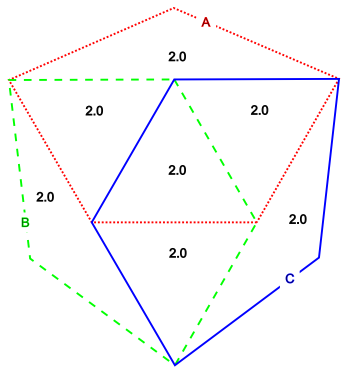



**Fig. 2.** An area-proportional Venn-3 diagram

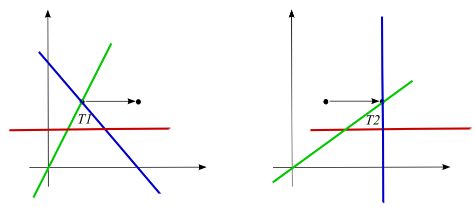



**Fig. 3.** A shear of the plane

For example, the diagram in Figure 2 represents the area specification given by the areas written inside the minimal regions and was produced using our software. For convenience and readability, we will further blur the distinction between regions, their descriptions, and their areas. For instance, we will take the region description $ABC$ to also mean either the minimal region it is describing or the area of that region; context will make the meaning clear.

Given a Venn-3 diagram, we might be able to apply a transformation to it that converts it to a form that is more easily analyzable, whilst maintaining the areas. Since we are constructing diagrams with polygons, we can apply a type of linear transformation, called a shear, to a diagram that alters its appearance but maintains both convexity and the areas of the regions. A shear of the plane can be seen in Figure 3, which keeps the $x$-axis fixed and moves the point as indicated by the arrow. The effect of the transformation can be seen in the righthand diagram. The two triangles $T1$ and $T2$ have the same area.

**Definition 3.** *A shear of the plane,* $\mathbb{R}^2$, *is a linear transformation defined by fixing a line,* $ax + by = c$, *and moving some other point,* $(p, q)$, *some distance,* $d$, *parallel to the line.*

For example, if we keep the $x$-axis fixed and the point $(0, 1)$ moves to $(1, 1)$ (here, $d = 1$) then each point, $(x, y)$, maps to $(x + y, y)$.

**Lemma 1.** *Let* $l$ *be a shear of the plane. Then* $l$ *preserves both lines and areas.*

As a consequence of Lemma 1, we know that, under a shear of the plane, any triangle maps to another triangle and that any convex polygon remains convex.

## 3 Drawing Convex Venn-3 Diagrams

To present our analysis of area specifications that are drawable by convex, Venn-3 diagrams, we introduce several classes of Venn-3 diagram. These classes are characterized by the shapes of (some of) the regions in the diagrams. The first two classes allow us to identify some area specifications as drawable by a convex Venn-3 diagram and, moreover, we demonstrate how to draw such a diagram. The remaining two classes allow more area specifications to be drawn.

For all the diagrams in this section we can choose to use an equilateral triangle for the triple intersection as any triangle can be transformed into an equilateral triangle by applying two shears.

### 3.1 Core-Triangular Diagrams

To define our first class of diagram, called the core-triangular class, we require the notion of an inscribed triangle: a triangle, $T_1$, is **inscribed** inside triangle $T_2$ if the corners of $T_1$ lie on the edges of $T_2$.

**Definition 4.** *A Venn-3 diagram is **core-triangular** if it is convex and*

1. *the core is triangular,*
2. *ABC is a triangle inscribed inside the core, and*
3. *the regions A, B and C are also triangles.*

For example, the diagrams in Figure 4 are core-triangular. We observe that, in any core-triangular diagram, the regions $AB$, $AC$, and $BC$ are also triangles. We can derive a relationship between the sum of the double and triple intersection areas and a product involving these areas that establishes whether an area

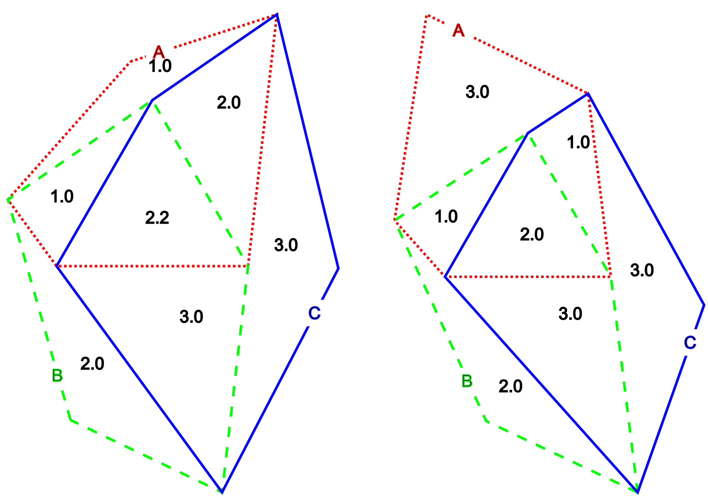



**Fig. 4.** Core-triangular diagrams

specification is drawable by a diagram in this class. The derivation relies on an analysis of the geometry of these diagrams, but for space reasons we omit the details.

**Theorem 1 (Representability Constraint: Core-Triangular).** *An area specification,* $area\colon R \to \mathbb{R}^+ - \{0\}$, *is representable by a core-triangular diagram if and only if*

$$AB + AC + BC + ABC \geq 4 \times \left( \frac{AB}{ABC} \times \frac{AC}{ABC} \times \frac{BC}{ABC} \right) \times ABC.$$

To illustrate, the area specifications as illustrated in Figure 4 satisfy the inequality in Theorem 1. However, any area specification with $AB = AC = BC = 2$ and $ABC = 1$ is not representable by a core-triangular diagram, since

$$2 + 2 + 2 + 1 = 7 \geq 4 \times 2 \times 2 \times 2 \times 1 = 32$$

is false.

Core-triangular diagrams form a sub-class of our next diagram type, triangular diagrams, in which all minimal regions are triangles but the core need not be triangular. We provide many more details for the derivation of a representability constraint for triangular diagrams, with that for core-triangular diagrams being a special case. Moreover, we provide a method for drawing triangular diagrams which can be used to draw core-triangular diagrams.

### 3.2 Triangular Diagrams

The diagrams in Figure 5 are triangular. As with core-triangular diagrams, we also identify exactly which area specifications can be drawn with triangular diagrams.

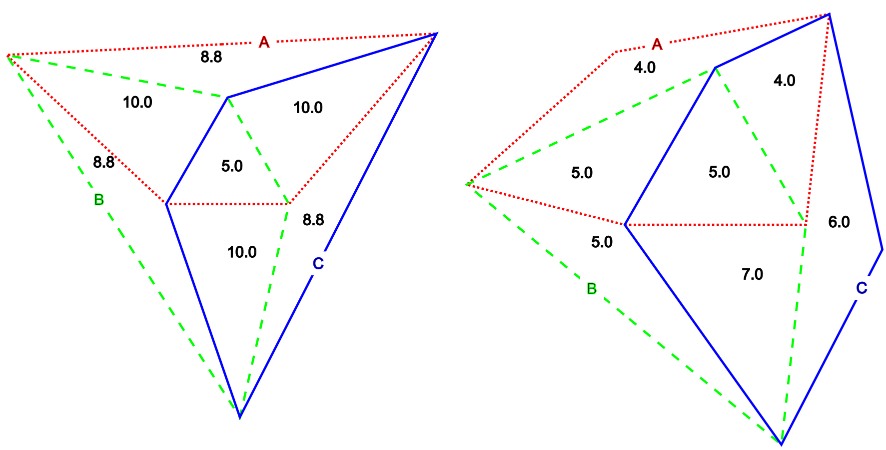



**Fig. 5.** Triangular diagrams

**Definition 5.** *A Venn-3 diagram is* ***triangular*** *if it is convex and all of the minimal regions are triangles.*

If an area specification can be represented by a diagram in the triangular class then it can be represented by such a diagram where the inner-most triangle, $ABC$, is rightangled (Theorem 2 below). We use this insight to establish exactly which area specifications are representable by triangular diagrams.

**Theorem 2.** *Let $d_1$ be a triangular diagram. Then there exists a triangular diagram, $d_2$, such that the region $ABC$ in $d_2$ is an rightangled triangle with the two edges next to the rightangled corner having the same length and $d_1$ and $d_2$ both represent the same area specification.*

*Proof.* We can apply two shears in order to obtain $d_2$. Assume, without loss of generality, that $A$ is located at $(0, 0)$ and that $B$ does not lie on either axis. Apply a shear such that $A$ is fixed and $B$ maps to $B' = (\sqrt{2 \times area}, 0)$ where $area$ is the area of $ABC$. Define $C'$ to be the point to which $C$ maps under this shear. The triangle $AB'C'$ has the same area as the triangle $ABC$, so $C'$ is at $(\lambda, \sqrt{2 \times area})$ for some $\lambda$. The second shear fixes the line $AB'$ and maps $C'$ to $C'' = (0, \sqrt{2 \times area})$. The final triangle $AB'C''$ has a rightangle at $A$ and the adjacent sides both of length $\sqrt{2 \times area}$. Since we have applied shears, we remain in the triangular class, by Lemma 1, as required.

We can, therefore, assume that the region $ABC$ is a rightangled triangle, with the 90-degree corner at $(0, 0)$, as indicated in Figure 6 which shows a partially drawn Venn-3 diagram. Given such a drawing of $ABC$, we can determine three lines, each parallel with one of the edges of $ABC$, the distance from which is

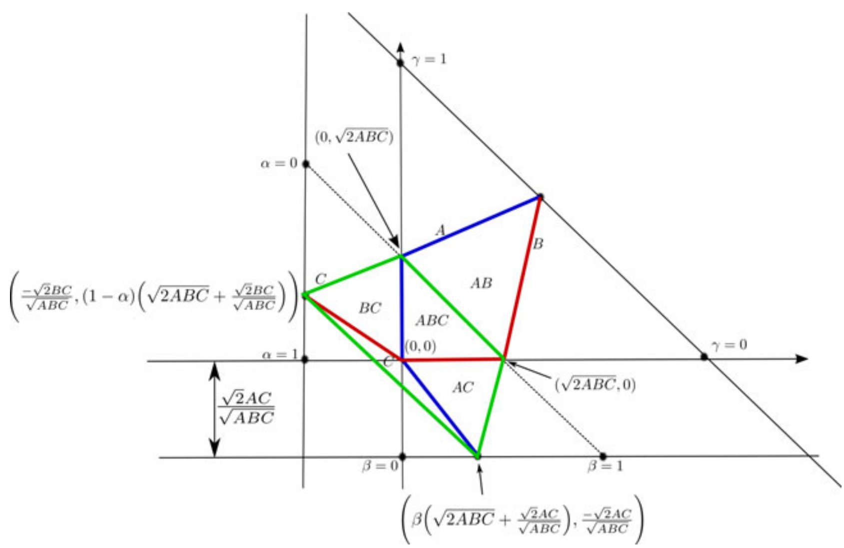



**Fig. 6.** Deriving area constraints

determined by the required area of the double intersections. The triangles $AB$, $AC$ and $BC$ each have a vertex on one of these lines. The location of the vertex is constrained, since we must ensure convexity. For example, $AB$ in Figure 6 must have a vertex lying between the points $\gamma = 0$ (if negative, the $B$ curve becomes non-convex) and $\gamma = 1$ (if bigger than 1, the $A$ curve becomes non-convex). When attempting to construct a triangular diagram for a given area specification, our task is to find a suitable $\alpha$, $\beta$ and $\gamma$.

Now, the area of a triangle can be computed via the determinant of a matrix: a triangle, $T$, with vertices at $(x_1, y_1)$, $(x_2, y_2)$, and $(x_3, y_3)$ has area

$$area(T) = \frac{\det}{2}\begin{pmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{pmatrix}.$$

In what follows, we assume that two sides of each of the triangles $A$, $B$ and $C$ are formed from edges of the double intersections, discussing the more general case later. The triangle $C$ has area:

$$\begin{aligned} C &= \frac{\det}{2}\begin{pmatrix} 1 & 1 & 1 \\ 0 & \frac{-\sqrt{2}BC}{\sqrt{ABC}} & \beta\left(\sqrt{2ABC} + \frac{\sqrt{2}AC}{\sqrt{ABC}}\right) \\ 0 & (1-\alpha)\left(\sqrt{2ABC} + \frac{\sqrt{2}BC}{\sqrt{ABC}}\right) & \frac{-\sqrt{2}AC}{\sqrt{ABC}} \end{pmatrix} \\ &= \frac{AC \times BC}{ABC} - \frac{(1-\alpha)\beta}{ABC}\Big(ABC + BC\Big)\Big(ABC + AC\Big) \end{aligned}$$

Rearranging the above, setting $X = (1-\alpha)\beta$, we have:

$$X = (1-\alpha)\beta = \frac{AC \times BC - C \times ABC}{(AC + ABC)(BC + ABC)} \tag{1}$$

Similarly, using the triangles $A$ and $B$, we can deduce

$$Y = (1-\beta)\gamma = \frac{AB \times AC - A \times ABC}{(AB + ABC)(AC + ABC)} \tag{2}$$

and

$$Z = (1-\gamma)\alpha = \frac{BC \times AB - B \times ABC}{(BC + ABC)(AB + ABC)}. \tag{3}$$

Thus, we have three equations, (1), (2) and (3), with three unknowns, $\alpha$, $\beta$ and $\gamma$, from which we can derive a quadratic in $\alpha$:

$$(1-Y)\alpha^2 + (X + Y - Z - 1)\alpha + Z(1-X) = 0.$$

This has real solutions provided the discriminant, $(1 - X - Y - Z)^2 - 4XYZ$, is not negative. Once we have found $\alpha$, we can then compute $\beta$ and $\gamma$. If solvable, with all of $\alpha$, $\beta$ and $\gamma$ between 0 and 1 (for convexity; see Figure 6), then the area specification is representable by a triangular, convex diagram. Such a solution is called **valid**.

To illustrate, consider the area specification for the lefthand diagram in Figure 5. We have

$$X = (1-\alpha)\beta = \frac{AC \times BC - C \times ABC}{(BC + ABC)(AC + ABC)} = \frac{10 \times 10 - 8.8 \times 5}{(10+5)(10+5)} = \frac{56}{225} \approx 0.25$$

Similarly, $Y = (1-\beta)\gamma = (1-\gamma)\alpha \approx 0.25$ which has solutions $\alpha = \beta = \gamma \approx 0.5$.

As previously stated, the algebra above assumes that two sides of each of the triangles $A$, $B$ and $C$ are formed from edges of the double intersection areas. For the general case, an area specification, *area*, is representable by a triangular diagram if there is a valid solution to (1), (2) and (3) with any of the single intersection areas reduced to zero. If we can determine drawability using the above method when some of these single intersection areas are reduced to 0 then we can enlarge those areas to produce a diagram with the required specification. For example, in Figure 5, the area specification for the righthand diagram would be deemed undrawable unless we reduce the $A$ and $C$ areas to zero before following the above process; without altering the area specification, there is no solution with all of $\alpha$, $\beta$ and $\gamma$ between 0 and 1. Taking the area specification, $area'$, equal to area but with $area'(A) = area'(C) = 0$, we find a valid solution with $\alpha = 0.65$, $\beta = 0.75$ and $\gamma = 0.87$. Using this solution, we can draw all of the diagram except $A$ and $C$. As a post process, we add triangles for $A$ and $C$ to give a diagram as shown. The arguments we have presented establish the following result:

**Theorem 3 (Representability Constraint: Triangular).** *An area specification, area*: $R \rightarrow \mathbb{R}^+ - \{0\}$, *is representable by a triangular diagram if and only if there exists another area specification,* $area'$: $R \rightarrow \mathbb{R}^+$ *which is the same as area, except that some of the single intersections may map to* 0, *and there exists valid solution to (1), (2), and (3) for* $area'$.

### 3.3 DT-Triangular Diagrams

A third class of diagram restricts the regions representing the double and triple set intersections to being triangular (hence the name DT-triangular):

**Definition 6.** *A Venn-3 diagram is* ***DT-triangular*** *if* $AB$, $AC$, $BC$ *and* $ABC$ *are all triangles and* $A$, $B$, *and* $C$ *are all polygons.*

The diagrams in Figure 7 are DT-triangular; the lefthand diagram has an area specification that cannot be drawn by a triangular diagram. Clearly, any triangular diagram is also DT-triangular and so, therefore, is any core-triangular

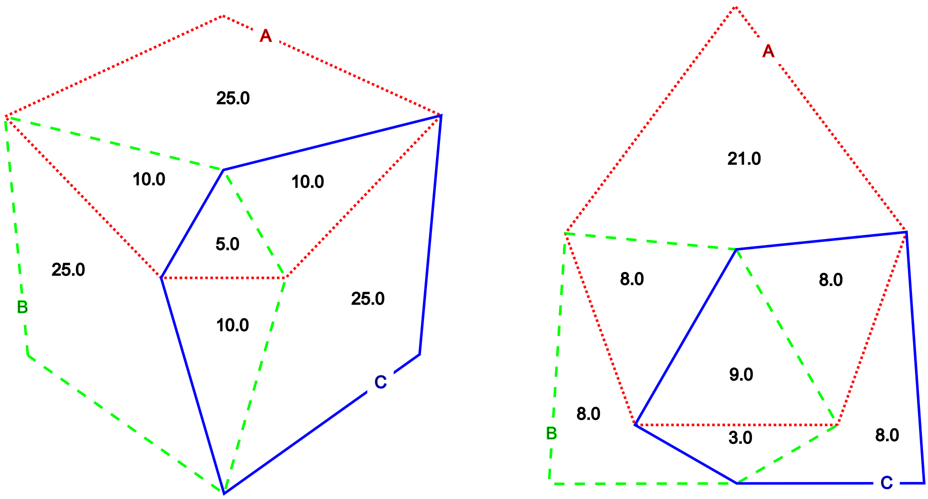



**Fig. 7.** DT-triangular diagrams

diagram. We will use our drawing method for triangular diagrams to enable the construction of DT-triangular diagrams. The method relies on a numerical search (which we have implemented) to find a 'reduced' area specification (reducing the single set areas) that can be drawn by a triangular diagram, $d$. We then enlarge the single set regions, to give a diagram with the required area specification. It can be shown that every area specification can be represented by a DT-triangular diagram, but not necessarily in a convex manner:

**Theorem 4.** *Any area specification, area*: $R \to \mathbb{R}^+ - \{0\}$, *can be represented by a DT-triangular diagram.*

To justify this result, draw an equilateral triangle for the triple intersection and appropriate triangles for the double intersections. Complete the polygons to produce the correct areas for the single intersections.

### 3.4 CH-Triangular Diagrams

Our final diagram class is illustrated in Figure 8. Here, the convex hull (CH) of the core is a triangle and the triple intersection is also a triangle. CH-triangular diagrams allow some area specifications to be drawn using only convex curves that cannot be drawn in this manner by diagrams from any of the other classes.

**Definition 7.** *Let $d$ be a Venn-3 diagram. Then $d$ is* ***CH-triangular*** *if*

1. *the convex hull of the core is a triangle, $T$,*
2. *$ABC$ is a triangle,*
3. *each of $AB$, $AC$, and $BC$ is a polygon with at most five sides,*

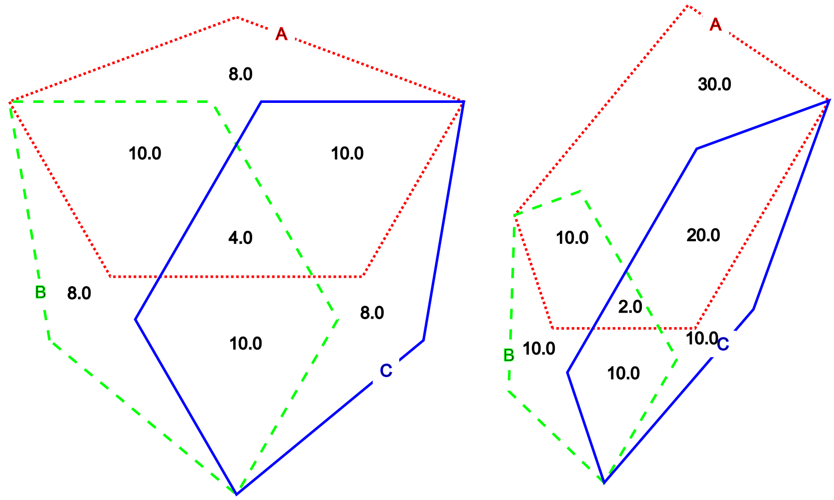



**Fig. 8.** CH-triangular diagrams

4. *the convex hull $T$ less the core consists of connected components that form triangles, of which there are at most three, each of which has two edges colinear with two edges of $ABC$, and*
5. *the remaining minimal regions, $A$, $B$, and $C$, are all polygons.*

In Figure 8, the area specification for the lefthand diagram cannot be represented by a convex DT-triangular diagram or, therefore, a triangular or a core-triangular diagram. We demonstrate a construction method for CH-triangular diagrams in the implementation section.

## 4 Implementation

In this section we discuss details of the implemented software, for a Java applet see *www.cs.kent.ac.uk/people/staff/pjr/ConvexVenn3/diagrams2010.html*. This software allows the user to enter an area specification and diagram type, and show the resultant diagram if a drawing is possible. The diagrams in all figures in this paper except 1, 3, 6 and 9 were drawn entirely with the software. In the case of DT-triangular diagrams, the software can produce diagrams with non-convex curves (when the double intersection areas are proportionally very small). As in the examples given previously, we use an equilateral triangle for the triple intersection of all diagram types because this tends to improve the usability of the final diagram, and also reduces the number of variables that need to be optimized during the search process described below.

In the case of core-triangular diagrams and triangular diagrams, the implementation uses the construction methods previously outlined. In the case of DT-triangular and CH-triangular diagrams, we have no analytical approach for drawing an appropriate diagram, instead we use search mechanisms, outlined in the following two sections.

### 4.1 Constructing DT-Triangular Diagrams

In order to draw a convex, DT-triangular diagram, we reduce the areas of the single intersections until we obtain an area specification that is drawable as a triangular diagram. That is, we seek $A'$, $B'$ and $C'$ where $A' \leq A$, $B' \leq B$, and $C' \leq C$ where the new area specification has a valid solution, as described in Section 3; note that $A'$, $B'$ and $C'$ could be negative. Moreover, we seek a solution where the discriminant for the quadratic arising from equations (1), (2) and (3) is zero. We note that, when the discriminant is zero, the to-be-drawn diagram will be more symmetric, as the outer points of $AB$, $AC$ and $BC$ will be closer to the centre of inner equilateral triangle. If this solution is valid, we can then proceed to draw a triangular diagram for the reduced specification. Once we have this drawing, we can enlarge the single intersection minimal regions, until they have the areas as required in the original area specification. If no valid solution can be found, it is possible that the original area specification can be drawn as a convex CH-triangular diagram, for which we discuss our drawing method in the next subsection. However, a non-valid solution can still yield a DT-triangular diagram, but it will not be convex.

To illustrate the drawing method for DT-triangular diagrams, we start with the area specification for the lefthand diagram in Figure 7. Here, the single set areas are all 25. Reducing them to 8.8 yields an area specification that is representable by a triangular diagram; see Figure 5. We can then enlarge the single intersections, pulling the polygons outwards, to produce the shown DT-triangular diagram (lefthand side of Figure 7). Drawing the righthand diagram of Figure 7 requires both $B'$ and $C'$ to be negative, however this is not a barrier to the drawing of the diagram, and the same process can be applied.

### 4.2 Constructing CH-Triangular Diagrams

If an area specification cannot be represented by a DT-triangular, convex diagram then it might be representable by a CH-triangular, convex diagram, as illustrated previously. In fact, we have not found any area specifications that can be represented by a DT-triangular, convex diagram that cannot also be represented by a CH-triangular, convex diagram. A further advantage of using CH-triangular diagrams is that they can have fewer points where two curves intersect and that is also a vertex of a polygon, which can increase the usability of the diagram. The process for drawing a CH-Triangular diagram is as follows:

1. The method first finds a drawing (convex or non-convex) for the specification with the DT-Triangular method, as given in the previous subsection. The convex hull of the core of this diagram forms a starting point for the search that finds the core of the CH-triangular diagram.
2. The DT-Triangular diagram is converted to a CH-Triangular diagram by extending the vertices of the curves beyond the middle triangle as shown in Figure 9: the solid lines show the DT-Triangular core, the dotted lines show how the curve line segments change when the diagram is converted to a CH-Triangular core.

3. An attempt is made to produce the correct double intersection areas using a search mechanism. The search ensures that if the diagram can be drawn using only convex curves with the given single intersection areas, then it will be. Each point on the convex hull of the core is tested for a number of possible moves close to the current location, to see whether there is a location that improves a heuristic. The heuristic is based on minimizing the variance of difference of the current double intersection areas against required areas. However, a move is only made as long as it results in a convex diagram. The process is repeated until the heuristic gives a zero result or no more improvement can be made. If the search finishes and the heuristic is zero, the diagram can be drawn in a convex manner. If it is not zero, then an additional search is made, this time relaxing the convex requirement until the heuristic becomes zero. In this case the diagram will be non-convex.
4. The outer vertices of the single intersections are placed so that the corresponding areas are enlarged until they match the area specification. This results in diagrams of the form shown in Figure 8. If the diagram can only be drawn with a CH-Triangular diagram type in a non-convex manner (in the case the extra search is required in the previous step), then a non-convex drawing is be generated by indenting the one set areas into the cut outs.

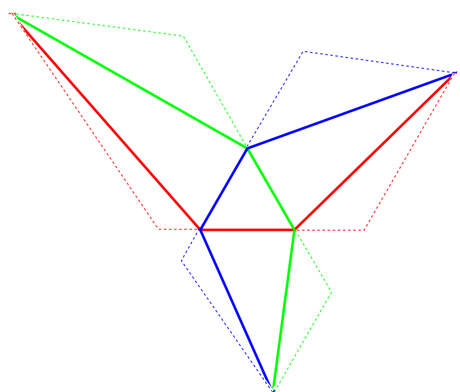


**Fig. 9.** Converting between DT and CH diagram types

### 4.3 Layout Improvements

One significant usability issue is that, in these diagram types, the vertices of the curves often coincide with intersection points. This can make following the correct curve through the intersection difficult, particularly when colour cannot be used to distinguish the curves. As a result, we demonstrate a layout improvement mechanism that could be applied to all diagram types, but is only implemented for CH-triangular diagrams. The layout improvement method moves the vertices away from the intersection points by elongating the relevant curve segment out from the centre of the diagram. The vertices of the affected polygons are then moved to compensate for this change. An example for the CH-triangular diagram shown in Figure 10, where the lefthand diagram is modified to give the (improved) righthand diagram. Here, only the outer intersection points are affected, as the diagram type naturally separates the $ABC$ triangle intersection points from the curve vertices.

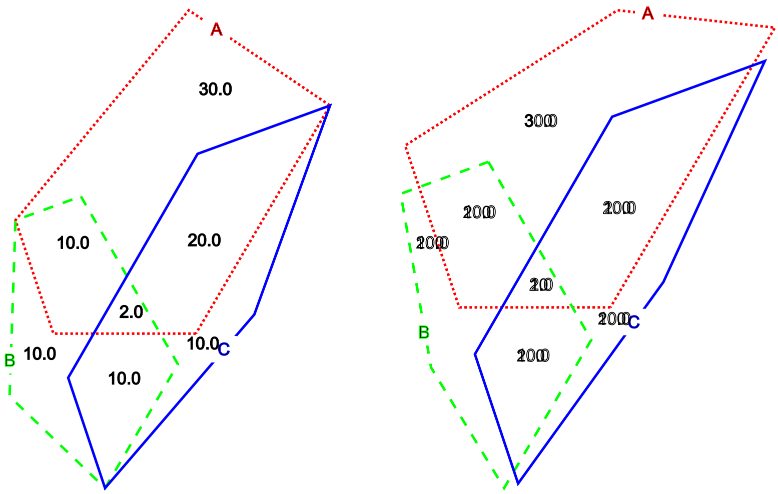



**Fig. 10.** Improving the layout of a CH-Triangular diagram

In these cases, for a given polygon, the line segments that border the two set areas point 'outwards' when the two set areas are of sufficient size, allowing the one set area enclosed by the curve to be as large as required. However, as the two set areas bordered by the curve reduce in size, they become closer to the three set border to the point where they become parallel. If the two set areas reduce further in size, these line segments start to point 'inwards'. At this stage the one set area is restricted in maximum size, if the diagram is still to remain convex. However, a more sophisticated implementation would avoid this by a more exact measurement of the elongation.

## 5 Conclusion

We have provided several classes of Venn-3 diagram that have allowed us to identify some area specifications as drawable with a convex diagram. Given an area specification, we have provided construction methods that draw a diagram representing it. In order to enhance the practical applicability of our results, we have provided a software implementation that draws an appropriate diagram, given an area specification.

Future work will involve a further analysis of which area specifications can be represented by a convex diagram. Ultimately, we would like to know exactly which area specifications can be represented in this way and how to construct such drawings of them. By considering convex polygons, we restricted the kinds of diagrams that could be drawn. The general case, where arbitrary convex curves can be used, is likely to be extremely challenging. In addition, natural extensions of the research are to consider Euler-3 diagrams, where not all of the minimal regions need to be present, and to examine diagrams with more than three curves.

**Acknowledgements.** We thank Shena Flower for helpful discussions on aspects of this paper. This research forms part of the Visualization with Euler Diagrams project, funded by EPSRC grants EP/E011160/1 and EP/E010393/1. The work has also been supported by EPSRC grants EP/H012311/1 and EP/H048480/1.

## References

1. Google Charts API, `http://code.google.com/apis/chart/` (accessed August 2009)
2. Chow, S.: Generating and Drawing Area-Proportional Euler and Venn Diagrams. PhD thesis, University of Victoria (2007), `hdl.handle.net/1828/128`
3. Chow, S., Rodgers, P.: Constructing area-proportional Venn and Euler diagrams with three circles. In: Proceedings of Euler Diagrams 2005 (2005)
4. Chow, S., Ruskey, F.: Drawing area-proportional Venn and Euler diagrams. In: Liotta, G. (ed.) GD 2003. LNCS, vol. 2912, pp. 466–477. Springer, Heidelberg (2004)
5. Chow, S., Ruskey, F.: Towards a general solution to drawing area-proportional Euler diagrams. In: Euler Diagrams 2004. ENTCS, vol. 134, pp. 3–18 (2005)
6. Farfel, M., et al.: An overview of 9/11 experiences and respiratory and mental health conditions among world trade center health registry enrollees. Journal of Urban Health 85(6), 880–909 (2008)
7. Ip, E.: Visualizing multiple regression. Journal of Statistics Education 9(1) (2001)
8. Kestler, H., Muller, A., Kraus, J., Buchholz, M., Gress, T., Liu, H., Kane, D., Zeeberg, B., Weinstein, J.: Vennmaster: Area-proportional Euler diagrams for functional GO analysis of microarrays. BMC Bioinformatics 9(67) (2008)
9. Rodgers, P., Flower, J., Stapleton, G., Howse, J.: Some results for drawing area proportional Venn3 with convex curves. In: Information Visualization, pp. 667–672. IEEE, Los Alamitos (2009)
10. Soriano, J., Davis, K., Coleman, B., Visick, G., Mannino, D., Pride, N.: The proportional Venn diagram of obstructive lung disease. Chest 124, 474–481 (2003)

# Fragments of Spider Diagrams of Order and Their Relative Expressiveness

Aidan Delaney[1], Gem Stapleton[1], John Taylor[1], and Simon Thompson[2]

[1] Visual Modelling Group, University of Brighton
{a.j.delaney,g.e.stapleton,john.taylor}@brighton.ac.uk
[2] Computing Laboratory, University of Kent
s.j.thompson@kent.ac.uk

**Abstract.** Investigating the expressiveness of a diagrammatic logic provides insight into how its syntactic elements interact at the semantic level. Moreover, it allows for comparisons with other notations. Various expressiveness results for diagrammatic logics are known, such as the theorem that Shin's Venn-II system is equivalent to monadic first order logic. The techniques employed by Shin for Venn-II were adapted to allow the expressiveness of Euler diagrams to be investigated. We consider the expressiveness of spider diagrams of order (SDoO), which extend spider diagrams by including syntax that provides ordering information between elements. Fragments of SDoO are created by systematically removing each aspect of the syntax. We establish the relative expressiveness of the various fragments. In particular, one result establishes that spiders are syntactic sugar in any fragment that contains order, negation and shading. We also show that shading is syntactic sugar in any fragment containing negation and spiders. The existence of syntactic redundancy within the spider diagram of order logic is unsurprising however, we find it interesting that spiders or shading are redundant in fragments of the logic. Further expressiveness results are presented throughout the paper. The techniques we employ may well extend to related notations, such as the Euler/Venn logic of Swoboda et al. and Kent's constraint diagrams.

## 1 Introduction

Recent years have seen the development of a number of diagrammatic logics, including constraint diagrams [1], existential graphs [2], Euler diagrams [3], Euler/Venn [4], spider diagrams [5], and Venn-II [6]. Each of these logics, except constraint diagrams, have sound and complete reasoning systems; for constraint diagrams, complete fragments exist, such as that in [7]. Recently, an extension of spider diagrams has been proposed that permits the specification of ordering information on the universal set [8]; this extension is called spider diagrams of order and is the primary focus of this paper.

By contrast to the relatively large body of work on reasoning with these logics, relatively little exploration has been conducted into their expressive power. To our knowledge, the first expressiveness result for formal diagrammatic logics

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 69–83, 2010.

was due to Shin, who proved that her Venn-II system is equivalent to Monadic First Order Logic (MFOL) [6]; recall, in MFOL all predicate symbols are one place. Her proof strategy used syntactic manipulations of sentences in MFOL, turning them into a normal form that could easily be translated into a Venn-II diagram. Shin's strategy was adapted to establish that the expressiveness of Euler diagrams with shading was also that of MFOL [9]. Thus, the general techniques used to investigate and evaluate expressiveness in one notation may be helpful in other domains.

It has also been shown that spider diagrams are equivalent to MFOL with equality [10]; MFOL[=] extends MFOL by including =, allowing one to assert the distinctness of elements. To establish the expressiveness of spider diagrams, a different approach to that of Shin's for Venn-II was utilized. The proof strategy involved a model theoretic analysis of the closure properties of the model sets for the formulas of the language. In the case of spider diagrams of order, so-called becasue they provide ordering constraints on elements, it has been shown that they are equivalent to MFOL of Order [11]; MFOL[<] extends MFOL by including $<$, which is interpreted as a strict total order. MFOL[<] is strictly more expressive than MFOL[=] which, in turn, is strictly more expressive than MFOL. For this expressiveness result, spider diagrams of order were not directly compared MFOL[<]. Instead, it was shown that spider diagrams of order could define precisely the star-free regular languages. It is known that these languages are also precisely those definable by MFOL[<] [12].

In this paper, we establish the relative expressiveness of fragments of spider diagrams of order. If two distinct fragments are equivalent in expressive power then this gives insight into what may be expressed by syntactically different but semantically equivalent fragments. Such insight allows one to consider the manner in which any particular semantic concept may be defined syntactically, possibly leading to more helpful or more appropriate diagrams. If two fragments have differing expressive power then this allows us to identify when certain syntactic elements are necessary for formulating particular semantic concepts. This can allow for more effective diagrams to be chosen when defining concepts. In section 2, we define the syntax and semantics of spider diagrams of order. In section 3, we identify natural fragments of spider diagrams of order and our novel expressiveness results concerning their relative expressiveness.

## 2 Spider Diagrams of Order

This section provides a brief overview of spider diagrams of order (SDoO), slightly modified from [8]. Diagram $d_1$ in figure 1 contains two labelled closed curves, called *contours*. The diagram $d_1$ contains four minimal regions, called *zones*: one zone is inside just $P$, another inside just $Q$, and another is outside both $P$ and $Q$. The zone inside both $P$ and $Q$ of $d_1$ is *shaded*. This diagram also contains two *spiders*, $s$ and $r$. The diagram $d_2 \lhd d_3$ is a compound spider diagram of order.

First, we formally define the syntax, before proceeding to the semantics. The contour labels in spider diagrams are selected from a set $\mathcal{C}$. A *zone* is defined to

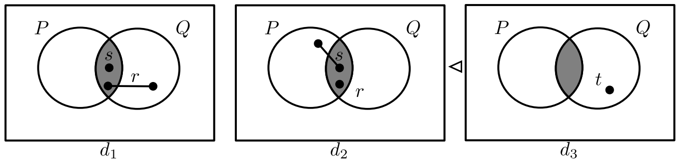



**Fig. 1.** A unitary diagram and a compound spider diagram

be a pair, $(in, out)$, of finite, disjoint subsets of $\mathcal{C}$. The set $in$ contains the labels of the contours that the zone is inside whereas $out$ contains the labels of the contours that the zone is outside. The set of all zones is denoted $\mathcal{Z}$. A *region* is a set of zones. To describe the spiders in a diagram, it is sufficient to say how many spiders are placed in each region. Thus, the abstract definition of a spider diagram will specify the labels used, the zones, the shaded zones and use a set of spider identifiers to describe the spiders.

**Definition 1 (Delaney et. al. [11]).** *A **unitary spider diagram of order**, $d$, is a quadruple $\langle C, Z, ShZ, SI\rangle$ where:*

1. *$C = C(d) \subseteq \mathcal{C}$ is a finite set of contour labels,*
2. *$Z = Z(d) \subseteq \{(in, C - in) : in \subseteq C\}$ is a set of zones,*
3. *$ShZ \subseteq Z(d)$ is a set of shaded zones, and*
4. *$SI = SI(d) \subsetneq \mathbb{N}^+ \times \mathbb{P}Z$ is a finite set of spider identifiers such that for all $(n, r), (m, s) \in SI(d)$ if $r = s$ then $n = m$.*

*The set of spiders in $d$ is defined to be $S(d) = \{(i, r) : (n, r) \in SI(d) \wedge 1 \leq i \leq n\}$. The symbol $\bot$ is also a unitary spider diagram of order. If $d_1$ and $d_2$ are spider diagrams of order then $(d_1 \vee d_2)$, $(d_1 \wedge d_2)$, $(d_1 \lhd d_2)$ and $\neg d_1$ are **spider diagrams of order**. Any diagram that is not a unitary diagram is a **compound** diagram.*

The abstract syntax of the diagram $d_1$ in figure 1 is

$$
\begin{aligned}
C(d_1) &= \{P, Q\},\\
Z(d_1) &= \{(\{\}, \{P, Q\}), (\{P\}, \{Q\}), (\{Q\}, \{P\}), (\{P, Q\}, \{\})\},\\
ShZ(d_1) &= \{(\{P, Q\}, \{\})\},\\
SI(d_1) &= \{(1, \{(\{P, Q\}, \{\})\}), (1, \{(\{Q\}, \{P\}), (\{P, Q\}), \{\}\})\}.
\end{aligned}
$$

By convention, we employ a lower-case $d$ to denote a spider diagram. An upper case $D$ will denote an arbitrary diagram. It is also useful to identify which zones could be present in a unitary diagram, given the label set, but are not present; semantically, *missing* zones provide information.

**Definition 2 (Howse et. al. [5]).** *Given a unitary diagram, $d$, a zone $(in, out)$ is **missing** from $d$ if it is in the set $\{(in, C(d) - in) : in \subseteq C(d)\} - Z(d)$ with the set of such zones denoted $MZ(d)$. If $MZ(d) = \emptyset$ then $d$ is in **Venn form**.*

Unitary diagrams make statements about sets (represented by contours) and their cardinalities (by using spiders and shading). The spiders in $d_1$, figure 1, represent distinct elements in the sets represented by the regions in which they are placed; spiders provide lower bounds on set cardinality. The spider $r$ provides disjunctive information: the element it represents is in one of the sets represented by the zones in which it is placed. Shading places an upper bound on set cardinality: in a shaded region, all elements must be represented by spiders. Taken together, the spiders $s$ and $r$ allow for the set represented by the shaded zone to contain between 1 and 2 elements. The semantics of spider diagrams are model-based: a model is an assignment of sets to contour labels that agrees with the intended meaning of the diagram.

**Definition 3 (Delaney et. al. [11]).** *An **interpretation** is a triple $I = (U, <, \Psi)$ where $U$ is called the universal set and $\Psi\colon \mathcal{C} \to \mathbb{P}U$ is a function that assigns a subset of $U$ to each contour label and $<$ is a strict total order on $U$. The function $\Psi$ can be extended to interpret zones and regions as follows:*

1. *each zone, $(a, b) \in \mathcal{Z}$, represents the set $\bigcap_{l\in a} \Psi(l) \cap \bigcap_{l\in b}(U - \Psi(l))$ and*
2. *each region, $r \in \mathbb{P}\mathcal{Z}$, represents the set which is the union of the sets represented by $r$'s constituent zones.*

*If $U = \emptyset$ then $I$ is the **empty** interpretation.*

**Definition 4 (Delaney et. al. [11]).** *Let $I = (U, <, \Psi)$ be an interpretation and let $d$ ($\neq\perp$) be a unitary spider diagram. Then $I$ is a **model** for $d$, denoted $m \models d$, if and only if the following conditions hold.*

1. ***The missing zones condition.*** *All of the missing zones represent the empty set, that is, $\bigcup_{z\in MZ(d)} \Psi(z) = \emptyset$.*
2. ***The function extension condition.*** *There exists an extension of $\Psi$ to spiders, $\Psi : \mathcal{C} \cup S(d) \to \mathbb{P}U$ for which the following hold.*
   (a) ***The spiders' locations condition.*** *All spiders represent elements (strictly, singleton sets) in the sets represented by the regions in which they are placed: $\forall (i, r) \in S(d)\, (\Psi(i, r) \subseteq \Psi(r) \wedge |\Psi(i, r)| = 1)$.*
   (b) ***The distinct spiders condition.*** *Distinct spiders denote distinct elements: $\forall s_1, s_2 \in S(d)(\Psi(s_1) = \Psi(s_2) \Rightarrow s_1 = s_2)$.*
   (c) ***The shading condition.*** *Shaded regions represent a subset of elements denoted by spiders: $\Psi(ShZ(d)) \subseteq \bigcup_{s\in S(d)} \Psi(s)$.*

*If $d = \perp$ then no interpretation is a model for $d$.*

The interpretation $m = (U, <, \Psi)$ where $U = \{1, 2, 3, 4\}$, $<$ is the natural order over $U$, $\Psi(P) = \{2\}$ and $\Psi(Q) = \{2, 3\}$ is a model for the diagram $d_1$ in figure 1, but not for $d_2$ or $d_3$. For compound diagrams, the definition of a model extends inductively. In the case of $\neg D_1$, $D_1 \vee D_2$ and $D_1 \wedge D_2$ the extension is obvious. A diagram of the form $D_1 \lhd D_2$ provides a constraint on the interpretation of

$<$. For instance, $d_2 \lhd d_3$ in figure 1 asserts, in part, that no elements in $P \cap Q$ can be ordered after the elements represented by $s$ and $r$ in $d_2$. We need to ensure that the ordering information provided by an interpretation respects the intended meaning of the diagram.

**Definition 5 (adapted from Ebbinghaus & Flum [13]).** *Let* $I_1 = (U_1, <_1, \Psi_1)$ *and* $I_2 = (U_2, <_2, \Psi_2)$ *be interpretations where* $U_1$ *and* $U_2$ *are disjoint. The* ***ordered sum*** *of* $I_1$ *and* $I_2$*, denoted* $I_1 + I_2$*, is defined to be the interpretation* $I_3 = (U_3, <_3, \Psi_3)$ *such that*

1. $U_3 = U_1 \cup U_2$
2. $<_3 = <_1 \cup <_2 \cup \{(u_1, u_2) : u_1 \in U_1 \wedge u_2 \in U_2\}$*, and*
3. *for each* $c \in \mathcal{C}$*,* $\Psi_3(c) = \Psi_1(c) \cup \Psi_2(c)$*.*

**Definition 6.** *Let* $I = (U, <, \Psi)$ *be an interpretation and let* $D$ *be a compound diagram. Then* $I$ *is a* ***model*** *for* $D$ *provided:*

1. *if* $D = D_1 \vee D_2$ *then* $I$ *models* $D$ *whenever* $I$ *models* $D_1$ *or* $I$ *models* $D_2$*,*
2. *if* $D = D_1 \wedge D_2$ *then* $I$ *models* $D$ *whenever* $I$ *models* $D_1$ *and* $I$ *models* $D_2$*,*
3. *if* $D = \neg D_1$ *then* $I$ *models* $D$ *whenever* $I$ *does not model* $D_1$*, and*
4. *if* $D = D_1 \lhd D_2$ *then* $I$ *models* $D$ *whenever there exist interpretations* $I_1$ *and* $I_2$ *such that* $I = I_1 + I_2$ *and* $I_1$ *models* $D_1$ *and* $I_2$ *models* $D_2$*.*

For the purpose of establishing relative expressiveness, we need the notion of *satisfiability* and to know when two diagrams are *semantically equivalent.*

**Definition 7 (Delaney et. al. [11]).** *Spider diagrams of order,* $D_1$ *and* $D_2$*, are* ***semantically equivalent*** *provided they have exactly the same models. If* $D_1$ *has a model then we say that* $D_1$ *is* ***satisfiable****.*

## 3 Expressiveness

We will now establish the relative expressiveness of various fragments of spider diagrams of order. In subsection 3.1 we define our notation for discussing fragments of SDoO and in subsection 3.2 we summarize previously known expressiveness results. Then in subsections 3.3 and 3.4 we provide definitions and results that are helpful for our analysis. The remainder of this section provides new expressiveness results.

### 3.1 Fragments of Spider Diagrams of Order

We observe that spider diagrams of order can be thought of as being built from Euler diagrams, with various syntactic additions. We view (unitary) Euler diagrams as the basic building blocks and this motivates our method of defining natural fragments of SDoO. To these basic building blocks we can add connectives ($\wedge$, $\vee$, $\lhd$), the negation operation ($\neg$), spiders, and shading. Using any set of these additions to Euler diagrams gives rise to a fragment of SDoO.

We denote the unitary Euler diagrams fragment by $ED$. Using notation similar to that seen in description logics, $ED[C]$ is taken to be the class of Euler diagrams formed by joining them with the conjunction, $\wedge$, operator. Equivalently, this is the fragment of SDoO in which there are no spiders, no shading, no negation, and the only logical connective is $\wedge$. If we wanted to include spiders, $Sp$ and conjunction, $C$, but no other operators and no shading then this fragment would be denoted by $ED[C, Sp]$. Given some list, $IncSyn$, that is a sublist of $[C, D, N, O, Sp, Sh]$, the fragment $ED[IncSyn]$ is then defined in the obvious manner, where $C$=conjunction, $D$=disjunction, $N$=negation, $O$=order ($\lhd$), $Sp$=spiders, and $Sh$=shading. Thus, 'full' SDoO is $ED[C, D, N, O, Sp, Sh]$. Importantly, we define it to be the case that fragments with no shading also do not include unitary diagrams with missing zones, since such zones can be replaced by shaded zones. We will frequently omit $ED$ from the fragment description and write, for example, $[CSp]$ rather than $ED[C, Sp]$.

### 3.2 Known Expressiveness Results

Known results for relative expressive power are summarized in table 1; all of these results were presented in the introduction, follow immediately from them, or appear elsewhere in the literature (primarily in [5]). The column headings give a fragment of SDoO, with the second column considering SDoO: $[CDNOSpSh]$. The third through eight columns define the fragment of $[CDNOShSp]$ without the syntax indicated by the heading i.e. the $-C$ column is the fragment $[DNOSpSh]$. Similarly, each row removes a (second) piece of syntax from the fragment, giving another fragment. Thus, column 3 in row 5 identifies that $[DNOSpSh]$ has greater expressiveness than $[DNSpSh]$. In this paper, we complete most of the missing entries in table 1.

**Table 1.** Summary of known relative expressiveness results

| | $[CDNOSpSh]$ | $-C$ | $-D$ | $-N$ | $-O$ | $-Sp$ | $-Sh$ | |
|---|---|---|---|---|---|---|---|---|
| $-C$ | $=$ | $-$ | | | $=$ | $=$ | $=$ | $-C$ |
| $-D$ | $=$ | | $-$ | | $=$ | $=$ | $=$ | $-D$ |
| $-N$ | | | | $-$ | $=$ | | | $-N$ |
| $-O$ | $<$ | $<$ | $<$ | $<$ | $-$ | $<$ | $<$ | $-O$ |
| $-Sp$ | | | | | $<$ | $-$ | | $-Sp$ |
| $-Sh$ | | | | | | | $-$ | $-Sh$ |

### 3.3 The $\alpha$-Diagram Fragments

Spiders whose habitats comprise more than one zone make disjunctive statements within a unitary diagram. However, it has been observed that this disjunctive information can also be made using a compound diagram. For example, $d_4$ in figure 2 is semantically equivalent to $d_5 \vee d_6$. One approach to investigating expressiveness is to consider only diagrams whose spiders are placed in single zones. Such diagrams are called **$\alpha$-diagrams** [5].

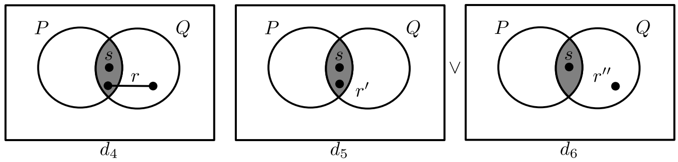



**Fig. 2.** Creating an $\alpha$-diagram

**Theorem 1 (Howse et al. [5]).** *Every unitary diagram is semantically equivalent to a disjunction of unitary $\alpha$-diagrams.*

**Theorem 2.** *Let $D_1$ be drawn from a fragment, $F$, of SDoO that contains*

1. *disjunction ($D$), or*
2. *conjunction ($C$) and negation ($N$).*

*Then there exists an $\alpha$-diagram, $D_2$, also in $F$, such that $D_1$ is semantically equivalent to $D_2$.*

*Proof (Sketch).* The proof follows by induction on the depth of $D_1$ in the inductive construction of diagrams, with the base case provided by theorem 1.

### 3.4 Literals

As well as reducing expressiveness questions to those for $\alpha$-diagrams, it is also helpful to consider unitary diagrams that contain information in, at most, one zone. For example, the unitary $\alpha$-diagram, $d_7$, in figure 3 contains exactly two zones which provide semantic information, and is semantically equivalent to $d_8 \wedge d_9$. The diagrams $d_8$ and $d_9$ are called *literals*, since they give information about exactly one zone; we say that they are *literal parts* of $d_7$. All diagrams in this example are in Venn form; missing zones would provide semantic information and we are seeking diagrams that provide information about a single zone. Our definition of a literal extends that of an Euler diagram literal [9].

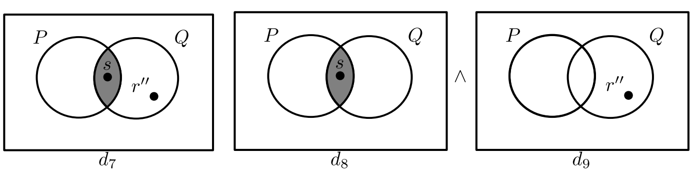



**Fig. 3.** Creating a diagram in literal form

**Definition 8.** *Let $d$ be a unitary $\alpha$-diagram in Venn form that contains at most one zone which contains spiders or shading. Then $d$ and $\neg d$ are called **literals**. The diagram $d$ is a **positive** literal, whereas $\neg d$ is a **negative** literal.*

**Definition 9.** *Let $D$ be an SDoO. If each unitary part of $D$ is a literal or $\perp$ then $D$ is in **literal form**.*

**Definition 10.** *Let $d_1$ ($\neq\perp$) be a unitary $\alpha$-diagram. A **literal part** of $d_1$ is a positive literal, $d_2$, that is formed from $d_1$ by adding all missing zones to the zone set and shaded zone set and, subsequently, deleting the spiders and shading from all except at most one zone.*

**Theorem 3.** *Let $d$ ($\neq\perp$) be a unitary $\alpha$-diagram. Then $d$ is semantically equivalent to the conjunction of its literal parts.*

*Proof (Sketch).* Since $d$ is a unitary $\alpha$-diagram it contains no disjunctive information, and so the semantics of the whole diagram is equivalent to the conjunction of the constraints in each zone.

**Theorem 4.** *Let $D_1$ be drawn from a fragment, $F$, of SDoO that contains either*

1. *disjunction ($D$) and conjunction ($C$), or*
2. *disjunction and negation ($N$), or*
3. *conjunction and negation ($N$).*

*Then there exists $D_2$, also in $F$, in literal form such that $D_1$ is semantically equivalent to $D_2$.*

*Proof (Sketch).* Noting that conversion of $D_1$ to an $\alpha$-diagram requires either disjunction or both conjunction and negation, we can use theorem 2 to reduce $D_1$ to an $\alpha$-diagram. The proof then follows by induction on the depth of $D_1$ in the inductive construction of diagrams, with the base case provided by theorem 3.

### 3.5 Removing Spiders

Some of our fragments do not contain spiders so we need to know whether their absence impacts expressiveness. Intuitively, one might expect their removal to decrease expressiveness, but this is not always so. Figure 4 demonstrates that it is possible to remove spiders from a positive literal without altering expressiveness provided we have access to negation, order, and shading: $d_8$ (figure 3) is semantically equivalent to $\neg d_{10} \wedge \neg(\neg d_{11} \lhd \neg d_{12})$, in figure 4.

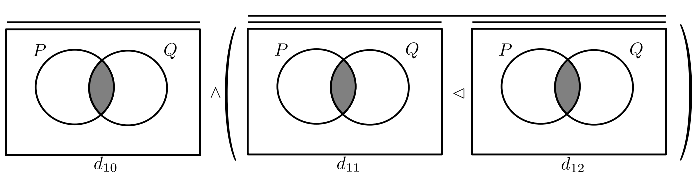



**Fig. 4.** Removing spiders from literal $d_8$ in figure 3

**Definition 11.** *Let* $d_1$ *be a positive literal with a zone* $z$ *containing spiders. Then the* ***spider-free*** *diagram associated with* $d_1$ *is a copy of* $d_1$ *except that* $z$ *contains no spiders and* $z$ *is shaded.*

We adopt the notation $d^n$ to mean $d \lhd d \lhd ... \lhd d$ ($n$ times).

**Theorem 5.** *Let* $d_1$ *be a positive literal, with the zone* $z$ *containing exactly* $n$ *spiders. Let* $d_2$ *be the spider-free diagram associated with* $d_1$*. If* $z$ *is not shaded then* $d_1$ *is semantically equivalent to* $(\neg d_2)^n$*. If* $z$ *is shaded then* $d_1$ *is semantically equivalent to* $(\neg d_2)^n \wedge \neg(\neg d_2)^{n+1}$*.*

*Proof (Sketch).* The models of $\neg d_2$ are those interpretations containing at least one element in $\Psi(z)$. The models of $\neg d_2 \lhd \neg d_2$ are, therefore, those interpretations which contain at least two elements in $\Psi(z)$. The result follows.

**Theorem 6.** *Let* $D_1$ *be be a diagram drawn from any fragment,* $F$*, of spider diagrams of order, provided that* $F$ *contains negation (*$N$*), order (*$O$*), and shading (*$Sh$*), and at least one of conjunction (*$C$*) and disjunction (*$D$*). Then* $D_1$ *is semantically equivalent to some diagram,* $D_2$*, also in* $F$*, such that* $D_2$ *contains no spiders.*

*Proof.* Since we have negation, having one or both of conjunction and disjunction does not alter expressiveness. Thus, without loss of generality, our proof assumes we have access to both $C$ and $D$. Theorem 2 allows us to replace $D_1$ by an $\alpha$-diagram, whilst remaining within $F$. Theorem 4 allows us to reduce the $\alpha$-diagram to literal form, again whilst remaining within $F$ (this replacement uses $C$). The result then essentially follows by theorem 5 (which uses $N$ and $C$).

Theorem 6 allows us to complete some of row concerning removal of spiders in table 1; see table 2 (new results shown in bold typeface).

**Table 2.** Expressiveness results when removing spiders

| | $[CDNOSpSh]$ | $-C$ | $-D$ | $-N$ | $-O$ | $-Sp$ | $-Sh$ | |
|---|---|---|---|---|---|---|---|---|
| $-Sp$ | $=$ | $=$ | $=$ | | $<$ | – | | $-Sp$ |

**Theorem 7.** *Euler diagrams of order are equivalent in expressiveness to SDoO.*

*Proof (Sketch).* This theorem is a restatement of theorem 6 with respect to the specific fragment $ED[C, D, N, O, Sh]$.

There are two entries to be completed in table 2. Concerning the first, spiders are removed from $ED[C, D, O, Sp, Sh]$ to give $ED[C, D, O, Sh]$. The following theorem allows us to deduce that this reduces expressiveness.

**Theorem 8.** *Let* $D \in ED[C, D, O, Sh]$*. If* $D$ *is satisfiable then* $I = (\emptyset, <, \Psi)$ *models* $D$*.*

*Proof (Sketch).* The proof proceeds by induction on the depth of $D$ in the inductive construction. Assume $D$ has a model. Trivially, if $D$ is a unitary diagram then $D$ contains no spiders and $I$ models $D$. If $D = D_1 \wedge D_2$ or $D = D_1 \vee D_2$ then the result follows trivially. Consider $D = D_1 \lhd D_2$. $D$ is satisfiable if and only if both $D_1$ and $D_2$ are satisfiable. Should this be the case, $I$ models $D_1$ and $I$ models $D_2$, by assumption. Now, $I = I + I$, so $I$ models $D$. Hence $D$ is modelled by the empty interpretation.

**Corollary 1.** *$ED[C, D, O, Sp, Sh]$ is more expressive than $ED[C, D, O, Sh]$.*

**Theorem 9.** *Any diagram drawn from $ED[C, D, N, O]$ is modelled by every interpretation or is not modelled by any interpretation.*

*Proof (Sketch).* The property of being equivalent to true or equivalent to false holds for the unitary diagrams in this language, and the property is preserved when formulas are conjoined, disjoined, negated or connected with product, ($O$).

Thus, the $ED[C, D, N, O]$ fragment is not terribly interesting: it can only make statements that are either valid or contradictory. We immediately have the following corollary and are now able complete the 'remove spiders' row.

**Corollary 2.** *$ED[C, D, N, O, Sp]$ is more expressive than $ED[C, D, N, O]$.*

### 3.6 Removing Shading

We now proceed to show that, under some circumstances, shading is syntactic sugar. For example, the diagram $d_{13} \wedge d_{14}$ presented in figure 3 is semantically equivalent to $d_8$ in figure 5. Intuitively, $d_8$ tells us that the shaded zone represents a set containing exactly 1 element, which is equivalent to saying there are at least 1 element ($d_{13}$) and not at least 2 elements ($\neg d_{14}$).

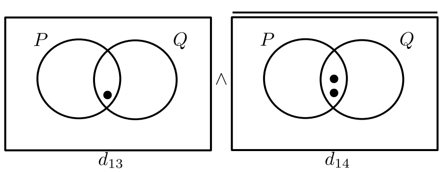



**Fig. 5.** Removing shading from literal $d_8$ in figure 3

**Theorem 10.** *Let $d$ be a positive literal with a shaded zone, $z$. Let $d_1$ be a copy of $d$, except that $z$ contains no shading. Let $d_2$ be a copy of $d$ except that $z$ contains no shading and exactly one more spider than in $d$. Then $d$ is semantically equivalent to $d_1 \wedge \neg d_2$.*

**Table 3.** Expressiveness results when removing shading

| | $[CDNOSpSh]$ | $-C$ | $-D$ | $-N$ | $-O$ | $-Sp$ | $-Sh$ | |
|---|---|---|---|---|---|---|---|---|
| $-Sh$ | $=$ | $=$ | $=$ | | $=$ | | – | $-Sh$ |

*Proof (Sketch).* Let $I = (U, <, \Psi)$ be an interpretation that models $d$. It follows that $|\Psi(z)| = n$, where $n$ is the number of spiders in $z$, since $z$ is shaded and $d$ is an $\alpha$-diagram. Clearly, $I$ is a model for $d_1$, since $z$ contains $n$ spiders in $d_1$. In $d_2$, $z$ contains $n+1$ spiders, so any model for $d_2$ has $|\Psi(z)| \geq n+1$. Thus, $I$ does not model $d_2$, so $I$ models $\neg d_2$. Hence, $I$ models $d_1 \wedge \neg d_2$. Conversely, suppose that $I$ models $d_1 \wedge \neg d_2$. Then $|\Psi(z)| \geq n$ (from $d_1$) and $|\Psi(z)| < n+1$ (from $d_2$). Since $d$ is a literal, $I$ models $d$. Hence $d$ is semantically equivalent to $d_1 \wedge \neg d_2$.

**Theorem 11.** *Let $D_1$ be a diagram drawn from any fragment, $F$, of spider diagrams of order, provided that $F$ contains negation ($N$), spiders, and either conjunction ($C$) or disjunction ($D$) (or both). Then $D_1$ is semantically equivalent to some diagram, $D_2$, also in $F$, such that $D_2$ contains no shading.*

*Proof (Sketch).* The proof is similar to that of theorem 6.

This theorem allows us to complete some of row concerning removal of shading in table 1; see table 3 (all entries are new results). There are two entries left to be completed in the removal of shading row in table 3. Concerning the first, shading is removed from $ED[C, D, O, Sp, Sh]$ to give $ED[C, D, O, Sp]$. We observe that an entirely shaded unitary diagram is satisfiable, but does not have models of arbitrarily large cardinality. Without shading and negation, we cannot provide upper bounds on set cardinality, captured by the following theorem.

**Theorem 12.** *Any diagram drawn from $ED[C, D, O, Sp]$ that is satisfiable has models of arbitrarily large cardinality.*

*Proof (Sketch).* This property holds of unitary diagrams which contain only spiders, and it is preserved when formulas are combined using conjunction, disjunction and product.

**Corollary 3.** *$ED[C, D, O, Sp, Sh]$ is more expressive than $ED[C, D, O, Sp]$.*

For the final entry in this row, we have a further corollary to theorem 9:

**Corollary 4.** *$ED[C, D, N, O, Sh]$ is more expressive than $ED[C, D, N, O]$.*

### 3.7 Removing Logical Operators

We now give a further four results concerning relative expressiveness, where we consider the removal of a logical operator from a fragment. The proofs of these results all use model theoretic arguments. First, we observe that any diagram, $D$, drawn from $ED[C, O, Sp, Sh]$ that is satisfied by the empty interpretation does not contain spiders. Thus, $D$ can make assertions such as a particular zone represents the empty set, or that elements in the set represented by one zone cannot be ordered before elements in another such set. Therefore, given a non-empty model for $D$, we can remove elements from the universal set (updating the interpretations of $<$ and the contour labels appropriately) and obtain another model for $D$. To make this insight precise, we first define a *sub-interpretation* of an interpretation.

**Definition 12 (adapted from Manzano [14]).** *A* ***sub-interpretation*** *of an interpretation* $I = (U, <, \Psi)$ *is an interpretation,* $I_r = (U_r, <_r, \Psi_r)$ *where*

1. $U_r \subseteq U$
2. $<_r = < \cap (U_r \times U_r)$, *and*
3. $\Psi_r(c) = \Psi(c) \cap U_r$, *for each* $c \in \mathcal{C}$.

**Lemma 1.** *Let* $I = (U, <, \Psi)$ *be an interpretation. with sub-interpretations* $I_s = (U_s, <_s, \Psi_s)$. *If* $I = I_1 + I_2$ *for some interpretations* $I_1 = (U_1, <_1, \Psi_1)$ *and* $I_2 = (U_2, <_2, \Psi_2)$ *then*

$$I_s = I_{1,s} + I_{2,s}$$

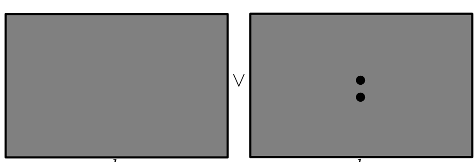
*where* $I_{1,s}$ *and* $I_{2,s}$ *are sub-interpretations of* $I_1$ *and* $I_2$ *with universal sets* $U_1 \cap U_s$ *and* $U_2 \cap U_s$ *respectively.*

**Theorem 13.** *Let* $D$ *be a diagram in* $ED[C, O, Sp, Sh]$ *such that* $(\emptyset, <, \Psi)$ *models* $D$. *Let* $I = (U, <, \Psi)$ *be a model for* $D$. *Then any sub-interpretation of* $I$ *also models* $D$.

*Proof.* Again, the result proceeds by induction where the interesting case is $D = D_1 \lhd D_2$. Given that the empty interpretation models $D$, it also models both $D_1$ and $D_2$. $I = I_1 + I_2$ where $I_1$ and $I_2$ are models for $D_1$ and $D_2$ respectively. Given a sub-interpretation, $I_s$, of $I$, by lemma 1 $I_s = I_{1,s} + I_{2,s}$ and by assumption $I_{1,s}$ and $I_{2,s}$ model $D_1$ and $D_2$ respectively. Hence $I_s$ models $D_1$.

**Corollary 5.** $ED[C, D, O, Sp, Sh]$ *is more expressive than* $ED[C, O, Sp, Sh]$.

To justify corollary 5, construct $D = d_{15} \vee d_{16}$ (figure 6) in $ED[C, D, O, Sp, Sh]$ where $d_{15}$ is unitary, containing no spiders and fully shaded, and $d_{16}$ is unitary, containing exactly two spiders and fully shaded. $D$ is satisfied by the empty interpretation (this satisfies $d_{15}$) and is satisfied by any model with exactly two



$d_{15}$ $d_{16}$

**Fig. 6.** An example to justify corollary 5

elements in the universal set (this satisfies $d_{16}$). Take any model for $D$ with two elements and create a sub-interpretation with one element. This is not a model for $D$, so $ED[C, D, O, Sp, Sh]$ can axiomatise more classes of interpretations than $ED[C, O, Sp, Sh]$. For corollary 6, observe that $D = \neg(\neg d_{15} \wedge \neg d_{16})$ ($d_1$ and $d_2$ as above) has same models as $d_{15} \vee d_{16}$, so the proof is similar.

**Corollary 6.** *$ED[C, N, O, Sp, Sh]$ is more expressive than $ED[C, O, Sp, Sh]$.*

**Corollary 7.** *$ED[C, D, N, O, Sh]$ is more expressive than $ED[C, D, O, Sh]$.*

Finally, consider diagrams in $ED[C, D, O, Sp]$.

**Theorem 14.** *Let $D \in ED[C, D, O, Sp]$. If $D$ is modelled by $(\emptyset, <, \Psi)$ then every interpretation models $D$.*

*Proof (Sketch).* Any unitary diagram in this fragment satisfied by the empty interpretation does not contain any spiders. Since there is no shading (and, therefore, no missing zones), such a diagram is satisfied by every interpretation. It can be shown, by induction, that any compound diagram, $D$, satisfied by the empty interpretation is also satisfied by every interpretation:

**Corollary 8.** *$ED[C, D, N, O, Sp]$ is more expressive than $ED[C, D, O, Sp]$*

To justify corollary 8, observe that the unitary diagram that contains no contours and exactly one spider is modelled by every interpretation except the empty interpretation. Therefore its negation is modelled by the empty interpretation, but has no other models.

### 3.8 Summary

Table 4a summarises the relative expressive power of fragments of the spider diagram of order logic, including the results in this paper (presented in bold typeface) and previously known results. We can use table 4a to deduce the expressive power of some fragments, due to the previously known expressiveness results. These results and deductions are presented in table 4b. Each entry identifies the expressiveness of the fragment obtained by removing syntax as indicated by the row and column heading. For example, the last row of the third column gives the expressiveness of $ED[D, N, O, Sp]$.

A different way to view the expressive power of a logic is to identify which regular languages it is capable of defining. It is known that MFOL[<] (equivalently, SDoO) is capable of defining precisely the star-free regular languages [12] and we have recently shown that MFOL[=] (equivalently SD) is capable of defining precisely the commutative star-free regular languages [15] (a language is commutative if it is closed under permutation). Thus, the table 4b can be rewritten in terms of expressiveness as compared to regular languages, where $\top$ defines $\Sigma^*$ and $\perp$ defines $\emptyset$ (the empty language).

**Table 4.** A summary of the presented results

| | $[CDNOSpSh]$ | $-C$ | $-D$ | $-N$ | $-O$ | $-Sp$ | $-Sh$ | |
|---|---|---|---|---|---|---|---|---|
| $-C$ | $=$ | – | | | $=$ | $=$ | $=$ | $-C$ |
| $-D$ | $=$ | | – | $<$ | $=$ | $=$ | $=$ | $-D$ |
| $-N$ | | | $<$ | – | $=$ | $<$ | $<$ | $-N$ |
| $-O$ | $<$ | $<$ | $<$ | $<$ | – | $<$ | $<$ | $-O$ |
| $-Sp$ | $=$ | $=$ | $=$ | $<$ | $<$ | – | $<$ | $-Sp$ |
| $-Sh$ | $=$ | $=$ | $=$ | $<$ | $=$ | | – | $-Sh$ |

(a) Summary of results shown in this section.

| | $[CDNOSpSh]$ | $-C$ | $-D$ | $-N$ | $-O$ | $-Sp$ | $-Sh$ | |
|---|---|---|---|---|---|---|---|---|
| $-C$ | MFOL[<] | – | | | MFOL[=] | **MFOL**[<] | **MFOL**[<] | $-C$ |
| $-D$ | MFOL[<] | | – | | MFOL[=] | **MFOL**[<] | **MFOL**[<] | $-D$ |
| $-N$ | | | | – | MFOL[=] | | **MFOL**[<] | $-N$ |
| $-O$ | MFOL[=] | MFOL[=] | MFOL[=] | MFOL[=] | – | MFOL | **MFOL**[=] | $-O$ |
| $-Sp$ | **MFOL**[<] | **MFOL**[<] | **MFOL**[<] | **MFOL**[<] | MFOL | – | $\top,\bot$ | $-Sp$ |
| $-Sh$ | **MFOL**[<] | **MFOL**[<] | **MFOL**[<] | | **MFOL**[=] | $\top,\bot$ | – | $-Sh$ |

(b) Expressiveness in terms of classes of symbolic logic.

## 4 Conclusion

The key results in this paper concern the relative expressiveness of fragments of spider diagrams of order. Perhaps surprisingly, we have shown that spiders and shading can each be removed from certain fragments whilst maintaining expressiveness. The model theoretic analysis we have provided for some of the fragments also provide insight into the kinds of statements that the diagrams in these fragments can make. Whilst we completed 14 of the 36 entries in table 1, 5 gaps remain. We conjecture that the two missing entries in the $-N$ row will be $<$, but this is not clear. A difficultly with analysing these two cases stems from the fact that there is no analogy to De Morgan's Laws for negation and $\lhd$. Thus, in fragments containing $N$ and $O$ , there are no obvious normal forms that explicitly reflect the semantics of the diagrams.

The proof strategies used throughout the paper are likely to adapt to other systems, such as the Euler/Venn logic. Whilst this logic is less expressive than spider diagrams, its strong syntactic similarity justifies our claim. Moreover, the kinds of results we have provided concerning when the removal of syntax impacts expressiveness may well provide a basis for similar conjectures in Euler/Venn and other related notations. We expect to use these results when developing more expressive notations based on SDoO: they will inform us about what syntax it is necessary to include. Our immediate plans involve extending SDoO to a monadic second order logic, since MSOL is capable of defining precisely the regular languages.

As well as providing insight into what can be expressed with the presence or absence of certain pieces of syntax, there are further benefits to this work concerning, for instance, the development of reasoning systems. For example, theorem 5 can be restated as a (syntactic) inference rule. Now consider a fragment, $F_1$, from which we can remove spiders using this inference rule. We can, therefore, immediately obtain a sound and complete inference system for $F_1$ provided $F_2$ is sound and complete, where $F_2$ is $F_1$ without spiders; SDoO is an example of such an $F_1$. Currently, there is no sound and complete set of inference rules for SDoO, so the results in this paper may aid in their development.

**Acknowledgements.** This research forms part of the the Defining Regular Languages with Diagrams (EPSRC grant EP/H012311/1) and Sketching Euler Diagrams (EPSRC grant EP/H048480/1) projects.

## References

1. Kent, S.: Constraint diagrams: Visualizing invariants in object oriented modelling. In: Proceedings of OOPSLA 1997, pp. 327–341. ACM Press, New York (1997)
2. Dau, F.: The Logic System of Concept Graphs with Negations: And its Relationship to Prediacte Logic. Springer, Heidelberg (2003)
3. Hammer, E.: Logic and Visual Information. CSLI Publications, Stanford (1995)
4. Swoboda, N., Allwein, G.: Using DAG transformations to verify Euler/Venn homogeneous and Euler/Venn FOL heterogeneous rules of inference. Journal on Software and System Modeling 3, 136–149 (2004)
5. Howse, J., Stapleton, G., Taylor, J.: Spider diagrams. LMS Journal of Computation and Mathematics 8, 145–194 (2005)
6. Shin, S.J.: The Logical Status of Diagrams. Cambridge University Press, Cambridge (1994)
7. Stapleton, G., Howse, J., Taylor, J.: A decidable constraint diagram reasoning system. Journal of Logic and Computation 15(6), 975–1008 (2005)
8. Delaney, A., Stapleton, G.: Spider diagrams of order. In: International Workshop on Visual Languages and Logic (September 2007)
9. Stapleton, G., Masthoff, J.: Incorporating negation into visual logics: A case study using Euler diagrams. In: Visual Languages and Computing 2007, pp. 187–194. Knowledge Systems Institute (2007)
10. Stapleton, G., Thompson, S., Howse, J., Taylor, J.: The expressiveness of spider diagrams. Journal of Logic and Computation 14(6), 857–880 (2004)
11. Delaney, A., Taylor, J., Thompson, S.: Spider diagrams of order and a hierarchy of star-free regular languages. In: Stapleton, G., Howse, J., Lee, J. (eds.) Diagrams 2008. LNCS (LNAI), vol. 5223, pp. 172–187. Springer, Heidelberg (2008)
12. Thomas, W.: Classifying regular events in symbolic logic. Journal of Computer and System Sciences 25, 360–376 (1982)
13. Ebbinghaus, H.D., Flum, J.: Finite Model Theory, 2nd edn. Springer, Heidelberg (1991)
14. Manzano, M.: Model Theory. Oxford University Press, Oxford (1999)
15. Delaney, A., Stapleton, G., Taylor, J., Thompson, S.: A diagrammatic characterisation of commutative star-free regular languages (in preparation)

# A Calculus for Graphs with Complement

Renata de Freitas[1], Paulo A.S. Veloso[2], Sheila R.M. Veloso[3], and Petrucio Viana[1]

[1] Institute of Mathematics, UFF: Universidade Federal Fluminense; Niterói, Brazil
[2] Systems and Computer Engin. Program, COPPE, UFRJ: Universidade Federal do Rio de Janeiro; Rio de Janeiro, Brazil
[3] Systems and Computer Engin. Dept., Faculty of Engineering, UERJ: Universidade do Estado do Rio de Janeiro; Rio de Janeiro, Brazil

**Abstract.** We present a system for deriving inclusions between graphs from a set of inclusions between graphs taken as hypotheses. The novel features are the extended notion of graph with an explicitly representation of complement, the more involved definition of the system, and its completeness proof due to the embedding of complements. This is an improvement on former work, where complement was introduced by definition. Our calculus provides a basis on which one can construct a wide range of graph calculi for several algebras of relations.

**Keywords:** Reasoning with Diagrams, Graph Calculus, Complement, Completeness, Algebras of Relations.

## 1 Introduction

Our understanding of basic *reasoning with diagrams* can be better grasped by observing Figure 1 (a). In this paper, we show how the basic idea underlying reasoning with diagrams can be nicely adapted to the case of "formulas" being "terms from a relation algebraic language" and "implies" meaning "the relation defined by term $t_1$ is a sub-relation of the one defined by term $t_2$, under a set $\Sigma$ of hypotheses". In this case, certainly due to the conceptual proximity between *binary relations* and *graphs*, the diagrams that appear as the most appropriate to deal with are the *2-pointed labeled directed graphs* [1], and the picture in Figure 1 (a) converts to that in Figure 1 (b).

Graphs in the sense above mentioned have been used or, rather, have been proposed to be used as a tool in the investigation of relation algebraic formalisms from a long time ago. Usually, the graphs in general — not only those obtained from terms — have bigger expressive and proof powers and are reputed to be easier to use than the algebraic terms. Hence, by using graphs instead of terms, one can obtain results on graphs that remain true when restricted to the terms. Examples of this strategy in use and some of its developments can be found in the papers [1,6,4,7,5]. One can say that all these works, except [6], use graphs as auxiliary tools into the investigation of subsystems of relation algebras [15,20].

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 84–98, 2010.

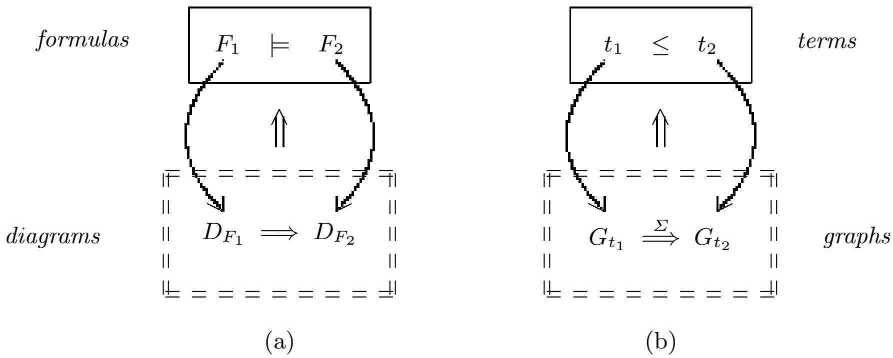



**Fig. 1.** Reasoning (a) with diagrams and (b) with graphs

Differently, Curtis and Lowe [6] suggest to investigate graph systems themselves as ordinary formal systems. Their main idea is to use the 2-pointed labeled directed graphs machinery as a formal system whose language has graphs as formulas and whose inference rules are applied to transform a graph into another. Under some general conditions, these transformation rules can be used to put the graphs into a certain normal form and the inclusions of graphs in normal form can be tested via an adequate notion of homomorphism between graphs. They also give hints of how to adapt their ideas to deal with a whole class of algebras built on the top of a lattice equipped with an associative binary operation compatible with the lattice operations.

We decided to investigate Curtis and Lowe ideas deeply and to develop logical systems having graphs as terms and inferences on graphs. In the previous papers [10,9,12,13] we investigate the use of graphs to decide the validities of languages representing just positive information. The next natural step was taken in [11] where we considered reasoning from hypotheses in a language where complementation is introduced by definition.

Here we present a system which improves the earlier ones on reasoning from hypotheses, having an explicit representation of complement. The basic intuitions are quite simple, leading to a playful and powerful system for deriving inclusions between graphs that are consequences of a set of inclusions between graphs taken as hypotheses. The novel features are the extended notion of graph with arcs labeled by boxes (to represent complement), the more involved definition of the system, and its completeness proof due to the embedding of complements. We leave for further investigation the use of our system to simplify previous reasoning about algebras of relations as well as to adapt our system to deal with algebras having a structured domain (cf. [11,12] for preliminary results in this direction).

The approach to reasoning with graphs we adopt here may be called the *logic systematic* approach: pictures are considered as ordinary terms of a (non-orthodox) logical system and a set of inference rules is provided for deriving pictures from pictures. This approach emphasizes notions of homomorphism for pictures, which are used to prove the inclusions and equalities. With no intention of being exhaustive, we would like to mention two other approaches in using pictures as a tool to help investigating and applying relational formalisms. The approach based on the *theory of allegories* [2,3,16] views pictures as arrows in a (unitary pretabular) allegory [8] and uses laws directly associated to the valid allegorical identities for transforming pictures. Results of the theory of allegories are used to show that two pictures can be proved equal by using the laws on pictures iff they represent the same relation. The approach based on the *rewriting systems* [17,18,19] endows pictures with a relational semantics, which allows them to be interpreted as terms of an algebraic language. A rewriting mechanism for pictures is built as a variant of the algebraic approach to graph rewriting. The way one can use rewriting sequences as proofs leads to a general and flexible tool for the proof of relational algebraic identities.

The structure of this paper is as follows. In Section 2, we present some intuitive ideas and examples to motivate the graph language and rules discussed in the other sections. In Section 3, we introduce more precisely our graphs, presenting its syntax and semantics, and defining the notion of consequence we deal with. We also present two schemes of axioms and two rules which can be used to transform a graph into another and prove soundness of the system. In Section 4, we indicate how one can characterize graph consequence by means of the axioms and rules, obtaining completeness of the calculus. Due to lack of space some minor details in the proofs were omitted.

## 2 Basic Ideas

We begin with some intuitive ideas behind our graph calculus, describing the aspect our 2-pointed labeled directed graphs have, how they can be used to represent relations, and how they can be transformed to represent inferences on relations. A graph is a finite set of slices. A slice consists of nodes, labeled arcs between nodes, and exactly two distinguished nodes we call input and output and represented by − and +, respectively. A label is a relation symbol or a box. A box is a figure of the form $\boxed{G}$, where $G$ is a graph. A box should be considered as a black-box, that means, the nodes, arcs, labels, and distinguished nodes of the graph it encloses do not count as nodes, arcs, labels, or distinguished nodes of the graph in which it occurs as a label. We identify a graph $\{S\}$, consisting of just one slice $S$, with the slice $S$.

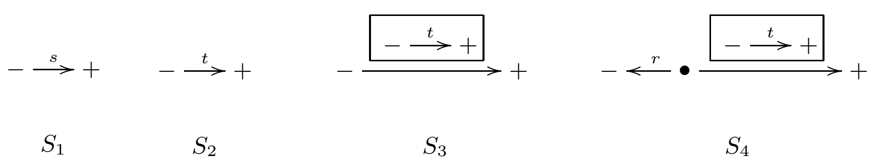



**Fig. 2.** Slices

Figure 2 shows slices $S_1$, $S_2$, $S_3$, and $S_4$. Slices $S_1$ and $S_2$ have single arcs, labeled by the relation symbols $s$ and $t$, respectively. Slice $S_3$ has a single arc labeled by the box $\boxed{S_2}$. Slice $S_4$ has two consecutive opposite arcs, one labeled by the relation symbol $r$ and the other labeled by the box $\boxed{S_2}$.

Figure 3 shows a single graph consisting of two slices, $S_5$ and $S_6$. Slice $S_5$ has two parallel paths from the non-distinguished node $\bullet$ to node the output $+$. One is the path $\bullet \xrightarrow{r} - \xrightarrow{s} +$ and the other is the arc $\bullet \xrightarrow{\boxed{S_2}} +$. Slice $S_6$ is like $S_5$, with arc $- \xrightarrow{\boxed{S_1}} +$ in place of arc $- \xrightarrow{s} +$.

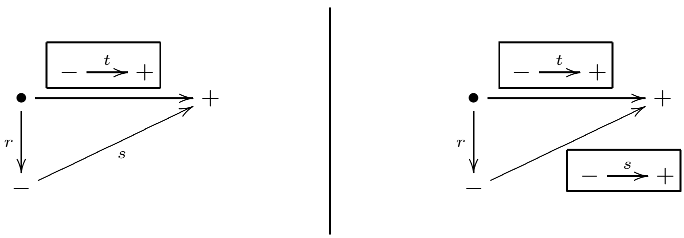



**Fig. 3.** Graph with two alternative slices $S_5$ and $S_6$, obtained from $S_4$

Now, we describe the way graphs, slices, and labels represent binary relations.

Given an arbitrary set $M$, a relation symbol represents an arbitrary binary relation on $M$. So, considering that the relation symbols $r$, $s$ and $t$, referred above, represent binary relations on a base set $M$, the slices and graph depicted in Figures 2 and 3 also represent binary relations on $M$, according to the following ideas.

A labeled arc from a node $u$ to a node $v$ with label $L$ represents a restriction imposed to $u$ and $v$, namely that $u$ and $v$ should be related by the relation $L$. A slice represents the set of pairs satisfying the restrictions imposed to its input-output nodes. Thus, $S_1$ represents the set of pairs related by $s$, i.e. the relation $s$. Analogously, $S_2$ represents relation $t$.

A box represents the complement of the relation the graph it encloses represents. Hence, since $S_3$ represents the set of pairs related by $\boxed{S_2}$, we have that $S_3$ represents the relation $t^{\mathsf{C}}$, the complement of $t$.

Consecutive arcs represent a concatenation of the restrictions each arc imposes on their input and output nodes, i.e. the "serialization" of the restrictions. Slice $S_4$ represents the set of pairs having an intermediate point $\bullet$ with which the input node is related by $r$ and that is related to the output node by $\boxed{S_2}$. Hence, $S_4$ represents the relation $r^{\mathsf{T}} \circ t^{\mathsf{C}}$, the conposition of the transpose of $r$ with the complement of $t$.

Parallel paths sharing the same extreme points represent simultaneous restrictions their extreme points should satisfy, i.e. the "parallelization" of the restrictions. So, $S_5$ represents the set of pairs related by both the relations $r^{\mathsf{T}} \circ t^{\mathsf{C}}$ and $s$, i.e. $S_5$ represents the intersection $(r^{\mathsf{T}} \circ t^{\mathsf{C}}) \cap s$. Similarly, $S_6$ represents the intersection $(r^{\mathsf{T}} \circ t^{\mathsf{C}}) \cap s^{\mathsf{C}}$.

In general, a slice imposes a set of restrictions on its input and output nodes that any pair of points in $M$ should satisfy to be in the relation the slice represents.

Finally, a graph represents the relation which is the union of the relations represented by its slices. Thus, the graph in Figure 3 represents the union $\{(r^{\mathsf{T}} \circ t^{\mathsf{C}}) \cap s\} \cup \{(r^{\mathsf{T}} \circ t^{\mathsf{C}}) \cap s^{\mathsf{C}}\}$.

Now, we describe, by way of an example, how our graphs can be manipulated to represent inferences on relations. The ideas here presented give the intuitions behind our graph calculus and will lead us to a set of inference rules which will characterize it.

It is known that, for all relations $r$, $s$ and $t$, the inclusion $r^{\mathsf{T}} \circ t^{\mathsf{C}} \subseteq s^{\mathsf{C}}$ follows from the inclusion $r \circ s \subseteq t$. Figure 4 is a proof of this fact in our graph calculus. It consists of a sequence $\langle G_1, G_2, G_3, G_4, G_5 \rangle$ of five graphs.

The graph $G_1$ is the single slice graph $S_4$ of Figure 2, that represents the relation $r^{\mathsf{T}} \circ t^{\mathsf{C}}$, i.e. $G_1$ represents the left-hand side of the inclusion we want to prove.

The graph $G_2$ is the two slices graph $\{S_5, S_6\}$ of Figure 3, obtained from graph $G_1$ by expanding slice $S_4$. Notice that slice $S_5$ and slice $S_6$ are obtained from slice $S_4$ by adding to it a new arc from $x_4$ to $y_4$, labeled by $s$ and the box $\boxed{S_1}$, respectively. To show that this passage from $S_4$ to $\{S_5, S_6\}$ is justified, we consider that, in general, a graph represents alternatives a pair of points may satisfy in order to belong to the relation defined by the graph. Also, as usual, any pair of points is related by a given relation or by its complement. Thus, slice $S_4$ and the graph $\{S_5, S_6\}$ impose the same restrictions to any pair of nodes, representing the same relation.

The graph $G_3$ is also a two slices graph, $\{S_7, S_6\}$, obtained this time from the graph $\{S_5, S_6\}$, according to the following idea that allow us to use the hypothesis $r \circ s \subseteq t$ to transform the slice $S_5$. Since slice $S_5$ has a path from node $\bullet$ to the output node $+$ through the input node $-$ that represents the relation $r \circ s$ and since, by hypothesis, $r \circ s \subseteq t$, we are allowed to transform slice $S_5$ into slice $S_7$ by adding an arc labeled $t$ from node $\bullet$ to the output node $+$.

Now, observe that slice $S_7$ of graph $G_3$ has two parallel arcs linking node $\bullet$ to the output node $+$, one labeled $t$ and the other labeled $\boxed{S_2}$. So, according to our conventions on parallel paths, the points $\bullet$ and $+$ are simultaneously in the relations $t$ and $t^{\mathsf{C}}$, so that slice $S_7$ represents the empty relation. Thus, we can erase slice $S_7$ from the graph $G_3$ obtaining the single slice graph $G_4$ which consists of the remaining slice $S_6$.

Finally, note that, inside graph $G_4$ we can locate a copy of the slice $S_8$, as shown in Figure 5. This means that $S_8$ imposes no more restrictions than $S_6$ in defining a relation and so, $S_6$ may be considered to represent a sub-relation of $S_8$. Thus, we finally move from graph $G_4$ to graph $G_5$ in constructing our graph

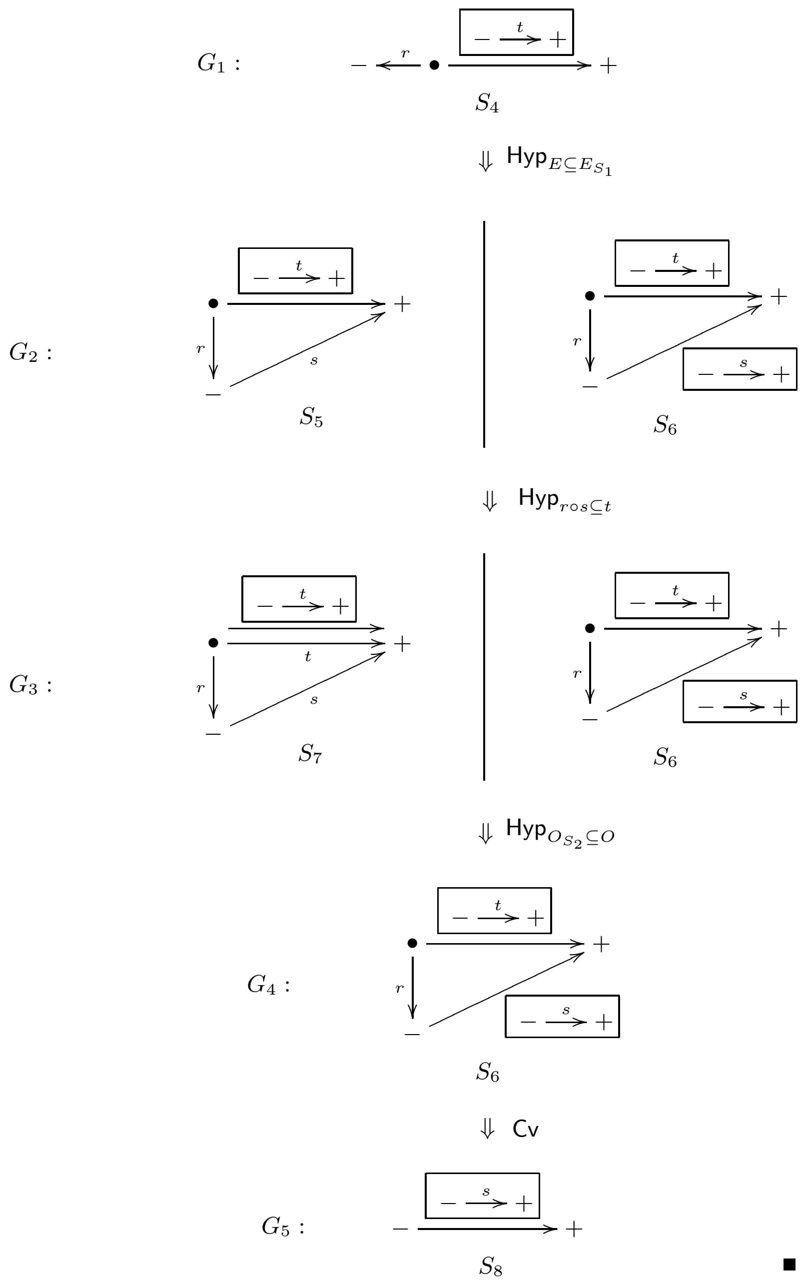



**Fig. 4.** Graph proof of $r \circ s \subseteq t \ \vdash \ r^{\mathsf{T}} \circ t^{\mathsf{C}} \subseteq s^{\mathsf{C}}$

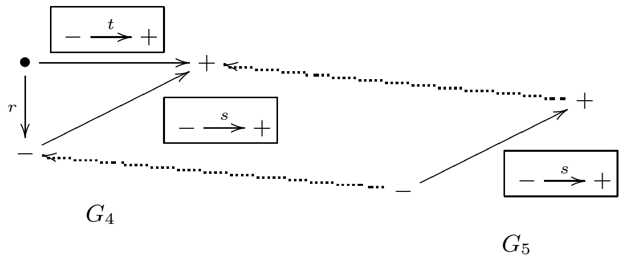



**Fig. 5.** Mapping slice $S_8$ to slice $S_6$

proof. The latter consists of a boxed slice which represents the relation $s^{\mathsf{C}}$, i.e. $G_5$ represents the right-hand side of the implication we want to prove.

Summarizing, we have ilustrated above how graphs can be used to prove a valid inference involving two relational inclusions, the latter having occurrences of complement. The usual setting where formal reasoning on relations is performed is *equational logic*. There, all the statements are equalities between relational terms, and the only primitive inference rule is the high-school rule of replacing equals by equals. To emphasize the playful aspect of our approach, we present below an equational proof of the above inference, based on the usual equational axioms for relations.

**Proposition 1.** *$r \circ s \subseteq t$ implies $r^{\mathsf{T}} \circ t^{\mathsf{C}} \subseteq s^{\mathsf{C}}$, for all relations $r$, $s$, and $t$.*

Proof. We use the equational reasoning (Con), which assures that $=$ is a congruence relation with respect to the operations on relations; we apply some usual Boolean properties (BA), the distributivity of $\circ$ on $\cup$ (Dis), and the awckward axiom $(r^{\mathsf{T}} \circ (r \circ s)^{\mathsf{C}}) \cup s^{\mathsf{C}} = s^{\mathsf{C}}$ (Ax), which plays an important role in the axiomatization of relation algebras [20].

| | | |
|---|---|---|
| 1. | $(r \circ s) \cap t = r \circ s$ | (Hyp) |
| 2. | $((r \circ s) \cap t)^{\mathsf{C}} = (r \circ s)^{\mathsf{C}}$ | (1, Con) |
| 3. | $(r \circ s)^{\mathsf{C}} \cup t^{\mathsf{C}} = (r \circ s)^{\mathsf{C}}$ | (2, BA) |
| 4. | $r^{\mathsf{T}} \circ ((r \circ s)^{\mathsf{C}} \cup t^{\mathsf{C}}) = r^{\mathsf{T}} \circ (r \circ s)^{\mathsf{C}}$ | (3, Con) |
| 5. | $(r^{\mathsf{T}} \circ (r \circ s)^{\mathsf{C}}) \cup (r^{\mathsf{T}} \circ t^{\mathsf{C}}) = r^{\mathsf{T}} \circ (r \circ s)^{\mathsf{C}}$ | (4, Dis) |
| 6. | $((r^{\mathsf{T}} \circ (r \circ s)^{\mathsf{C}}) \cup (r^{\mathsf{T}} \circ t^{\mathsf{C}})) \cup s^{\mathsf{C}} = (r^{\mathsf{T}} \circ (r \circ s)^{\mathsf{C}}) \cup s^{\mathsf{C}}$ | (5, Con) |
| 7. | $((r^{\mathsf{T}} \circ (r \circ s)^{\mathsf{C}}) \cup s^{\mathsf{C}}) \cup (r^{\mathsf{T}} \circ t^{\mathsf{C}}) = (r^{\mathsf{T}} \circ (r \circ s)^{\mathsf{C}}) \cup s^{\mathsf{C}}$ | (6, BA) |
| 8. | $s^{\mathsf{C}} \cup (r^{\mathsf{T}} \circ t^{\mathsf{C}}) = s^{\mathsf{C}}$ | (7, Ax) |

In the proof above we used the fact that, for relations, any inclusion $r \subseteq s$ is equivalent to the equalities $r \cap s = s$ and $s \cup r = r$. ∎

## 3 Graph Calculus

Now we formalize the intuitive ideas presented in Section 2 to define our graph calculus.

We start by presenting syntax. Our system has sets of 2-pointed labeled directed graphs as "terms" and inclusions between graphs as "formulas".

Nodes and labeled arcs are the building blocks of graphs. Hence, we first consider the sets INOD of individual *nodes* and RSYM of *relational symbols*, which we will keep fixed throughout.

A *slice*, typically denoted by $S$ or $T$, is a structure $(N, A, x, y)$, where $N$ is a finite nonempty set of nodes; $A \subseteq N \times \mathfrak{L} \times N$ is a finite set of labeled *arcs* ($\mathfrak{L}$ is the set of all labels); $x$ (*input*) and $y$ (*output*) are, not necessarily distinct, distinguished nodes in $N$. An *arc* of $A$ is a triple, denoted by $uLv$, with $u, v \in N$ and $L$ being a label. A *label* is a relational symbol or a *box* $\boxed{G}$, where $G$ is a concrete graph.

*Concrete graphs* are sets of slices defined by the following grammar.

$$G ::= \{S_L\} \mid E \mid I \mid O \mid G^{\mathsf{T}} \mid G \circ G \mid G \sqcap G \mid G \sqcup G,$$

where

$S_L = (\{x, y\}, \{xLy\}, x, y)$, with $x, y \in$ INOD and $L$ being a label,
$E = \{(\{x, y\}, \emptyset, x, y)\}$, with $x, y \in$ INOD and $x \neq y$,
$I = \{(\{x\}, \emptyset, x, x)\}$, with $x \in$ INOD,
$O = \emptyset$.

The operations on concrete graphs are defined based on their analogous to slices, except for union. Given slices $S = (N, A, x, y)$, $S_1 = (N_1, A_1, x_1, y_1)$, and $S_2 = (N_2, A_2, x_2, y_2)$, we define

$S^{\mathsf{T}} = (N, A, y, x)$, the *transposition* of $S$,
$S_1 \circ S_2 = (N_1 \uplus N_2, A_1 \uplus A_2, x_1, y_2)\frac{x_2}{y_1}$, the *composition* of $S_1$ and $S_2$,
$S_1 \sqcap S_2 = (N_1 \uplus N_2, A_1 \uplus A_2, x_1, y_1)\frac{x_1}{x_2}\frac{y_1}{y_2}$, the *intersection* of $S_1$ and $S_2$.

Here, we use the *node substitution* notation $\frac{u}{v}$ for replacing $u$ by $v$, which we extend naturally to sets as well as to tuples, e.g., for a set $A$ of arcs, we put $A\frac{u}{v} = \{w\frac{u}{v}Lz\frac{u}{v} : wLz \in A\}$.

Given concrete graphs $G = \{S_i : i \in I\}$ and $H = \{T_j : j \in J\}$, we define

$G^{\mathsf{T}} = \{{S_i}^{\mathsf{T}} : i \in I\}$, the *transposition* of $G$,
$G \circ H = \{S_i \circ T_j : i \in I, j \in J\}$, the *composition* of $G$ and $H$,
$G \sqcap H = \{S_i \sqcap T_j : i \in I, j \in J\}$, the *intersection* of $G$ and $H$,
$G \sqcup H = G \cup H$, the *union* of $G$ and $H$.

Slices $S_1 = (N_1, A_1, x_1, y_1)$ and $S_2 = (N_2, A_2, x_2, y_2)$ are *isomorphic* if there are bijections $f : N_1 \to N_2$ and $g : A_1 \to A_2$ such that

1. for all $urv \in A_1$, $g(urv) = furfv$,
2. for all $u\boxed{G}v \in A_1$, $g(u\boxed{G}v) = fu\boxed{G'}fv$ and $G$ and $G'$ are isomorphic,
3. $fx_1 = x_2$ and $fy_1 = y_2$.

Concrete graphs $G$ and $H$ are *isomorphic* if there is a bijection $h : G \to H$ such that $h(S)$ is isomorphic to $S$, for all $S \in G$. The usual identification of isomorphic concrete graphs is reflected in our figures by the representation of every non-distinguished node by $\bullet$, of every input node by $-$, and of every output node by $+$.

A *graph* is an equivalence class of isomorphic concrete graphs. In what follows, a graph is identified with each of the concrete graphs that represents the equivalence class.

A *graph inclusion* is an expression of the form $G \sqsubseteq H$.

We now move on to semantics. Given a base set $M$, the labels, slices, and graphs will denote binary relations on $M$. To define this in a proper way we need the notions of a model and an assignment of individual nodes.

A *model*, typically denoted by $\mathfrak{M}$, is a structure $\langle M, \{r^{\mathfrak{M}} : r \in \text{RSYM}\}\rangle$, where $M \neq \emptyset$ is the *universe* of $\mathfrak{M}$ and $r^{\mathfrak{M}} \subseteq M \times M$, for every $r \in$ RSYM. The *meaning* of a label $L$ in a model $\mathfrak{M}$, denoted by $[\![L]\!]_{\mathfrak{M}}$, is defined by $[\![r]\!]_{\mathfrak{M}} = r^{\mathfrak{M}}$ and $\left[\!\!\left[\boxed{G}\right]\!\!\right]_{\mathfrak{M}} = [\![G]\!]^{\complement}_{\mathfrak{M}}$, the complement of $\left[\!\!\left[\boxed{G}\right]\!\!\right]_{\mathfrak{M}}$. The *meaning* of a graph $G$ in a model $\mathfrak{M}$, denoted by $[\![G]\!]_{\mathfrak{M}}$, is defined in Table 1, where $R^{-1}$ is the *transpose* of relation $R$, i.e. $R^{-1} = \{(a,b) \in M \times M : (b,a) \in R\}$, and $R_1 \mid R_2$ is the *composition* of relations $R_1$ and $R_2$, i.e $R_1 \mid R_2 = \{(a,b) \in M \times M : (a,c) \in R_1 \text{ and } (c,b) \in R_2, \text{ for some } c \in M\}$.

**Table 1.** Meaning of graphs

| | |
|---|---|
| $[\![\{S_L\}]\!]_{\mathfrak{M}} = [\![L]\!]_{\mathfrak{M}}$ | $[\![G^{\mathsf{T}}]\!]_{\mathfrak{M}} = [\![G]\!]_{\mathfrak{M}}{}^{-1}$ |
| $[\![E]\!]_{\mathfrak{M}} = M \times M$ | $[\![G \circ H]\!]_{\mathfrak{M}} = [\![G]\!]_{\mathfrak{M}} \mid [\![H]\!]$ |
| $[\![O]\!]_{\mathfrak{M}} = \emptyset$ | $[\![G \sqcap H]\!]_{\mathfrak{M}} = [\![G]\!]_{\mathfrak{M}} \cap [\![H]\!]$ |
| $[\![I]\!]_{\mathfrak{M}} = \{(a,b) \in M \times M : a = b\}$ | $[\![G \sqcup H]\!]_{\mathfrak{M}} = [\![G]\!]_{\mathfrak{M}} \cup [\![H]\!]$ |

As usual, we introduce a notion of consequence between a set of graph inclusions and a graph inclusion based on meaning. We say that a model $\mathfrak{M}$ *verifies* a graph inclusion $G \sqsubseteq H$, denoted by $\mathfrak{M} \models G \sqsubseteq H$, iff $[\![G]\!]_{\mathfrak{M}} \subseteq [\![H]\!]_{\mathfrak{M}}$. We say that a graph inclusion is *valid*, denoted by $\models G \sqsubseteq H$, iff it is verified by any model. We say that a model $\mathfrak{M}$ *verifies* a set $\Gamma$ of graph inclusions, denoted by $\mathfrak{M} \models \Gamma$, iff $\mathfrak{M}$ verifies every graph inclusion in $\Gamma$. We say that a graph inclusion $G \sqsubseteq H$ is a *consequence* of a set $\Gamma$ of graph inclusions, denoted by $\Gamma \models G \sqsubseteq H$, iff $\mathfrak{M} \models G \sqsubseteq H$ whenever $\mathfrak{M} \models \Gamma$, for every model $\mathfrak{M}$. As usual, we have $\models G \sqsubseteq H$ iff $\emptyset \models G \sqsubseteq H$.

Now we present a set of valid inclusions and a set of rules to transform a graph into another. These rules will preserve meaning when applied to graphs.

We first introduce two families of valid inclusions. These will play the role of axioms in our graph calculus. Recall $O = \emptyset$, the empty graph, and $E = \{(\{x,y\}, \emptyset, x, y)\}$, the graph consisting of one arcless slice with two distinct nodes, input $x$ and output $y$. Given a slice $S = (N_S, A_S, x_S, y_S)$, we define two graphs as follows. The graph $O_S = \{(N_S, A_S \cup \{x_S \boxed{S} y_S\}, x_S, y_S)\}$ is obtained from $S$ by adding to it a new arc from the input of $S$ to the output of $S$ labeled by $\boxed{S}$. The graph $E_S = \{S, (\{x, y\}, \{x \boxed{S} y\}, x, y)\}$ is obtained from $S$ by adjoining to $S$ a new slice with two distinct nodes, input $x$ and output $y$, and a single arc $x \boxed{S} y$ (Figure 6).

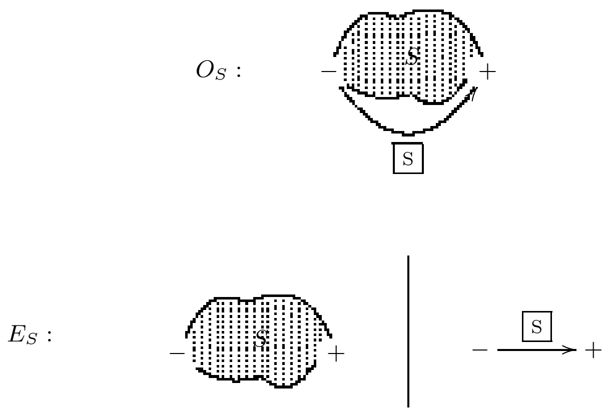



**Fig. 6.** Graphs $O_S$ and $E_S$

We shall take as schemes of axioms of the graph calculus the inclusions $O_S \sqsubseteq O$ and $E \sqsubseteq E_S$ for any every $S$ (Table 2). It follows immediately from the definitions that these inclusions are valid.

**Table 2.** Axioms

| | | |
|---|---|---|
| $O_S \sqsubseteq O$ | and | $E \sqsubseteq E_S$ |

**Lemma 1.** $\mathfrak{M} \models \{O_S \sqsubseteq O, E \sqsubseteq E_S\}$*, for every model* $\mathfrak{M}$ *and every slice* $S$*.*

The rules of our graph calculus are Graph Cover rule, Hypothesis rule and Box rule. Graph Cover rule is used to compare graphs with respect to inclusion, Hypothesis rule, to transform graphs according to the set of inclusions taken as hypotheses, and Box rule, to simplify the inner structure of box labels.

To define our first transformation rule, the concepts of homomorphism from a slice to another and that of a graph covering another will be crucial. Given slices $S = (N_S, A_S, x_S, y_S)$ and $T = (N_T, A_T, x_T, y_T)$, by a *slice homomorphism from* $T$ *to* $S$ we mean a function $\phi : N_T \to N_S$, denoted by $\phi : T \to S$, that preserves input, output, and arcs, i.e. $\phi x_T = x_S$, $\phi y_T = y_S$, and if $uLv \in A_T$ then $\phi u L \phi v \in A_S$. Given graphs $G$ and $H$, we say that $H$ *covers* $G$ or $G$ *is covered by* $H$, denoted by $G \leftarrow H$, iff for each slice $S \in G$ there exist a slice $T \in H$ and a slice homomorphism $\phi : T \to S$.

Rule Cv (Table 3) allows us to replace a graph by another one that covers it. The next result, showing that covering preserves meaning, i.e. that rule Cv is sound, follows from the fact that a slice homomorphism transfers assignments by composition.

**Table 3.** Graph Cover rule

$$\mathsf{Cv}\ \frac{G}{H} \quad \text{if} \ \ G \leftarrow H$$

**Lemma 2.** *If $G \leftarrow H$, then $[\![G]\!]_{\mathfrak{M}} \subseteq [\![H]\!]_{\mathfrak{M}}$, for every model $\mathfrak{M}$.*

We now introduce the concepts of gluing slices and draft homomorphism between slices, which will be central in applying a graph inclusion to transform a graph into another.

Intuitively, we glue slice $T$ onto slice $S$ by adding to $S$ a copy of $T$ and identifying designated nodes $u, v$ of $S$ to the input and output of $T$. More precisely, given slices $S = (N_S, A_S, x_S, y_S)$ and $T = (N_T, A_T, x_T, y_T)$, as well as designated nodes $u, v \in N_S$, the result of *gluing* $T$ onto $S$ via $u, v$ is the slice defined by $\mathsf{glue}_{(u,v)}(T, S) = (N_S \uplus N_T, A_S \uplus A_T, x_S, y_S)\frac{x_T}{u}\frac{y_T}{v}$. We *glue* a graph $H$ onto a slice $S$, via nodes $u, v$ of $S$, by gluing its slices to $S$, i.e. $\mathsf{glue}_{(u,v)}(H, S) = \{\mathsf{glue}_{(u,v)}(T, S) : T \in H\}$ (Figure 7).

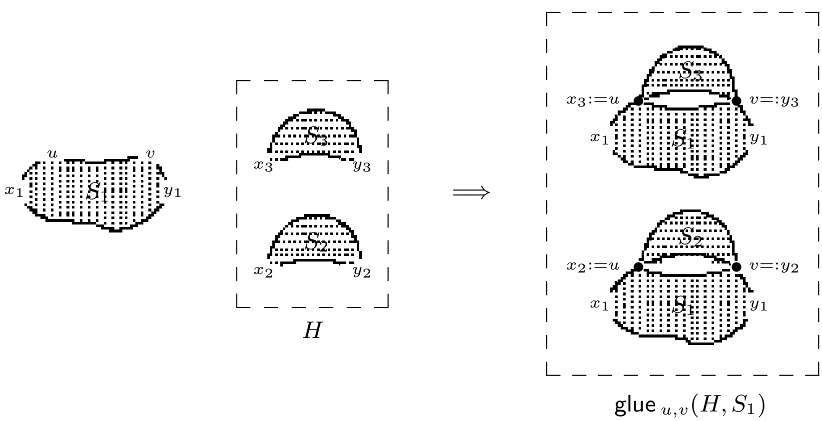



**Fig. 7.** Gluing $H$ in $S_1$ via $u, v$

Given slices $S = (N, A, x, y)$ and $S' = (N', A', x', y')$, by a *draft homomorphism from* $S'$ *to* $S$ we mean a function $\theta : N' \to N$, denoted by $\theta : S' \dashrightarrow S$, that preserves arcs. Now, given slices $S$ and $S'$ as before, a draft homomorphism $\theta : S' \dashrightarrow S$, and graph $H$, we set $\mathsf{glue}_\theta(H, S) = \mathsf{glue}_{(\theta x', \theta y')}(H, S)$. Rule $\mathsf{Hyp}_\Gamma$ (Table 4) allows us to glue a graph $H$ onto a slice $S$ of a graph under a draft homomorphism $\theta : S' \dashrightarrow S$ when $G' \cup \{S'\} \sqsubseteq H$ is a hypothesis in $\Gamma$ or is an axiom.

**Table 4.** Hypothesis rule

$$\mathsf{Hyp}_\Gamma \quad \frac{G \cup \{S\}}{G \cup \mathsf{glue}_\theta(H, S)}$$

if $\theta : S' \dashrightarrow S$ and $G' \cup \{S'\} \sqsubseteq H$ is in $\Gamma$ or is an axiom

The next result, showing that gluing preserves meaning, i.e. that rule $\mathsf{Hyp}_\Gamma$ is sound, follows from the fact that draft homomorphisms transfer assignments by composition.

**Lemma 3.** *For all slices* $S$ *and* $S'$ *and draft homomorphism* $\theta : S' \dashrightarrow S$*, if* $\mathfrak{M} \models \{S'\} \sqsubseteq H$*, then* $\mathfrak{M} \models \{S\} \sqsubseteq \mathsf{glue}_\theta(H, S)$*, for every model* $\mathfrak{M}$*.*

The Box rule (Table 5) is a two-way rule, i.e. it can be applied in the top-down and in the bottom-up directions. In the top-down direction, the Box rule allows us to replace an arc labeled by a box $\boxed{H}$ having a graph $H = \{S_i : i \in I\}$ inside of it, by a set $\{u \boxed{S_i} v : i \in I\}$ of parallel arcs, each one labeled by a box with the unary slice graph $S_i$ inside of it, and vice-versa, for the bottom-up direction. In the case $I = \emptyset$, the Box rule allows us to erase (top-down) or to add (bottom-up) an arc labeled by a box with the empty graph $O$ inside of it. Our calculus is heavily based on slice homomorphism, but a graph is a finite set of slices. Thus the De Morgan's laws expressed by our Box rule, in the top-down direction, allows one to obtain graphs where boxes have singleton graphs, preparing the application of the Cover rule.

Soundness of the Box rule follows immediately from the definitions of meaning of box label and meaning of graphs.

**Table 5.** Box rule

$$\mathsf{Box} \quad \frac{G \cup \{(N, A \cup \{u \boxed{\{S_i : i \in I\}} v\}, x, y)\}}{G \cup \{(N, A \cup \{u \boxed{S_i} v : i \in I\}, x, y)\}}$$

**Lemma 4.** *For every model* $\mathfrak{M}$*, it follows that*

$$[\![(N, A \cup \{u \boxed{\{S_i : i \in I\}} v\}, x, y)]\!]_{\mathfrak{M}} = [\![\{(N, A \cup \{u \boxed{S_i} v : i \in I\}, x, y)\}]\!]_{\mathfrak{M}}.$$

The notion of derivation is standard as in the rewriting systems for equational proofs. A proof of a graph inclusion $G \sqsubseteq H$ from a set of hypotheses is obtained starting with $G$ and applying our derivation rules and the hypotheses for rewriting $G$ until we obtain $H$. Given a set of graph inclusions $\Gamma$, by a *derivation from* $\Gamma$, or simply a $\Gamma$*-derivation*, we mean a sequence $(G_0, \ldots, G_n)$ of graphs such that each graph $G_i$, for $i \in \{1, \ldots, n\}$, is obtained from graph $G_{i-1}$ by an application of one of the rules $\mathsf{Cv}$, $\mathsf{Hyp}_\Gamma$, or $\mathsf{Box}$. A graph $H$ is *derivable from a graph* $G$ *using* $\Gamma$, or simply $H$ *is* $\Gamma$*-derivable from* $G$, denoted by $\Gamma \vdash G \sqsubseteq H$, when there is a $\Gamma$-derivation $(G_0, \ldots, G_n)$ such that $G_0 = G$ and $G_n = H$. An inclusion $G \sqsubseteq H$ is a *theorem*, denoted by $\vdash G \sqsubseteq H$, when $H$ is derivable from $G$ using the empty set of hypotheses.

The Box rule gives us a normal form for graphs.

**Lemma 5.** *For all graph* $G$*, there is a graph* $\mathsf{NF}G$ *such that* $\vdash G \sqsubseteq \mathsf{NF}G$*,* $\vdash \mathsf{NF}G \sqsubseteq G$*, and every box label occurring in* $\mathsf{NF}G$ *encloses a singleton graph.*

## 4 Completeness of the Graph Calculus with Complement

Soundness of our graph calculus follows from Lemmas 1, 2, 3, and 4. We will give the general idea of the completeness proof . We may assume that every box label have a unary graph inside, based on Lemma 5.

Given a set of graph inclusions $\Gamma$ and a graph inclusion $G \sqsubseteq H$, follow the procedure.

> Step 0. $G_0 := G$.
> Step $i+1$. Either $G_i \leftarrow H$, then stop, or else there is a slice $S \in G_i$ such that $\{S\} \not\leftarrow H$. Write $G_i = G' \cup \{S\}$. Take $G_i := G' \cup \mathsf{glue}_S(\theta, H')$, where $\theta : S' \dashrightarrow S$ and $G' \cup \{S'\} \sqsubseteq H' \in \Gamma$ or is an axiom.

If the procedure ever stops, the sequence $(G_0, \ldots, G_n)$ is a $\Gamma$-derivation of $G \sqsubseteq H$. Otherwise, we have a directed chain of slices $(S_n)_{n \in \mathbb{N}}$ such that, for all $i \in \mathbb{N}$, $\{S_i\}$ is not covered by $H$, and there is a slice homomorphism from $S_i$ to $S_{i+1}$. In fact, for each $i \in \mathbb{N}$, there is a slice $S_i \in G_i$ with $\{S\} \not\leftarrow H$ and a slice $S_{i+1} \in G_{i+1}$ with $\{S_{i+1}\} \not\leftarrow H$, for $G_i = G' \cup \{S_i\}$, $G_{i+1} = G' \cup \mathsf{glue}_{S_i}(\theta, H')$ and $S_{i+1} \in \mathsf{glue}_{S_i}(\theta, H')$. Since $S_{i+1} \in \mathsf{glue}_{S_i}(\theta, H')$, there is a slice homomorphism $\phi : S_i \to S_{i+1}$.

The *canonical model* $\mathfrak{M}^c = \langle \tilde{N}_*, \{r^{\mathfrak{M}^c} : r \in \textsc{Rsym}\}\rangle$ is obtained by a direct limit on slice chain $(S_n)_{n \in \mathbb{N}}$. More explicitly, form the set $N_*$ of all nodes occurring in the slices of the chain, and define the (equivalence) relation $\sim$ on it as follows: given nodes $u \in S_i$ and $v \in S_j$, set $u \sim v$ iff, for some $k \geq i, j$, $\phi^i_k u = \phi^j_k v$. Take $\tilde{N}_*$, the quotient set of $N_*$ by $\sim$, with the natural quotient map $\nu : N_* \to \tilde{N}_*$.

Now, interpret each relation symbol naturally by setting $(\tilde{u}, \tilde{v}) \in r^{\mathfrak{M}^c}$ iff there exist $n \in \mathbb{N}$, $u' \sim u$, $v' \sim v$, and an arc $u'rv' \in A_{S_n}$.

We then have the Satisfiability Lemma, whose proof will be omitted, by the lack of space.

**Lemma 6 (Satisfiability).** *Consider the canonical model $\mathfrak{M}^c$ for the graph inclusion $G \sqsubseteq H$ with set of hypotheses $\Gamma$. Hence, (1) $\mathfrak{M}^c \models \Gamma$ and (2) $\mathfrak{M}^c \not\models G \sqsubseteq H$.*

We thus have completeness of our graph calculus.

**Theorem 1 (Completeness).** *If $\Gamma \models G \sqsubseteq H$, then $\Gamma \vdash G \sqsubseteq H$.*

Proof. Suppose $\Gamma \models G \sqsubseteq H$. Hence, there is no (counter-)model $\mathfrak{M}$ such that $\mathfrak{M} \models \Gamma$ and $\mathfrak{M} \not\models G \sqsubseteq H$. Hence, the procedure stops. Hence, $\Gamma \vdash G \sqsubseteq H$. ∎

From the considerations above, we have that whenever $\Gamma \models G \sqsubseteq H$, there is a derivation in a normal form, i.e. a derivation consisting of a sequence of applications of Box followed by a sequence of applications of $\mathsf{Hyp}_\Gamma$ followed by a single application of Cv followed by a sequence of applications of Box.

**Corollary 1 (Normal form of derivations).** *Given a set $\Gamma \cup \{G \sqsubseteq H\}$ of graph inclusions, if $\Gamma \models G \sqsubseteq H$, then there are graphs $G_1, G_2, G_3$ with all box label having a unary graph inside such that $G \overset{\mathsf{Box}^*}{\Longrightarrow} G_1 \overset{\mathsf{Hyp}^*_\Gamma}{\Longrightarrow} G_2 \leftarrow G_3 \overset{\mathsf{Box}^*}{\Longrightarrow} H$.*

Our calculus can be adapted to be used with graphs whose labels are more complex, i.e. we can assume our labels are terms generated from relational symbols by application of relational algebraic operators as the Boolean, Peircean and others. For instance, the operations complement, union, intersection, composition, and reversion, whose meaning are the expected ones, as well as constant terms interpreted in a model whose universe is $M$ as $M \times M$, the identity and the empty relations. For this, one has to provide meaning preserving transformation rules to simplify the labels of the slice to boxes (with slices inside) and atomic relations.

**Acknowledgment.** We are indebted to the anonymous referees for valuable comments which enhanced the quality of the paper. In particular, we would like to thank the first referee for the very interesting remarks he/she has risen.

## References

1. Andréka, H., Bredikhin, D.A.: The equational theory of union-free algebras of relations. Algebra Universalis 33, 516–532 (1995)
2. Brown, C., Hutton, G.: Categories, allegories and circuit design. LICS 94, 372–381 (1994)
3. Brown, C., Jeffrey, A.: Allegories of circuits. In: Proc. Logical Foundations of Computer Science, pp. 56–68. St. Petersburg (1994)

4. Bredikhin, D.A.: On clones generated by primitive-positive operations of Tarskis relation algebras. Algebra Universalis 38, 165–174 (1997)
5. Cantone, D., Formisano, A., Omodeo, E.G., Zarba, C.G.: Compiling dyadic specifications into map algebra. Theoret. Comput. Sci. 293, 447–475 (2003)
6. Curtis, S., Lowe, G.: Proofs with graphs. Sci. Comput. Programming 26, 197–216 (1996)
7. Formisano, A., Omodeo, E.G., Simeoni, M.: A graphical approach to relational reasoning. ENTCS 44, 1–22 (2003)
8. Freyd, P.J., Scedrov, A.: Categories, Allegories. Elsevier, Amsterdam (1990)
9. de Freitas, R., Viana, P.: A note on proofs with graphs. Sci. Comput. Programming 73, 129–135 (2008)
10. de Freitas, R., Veloso, P.A.S., Veloso, S.R.M., Viana, P.: On positive relational calculi. Logic J. IGPL 15, 577–601 (2007)
11. de Freitas, R., Veloso, P.A.S., Veloso, S.R.M., Viana, P.: On a graph calculus for algebras of relations. In: Hodges, W., de Queiroz, R. (eds.) WoLLIC 2008. LNCS (LNAI), vol. 5110, pp. 298–312. Springer, Heidelberg (2008)
12. de Freitas, R., Veloso, P.A.S., Veloso, S.R.M., Viana, P.: Positive fork graph calculus. In: Artemov, S., Nerode, A. (eds.) LFCS 2009. LNCS, vol. 5407, pp. 152–163. Springer, Heidelberg (2008)
13. de Freitas, R., Veloso, P.A.S., Veloso, S.R.M., Viana, P.: On graph reasoning. Information and Computation 207, 1000–1014 (2009)
14. Grätzer, G.: Universal Algebra. Springer, New York (1979)
15. Hirsch, R., Hodkinson, I.: Relation Algebras by Games. Elsevier, Amsterdam (2002)
16. Hutton, G.: A relational derivation of a functional program. In: Lecture Notes of the STOP Summer School on Constructive Algorithms (1992)
17. Kahl, W.: Algebraic graph derivations for graphical calculi. In: D'Amore, F., Marchetti-Spaccamela, A., Franciosa, P.G. (eds.) WG 1996. LNCS, vol. 1197, pp. 224–238. Springer, Heidelberg (1997)
18. Kahl, W.: Relational treatment of term graphs with bound variables. Logic J. IGPL 6, 259–303 (1998)
19. Kahl, W.: Relational matching for graphical calculi of relations. Inform. Sciences 119, 253–273 (1999)
20. Maddux, R.: Relation Algebras. Elsevier, Amsterdam (2006)

# Two Types of Diagrammatic Inference Systems: Natural Deduction Style and Resolution Style

Koji Mineshima, Mitsuhiro Okada, and Ryo Takemura

Department of Philosophy, Keio University, Japan
{minesima,mitsu,takemura}@abelard.flet.keio.ac.jp

**Abstract.** Since the 1990s, reasoning with Venn and Euler diagrams has been studied from mathematical and logical viewpoints. The standard approach to a formalization is a "region-based" approach, where a diagram is defined as a set of regions. An alternative is a "relation-based" approach, where a diagram is defined in terms of topological relations (inclusion and exclusion) between circles and points. We compare these two approaches from a proof-theoretical point of view. In general, diagrams correspond to formulas in symbolic logic, and diagram manipulations correspond to applications of inference rules in a certain logical system. From this perspective, we demonstrate the following correspondences. On the one hand, a diagram construed as a set of regions corresponds to a disjunctive normal form formula and the inference system based on such diagrams corresponds to a resolution calculus. On the other hand, a diagram construed as a set of topological relations corresponds to an implicational formula and the inference system based on such diagrams corresponds to a natural deduction system. Based on these correspondences, we discuss advantages and disadvantages of each framework.

## 1 Introduction

Proof theory in logic has traditionally been developed based on linguistic (symbolic) representations of logical proofs. Recently, logical reasoning based on diagrammatic or graphical representations has been investigated by many logicians. In particular, Euler diagrams, introduced in the 18th century to illustrate syllogistic reasoning, began to be studied in the 1990s from a mathematical and formal logical viewpoint. However, until now, proof theory of Euler diagrams has not been that well developed.

In literature on diagrammatic reasoning, Euler diagrams have been formalized based on the method developed in the study on Venn diagrams. A Venn diagram is abstractly defined as a set of regions, where some of them may be shaded. In the same way, an Euler diagram is defined by considering shaded regions of a Venn diagram as "missing" regions. (E.g., Howse et al. [8]; for a survey, see Stapleton [19].) Thus, both Venn and Euler diagrams are abstractly defined in terms of regions, and hence we call this framework a "region-based" framework. Moreover, the inference rule of *unification*, which plays a central role in Euler diagrammatic reasoning, is defined by means of superpositions of Venn diagrams. The operation

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 99–114, 2010.

of superposition is uniformly defined for any two Venn diagrams that have the same circles as a simple union operation of shaded regions of the given diagrams. This uniformity of superposition produces an effectiveness that makes it easy to control theorem proving using diagrams, see, e.g. Stapleton et al. [20,5].

Nevertheless, the superposition rule has some disadvantages. In particular, by making a detour to Venn diagrams, some redundant steps are introduced in formalizing simple processes employed in Euler diagrammatic reasoning. For example, in order to derive $\mathcal{E}$ of Fig. 1 from given diagrams $\mathcal{D}_1$ and $\mathcal{D}_2$, they are first transformed into Venn diagrams $\mathcal{D}_1^v$ and $\mathcal{D}_2^v$ of Fig. 2, respectively; then, by superposing the shaded regions of $\mathcal{D}_1^v$ and $\mathcal{D}_2^v$, and by erasing the circle $B$, the Venn diagram $\mathcal{E}^v$ is obtained, which is transformed into the region-based Euler diagram $\mathcal{E}$. Thus, within the region-based framework, it is difficult to capture the appropriate notion of Euler diagrammatic proofs, in particular that of "normal diagrammatic proofs." Accordingly, the notion of proof normalization, which plays an essential role in proof theory, has not been developed to date for diagrammatic proofs.

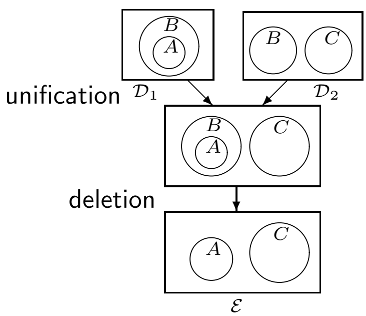



**Fig. 1.** Syllogism with Euler diagrams

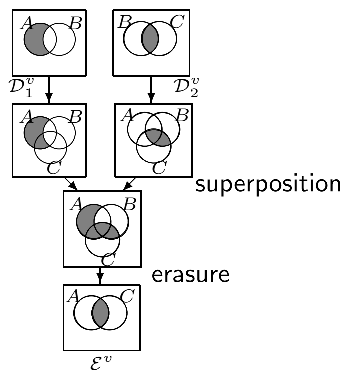



**Fig. 2.** Syllogism with Venn diagrams

In contrast to studies elaborated using this region-based framework, we introduced in [11,9] an Euler diagrammatic reasoning system, where diagrams are defined in terms of topological relations (inclusion and exclusion relations) between circles and points. We describe our approach as "relation-based." We formalized the unification of Euler diagrams in the manner developed in natural deduction, a well-known formalization of logical reasoning in traditional symbolic logic. Inference rules of Gentzen's natural deduction are intended to be as close as possible to actual reasoning ([6]). Along similar lines, our unification rule is designed to be as natural as possible to reflect intuitive manipulations of Euler diagrams as seen in Fig. 1.

In this paper, we discuss from a proof-theoretical viewpoint the following contrast between the two frameworks: At the representation level, diagrams abstractly defined in the relation-based framework correspond to implicational formulas, whereas diagrams defined in the region-based framework correspond to disjunctive normal form formulas. At the inference level, unification rules in the relation-based framework correspond to natural deduction style inference rules associated with the implicational connective, whereas inference rules in the region-based framework, in particular the erasure rule, correspond to the resolution principle. Thus, an inference system in the relation-based framework corresponds to a natural deduction system, and an inference system in the region-based framework corresponds to a resolution calculus. The contrast between the two frameworks is summarized in the following table:

| | | Relation-based framework (Euler diagrams) | Region-based framework (Venn diagrams) |
|---|---|---|---|
| Representation | Diag. | topological relations | regions with shading |
| | *Ling.* | *implicational formulas* | *disjunctive normal formulas* |
| Inference | Diag. | unification and deletion | superposition and erasure |
| | *Ling.* | *natural deduction* | *resolution calculus* |

These correspondences enable us to apply well-developed techniques used in traditional proof theory within symbolic logic in the field of diagrammatic reasoning. For our Euler diagrammatic inference system in the relation-based framework, we introduce the notion of a *normal diagrammatic proof*, i.e., a proof in which unification and deletion appear alternately. We show the normalization theorem of [11] by using the correspondence theorem between our Euler diagrammatic inference system and a natural deduction system. The normalization theorem is used to show that each chain of traditional Aristotelian categorical syllogisms corresponds to a normal diagrammatic proof in our system (cf. [9]).

The rest of this paper is organized as follows. In Section 2, we summarize the contrast between natural deduction and resolution. In Section 3, we show that our Euler diagrammatic inference system in the relation-based framework corresponds to a natural deduction system (Theorems 1 and 2). As a corollary, we show a normalization theorem for Euler diagrammatic proofs (Corollary 1). In Section 4, we show that a Venn diagrammatic inference system in the region-based framework corresponds to a resolution calculus (Theorems 3 and 4). In Section 5, we discuss advantages and disadvantages of the relation-based and the region-based frameworks.

## 2 Natural Deduction and Resolution

**Natural deduction.** Natural deduction was introduced by Gentzen [6], and studied extensively by Prawitz [14]. Natural deduction is one of the major logical inference systems for propositional and first-order logic in proof theory. Gentzen wrote of natural deduction "(Engl. Transl.:) . . . I intended first to set up a formal system which comes as close as possible to actual reasoning." ([6, p.68].) Natural deduction is applied in various areas and not just in mathematical logic. Indeed, some cognitive psychologists in their studies on Mental

Logic or Formal Rule Theory have admitted the naturalness of the logical inference rules, and adopted it as a theoretical basis for their studies (e.g. Rips [15], Braine-O'Brien [1]). See, for example, [14,2] for detailed descriptions of natural deduction. In natural deduction, inference rules are defined over the full syntax of propositional/first-order logic and defined as primitive as possible. In particular, the introduction rule of each connective can be regarded as the definition of the operational meaning of the connective, and the elimination rule, which defines the use of the connective, is justifiable by the definition. Thus, their operational meaning and validity can be immediately grasped by the definition. Furthermore, the normalization theorem for natural deduction plays an essential role in the development of proof-theory. Most of the proof-theoretical results depend on the theorem. We also use the theorem to show the correspondence between natural deduction and our Euler diagrammatic inference system in Section 3. In the natural deduction system, in order to determine whether or not a given formula is provable, we would try to construct a proof of that formula. However, such a proof-search procedure in natural deduction is complicated and not very efficient, since the way to handle assumptions is not well-suited to computer implementation (see [2]). In contrast, resolution provides a more efficient procedure to decide provability of formulas.

**Resolution.** Propositional and first-order resolution were introduced by Robinson "for use as a basic theoretical instrument of the computer theorem-proving program" [16]. Thus, from the outset, it is machine-oriented, rather than is human-oriented as exemplified by natural deduction. Resolution works in essence only for clauses, which consist only of atoms and their negation, with no other logical connectives explicitly occurring. The *logic-free* structure of clauses enables us to formalize a complete system by using only the resolution principle. The efficiency of the rule makes it easy to implement decision procedures involved in establishing the provability of given formulas, and today's most automated theorem-proving programs adopt essentially this principle. The resolution principle also plays an important role in the design of logic programming such as Prolog. (See, e.g. [2].) However, such a refined rule is composed of more primitive rules and is usually explained in terms of rules in natural deduction, and hence its validity and operational meaning are not immediately grasped. We refer to [4,13,2] for explanations of resolution.

The contrast between the two systems is summarized in Table 1:

**Table 1.** The contrast between natural deduction and resolution

| | Natural deduction | Resolution |
|---|---|---|
| Motivation | human-oriented<br>proof construction | machine-oriented<br>refutation procedure |
| Formalization | full syntax of the first order logic<br>rules for each logical connective<br>simplicity of each rule<br>normalization | restricted to clause (logic-free)<br>single rule<br>not primitive but effective rule |
| Implementation | difficult to implement | amenable to implementation |

# 3 Relation-Based Framework and Natural Deduction

## 3.1 Euler Diagrammatic Representation System EUL

We roughly review syntax and semantics of EUL-diagrams of [10,11].

**Definition 1 (EUL-diagrams).** An EUL-*diagram* is a plane ($\mathbb{R}^2$) with a finite number, at least one, of *named simple closed curves* (denoted by $A, B, C, \dots$) and *named points* (denoted by $a, b, c, \dots$), where each named simple closed curve or named point has a unique name. EUL-diagrams are denoted by $\mathcal{D}, \mathcal{E}, \mathcal{D}_1, \mathcal{D}_2, \dots$. An EUL-diagram consisting of at most two objects is called a *minimal diagram*. Minimal diagrams are denoted by $\alpha, \beta, \gamma, \dots$.

In what follows, a named simple closed curve is called a *named circle*. Named circles and named points are collectively called *objects*, and denoted by $s, t, u, \dots$.

**Definition 2.** EUL-*relations* are the following binary relations:

$A \sqsubset B$ "the interior of $A$ is *inside of* the interior of $B$,"
$A \vdash\!\dashv B$ "the interior of $A$ is *outside of* the interior of $B$,"
$A \bowtie B$ "there is at least one *crossing* point between $A$ and $B$,"
$b \sqsubset A$ "$b$ is *inside of* the interior of $A$,"
$b \vdash\!\dashv A$ "$b$ is *outside of* the interior of $A$,"
$a \vdash\!\dashv b$ "$a$ is *outside of* $b$ (i.e. $a$ is not located at the point of $b$)."

EUL-relation $\sqsubset$ is reflexive asymmetric relation, and $\vdash\!\dashv$ and $\bowtie$ are irreflexive symmetric relations.

**Proposition 1.** *Let $\mathcal{D}$ be an* EUL*-diagram. For any distinct objects $s$ and $t$ of $\mathcal{D}$, exactly one of the* EUL*-relations $s \sqsubset t, t \sqsubset s, s \vdash\!\dashv t, s \bowtie t$ holds.*

Observe that, by Proposition 1, the set of EUL-relations holding on a given EUL-diagram $\mathcal{D}$ is uniquely determined. We denote the set by $\mathsf{rel}(\mathcal{D})$. We also denote by $pt(\mathcal{D})$ the set of named points of $\mathcal{D}$, by $cr(\mathcal{D})$ the set of named circles of $\mathcal{D}$, and by $ob(\mathcal{D})$ the set of objects of $\mathcal{D}$. As an illustration, for the diagram $\mathcal{D}_1$ of Fig. 3 below, we have $pt(\mathcal{D}_1) = \{a\}$, $cr(\mathcal{D}_1) = \{A, B, C\}$, and $\mathsf{rel}(\mathcal{D}_1) = \{A \bowtie B, A \bowtie C, B \bowtie C, a \vdash\!\dashv A, a \sqsubset B, a \vdash\!\dashv C\}$. In the description of a set of relations, we usually omit the reflexive relation $s \sqsubset s$ for each object $s$.

**Definition 3 (Equivalence).** When any two objects of the same name appear in different diagrams, we identify them up to isomorphism. Any EUL-diagrams $\mathcal{D}$ and $\mathcal{E}$ such that $ob(\mathcal{D}) = ob(\mathcal{E})$ are *syntactically equivalent* when $\mathsf{rel}(\mathcal{D}) = \mathsf{rel}(\mathcal{E})$.

*Example 1 (Equivalence of diagrams).* For example, diagrams $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$, and $\mathcal{D}_4$ of Fig. 3 below are equivalent since $\mathsf{rel}(\mathcal{D}_1) = \mathsf{rel}(\mathcal{D}_2) = \mathsf{rel}(\mathcal{D}_3) = \mathsf{rel}(\mathcal{D}_4)$.

On the other hand, while $\mathcal{D}_1$ and $\mathcal{D}_5$ (resp. $\mathcal{D}_1$ and $\mathcal{D}_6$) consist of the same objects, they are not equivalent since different EUL-relations hold on them: $A \sqsubset C$ holds on $\mathcal{D}_5$ in place of $A \bowtie C$ of $\mathcal{D}_1$ (resp. $C \sqsubset A$ and $C \sqsubset B$ hold on $\mathcal{D}_6$ in place of $A \bowtie C$ and $C \bowtie B$ of $\mathcal{D}_1$).

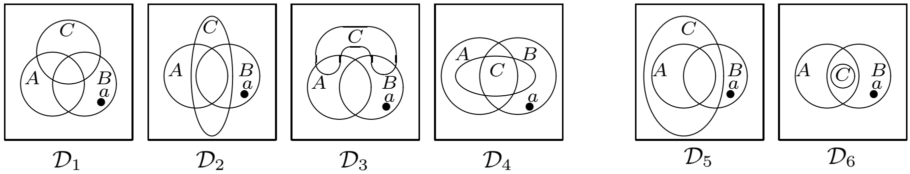



**Fig. 3.** Equivalence of EUL-diagrams

See [10], in which EUL is extended by introducing intersection, union, and complement regions, respectively, as diagrammatic objects, and $\mathcal{D}_1, \mathcal{D}_2$, $\mathcal{D}_3$, and $\mathcal{D}_4$ are distinguished.

In what follows, the diagrams which are syntactically equivalent are identified, and they are referred by a single name. Note that, given the equivalence of EUL-diagrams, there is a one-to-one correspondence between minimal diagrams and EUL-relations. Thus we also denote EUL-relations by $\alpha, \beta, \gamma, \ldots$.

Our semantics is distinct from usual ones, e.g., [7,8] in that diagrams are interpreted in terms of binary relations. In order to interpret the EUL-relations $\sqsubset$ and $\vdash\!\dashv$ uniformly as the subset relation and the disjointness relation, respectively, we regard each point of EUL as a special circle which does not contain, nor cross, any other objects.

**Definition 4 (Model).** A *model* $M$ is a pair $(U, I)$, where $U$ is a non-empty set (the domain of $M$), and $I$ is an interpretation function which assigns to each named circle or point a non-empty subset of $U$ such that $I(a)$ is a singleton for any named point $a$, and $I(a) \neq I(b)$ for any points $a, b$ of distinct names.

**Definition 5 (Truth conditions).** Let $\mathcal{D}$ be an EUL-diagram. $M = (U, I)$ is a *model of* $\mathcal{D}$, written as $M \models \mathcal{D}$, if the following *truth-conditions* (1) and (2) hold: For all objects $s, t$ of $\mathcal{D}$,
(1) $I(s) \subseteq I(t)$ if $s \sqsubset t$ holds on $\mathcal{D}$, and (2) $I(s) \cap I(t) = \emptyset$ if $s \vdash\!\dashv t$ holds on $\mathcal{D}$.

*Remark 1 (Semantic interpretation of $\bowtie$-relation).* By Definition 5, the EUL-relation $\bowtie$ does not contribute to the truth-condition of EUL-diagrams. Informally speaking, $s \bowtie t$ may be understood as $I(s) \cap I(t) = \emptyset$ *or* $I(s) \cap I(t) \neq \emptyset$, which is true in any model.

The semantic consequence relation $\models$ between EUL-diagrams is defined as usual. (See [11] for a detailed description.)

### 3.2 Euler Diagrammatic Inference System GDS as Natural Deduction

In this section, we show that the diagrammatic inference system for EUL-diagrams, called the Generalized Diagrammatic Syllogistic inference system GDS of [11,10], corresponds to the natural deduction system for disjunction-free minimal logic. We first give a translation of each EUL-diagram to an implicational formula.

We consider the propositional fragment of natural deduction. We denote atoms (propositional variables) by $A, B, C, \dots$. Formulas are defined inductively as usual by using connectives $\wedge, \vee, \rightarrow, \neg, \bot$, and formulas are denoted by $\varphi, \psi, \theta, \dots$. In what follows, we consider the $\wedge$ connective as an $n$-ary connective for an appropriate $n$. Furthermore, we denote by a sequence (set) $\varphi_1, \dots, \varphi_n$ a conjunction $\varphi_1 \wedge \cdots \wedge \varphi_n$, where we assume all conjuncts are distinct. We also generalize $\wedge I$ and $\wedge E$ rules of natural deduction to those for the $n$-ary $\wedge$ connective.

**Definition 6 (Translation of EUL-diagrams).** Each named circle or named point is translated into an atom. Then each EUL-relation $\alpha$ is translated into an implicational formula $\alpha^\circ$ as follows:

$$(s \sqsubset t)^\circ := s \rightarrow t \qquad (s \vdash\!\dashv t)^\circ := s \rightarrow \neg t \qquad (s \bowtie t)^\circ := s \rightarrow s, t \rightarrow t$$

Let $\mathcal{D}$ be an EUL-diagram whose set of relations $\mathsf{rel}(\mathcal{D})$ is $\{\alpha_1, \dots, \alpha_n\}$. The diagram $\mathcal{D}$ is translated into the conjunction $\mathcal{D}^\circ := \alpha_1^\circ, \dots, \alpha_n^\circ$.

We next give a translation of inference rules of GDS [11,10]. GDS consists of two inference rules: *unification* and *deletion*. Two kinds of constraint are imposed on unification. One is the *constraint for determinacy*, which blocks the disjunctive ambiguity with respect to locations of named points. For example, two diagrams $\mathcal{D}_1$ and $\mathcal{D}_2$ in Fig. 4 are not permitted to be unified into one diagram since the location of the point $c$ is not determined.

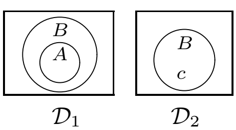



**Fig. 4.** Indeterminacy

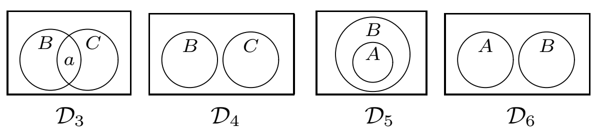



**Fig. 5.** Inconsistency

The other is the *constraint for consistency*, which blocks representing inconsistent information in a single diagram. For example, the diagrams $\mathcal{D}_3$ and $\mathcal{D}_4$ (resp. $\mathcal{D}_5$ and $\mathcal{D}_6$) in Fig. 5 are not permitted to be unified since they contradict each other based on the semantics of EUL. Unification rules of two diagrams are formalized by restricting one of the two diagrams to being a *minimal diagram*, except for one rule called the Point Insertion-rule. The restriction of unification makes the operational meaning of it clear. Our completeness ([11]) ensures that any diagrams $\mathcal{D}_1, \dots, \mathcal{D}_n$ may be unified, under the constraints for determinacy and consistency, into one diagram whose semantic information is equivalent to the conjunction of those of $\mathcal{D}_1, \dots, \mathcal{D}_n$. Inference rules are described in terms of EUL-relations: Given a diagram $\mathcal{D}$ and a minimal diagram $\alpha$, the set of relations $\mathsf{rel}(\mathcal{D} + \alpha)$ for the unified diagram $\mathcal{D} + \alpha$ is defined.

We give the definition of each inference rule of GDS, and in parallel with it, we give a translation of each rule into a combination of inference rules of natural deduction. Unification between $\mathcal{D}$ and $\alpha$ such that $\mathsf{rel}(\mathcal{D}+\alpha) = \mathsf{rel}(\mathcal{D}) \cup \{\gamma_1, \dots, \gamma_n\}$ is translated schematically as in Fig. 6 below.

$$\dfrac{\mathcal{D}^\circ \qquad \begin{array}{c}\mathcal{D}^\circ\ \alpha^\circ\\ \vdots\\ \gamma_1^\circ\end{array} \qquad \cdots \qquad \begin{array}{c}\mathcal{D}^\circ\ \alpha^\circ\\ \vdots\\ \gamma_n^\circ\end{array}}{\mathcal{D}^\circ, \gamma_1^\circ, \ldots, \gamma_n^\circ}\ \wedge I$$

**Fig. 6**

$$\left(\begin{array}{c}\vdots\\ \varphi_n\end{array}\right)_n$$

**Fig. 7**

In the following natural deduction proofs, by $(\varphi_n)_n$ we mean the set of formulas $\varphi_1, \ldots, \varphi_n$. Furthermore, for each formula $\varphi_n$, the repetition of the same inference steps is expressed as the skeleton of a proof as in Fig. 7:

**Definition 7 (Translation of GDS).** Inference rules of GDS are translated as follows. (Because of space limitations, we show here only U7 and U9 rules. See [12] for the remaining rules.)

U7: If $A \mathrel{\vdash\!\!\dashv} B$ holds on $\alpha$ and $A \in cr(\mathcal{D})$, and if $c \sqsubset A$ holds for all $c \in pt(\mathcal{D})$, then $\mathsf{rel}(\mathcal{D} + \alpha)$ is defined as follows:

$$\begin{aligned}&\mathsf{rel}(\mathcal{D}) \cup \{C \mathrel{\vdash\!\!\dashv} B \mid C \sqsubset A \in \mathsf{rel}(\mathcal{D})\} \cup \{c \mathrel{\vdash\!\!\dashv} B \mid c \in pt(\mathcal{D})\}\\ &\quad \cup \{C \bowtie B \mid A \sqsubset C \text{ or } A \mathrel{\vdash\!\!\dashv} C \text{ or } A \bowtie C \in \mathsf{rel}(\mathcal{D})\} \cup \{B \sqsubset B\}\end{aligned}$$

U7 is translated as follows:

$$\dfrac{\mathcal{D}^\circ \quad \left(\dfrac{\dfrac{\dfrac{[C_n]^1 \quad \dfrac{\mathcal{D}^\circ}{C_n \to A}}{A} \quad A \to \neg B}{\neg B}}{C_n \to \neg B}\,1\right)_n \quad \left(\dfrac{\dfrac{\dfrac{[c_m]^1 \quad \dfrac{\mathcal{D}^\circ}{c_m \to A}}{A} \quad A \to \neg B}{\neg B}}{c_m \to \neg B}\,1\right)_m \quad \dfrac{[B]^1}{B \to B}\,1}{\mathcal{D}^\circ, (C_n \to \neg B)_n, (c_m \to \neg B)_m, B \to B}$$

U9: If $A \sqsubset B$ holds on $\alpha$ and $A \bowtie B$ holds on $\mathcal{D}$, and if there is no object $s$ such that $s \sqsubset A$ and $s \mathrel{\vdash\!\!\dashv} B$ hold on $\mathcal{D}$, then $\mathsf{rel}(\mathcal{D} + \alpha)$ is defined as follows:

$$\begin{aligned}&\big(\mathsf{rel}(\mathcal{D}) \setminus \{D \bowtie C \mid D \sqsubset A \text{ and } B \sqsubset C \in \mathsf{rel}(\mathcal{D})\} \setminus \{C \bowtie D \mid C \sqsubset A \text{ and } D \mathrel{\vdash\!\!\dashv} B \in \mathsf{rel}(\mathcal{D})\}\big)\\ &\quad \cup \{D \sqsubset C \mid D \sqsubset A \text{ and } B \sqsubset C \in \mathsf{rel}(\mathcal{D})\} \cup \{C \mathrel{\vdash\!\!\dashv} D \mid C \sqsubset A \text{ and } D \mathrel{\vdash\!\!\dashv} B \in \mathsf{rel}(\mathcal{D})\}\end{aligned}$$

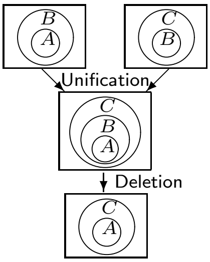
U9 is translated as follows:

$$\dfrac{\mathcal{D}^\circ \quad \left(\dfrac{\dfrac{\dfrac{\dfrac{[D_n]^1 \quad \dfrac{\mathcal{D}^\circ}{D_n \to A}}{A} \quad A \to B}{B} \quad \dfrac{\mathcal{D}^\circ}{B \to C_m}}{C_m}}{D_n \to C_m}\,1\right)_{n,m} \quad \left(\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[C_k]^2 \quad \dfrac{\mathcal{D}^\circ}{C_k \to A}}{A} \quad A \to B}{B} \quad \dfrac{[D_l]^1 \quad \dfrac{\mathcal{D}^\circ}{D_l \to \neg B}}{\neg B}}{\bot}}{\neg D_l}\,1}{C_k \to \neg D_l}\,2\right)_{k,l}}{\mathcal{D}^\circ, (D_n \to C_m)_{n,m}, (C_k \to \neg D_l)_{k,l}}$$

Definition 7 gives, by induction, a translation of any *diagrammatic proof* $\pi$ of GDS into a natural deduction proof $\pi^\circ$. Hence the following theorem is immediate:

**Theorem 1 (Translation of GDS).** *Let $\mathcal{D}_1, \ldots, \mathcal{D}_n, \mathcal{E}$ be* EUL*-diagrams. If $\pi$ is a diagrammatic proof of $\mathcal{E}$ from $\mathcal{D}_1, \ldots, \mathcal{D}_n$ in* GDS*, then $\pi^\circ$ is a natural deduction proof of $\mathcal{E}^\circ$ from $\mathcal{D}_1^\circ, \ldots, \mathcal{D}_n^\circ$.*

*Example 2 (Barbara in* GDS*).* The following diagrammatic proof on the left, which expresses the famous valid syllogism called *Barbara*, is translated by Definition 7 into the natural deduction proof on the right. For the sake of simplicity, we omit tautologies of the form $A \to A$ in the proof:




$$\dfrac{\dfrac{A \to B \quad B \to C \quad \dfrac{\dfrac{\dfrac{[A]^1 \quad A \to B}{B}{\scriptstyle \to E} \quad B \to C}{C}{\scriptstyle \to E}}{A \to C}{\scriptstyle \to I, 1}}{A \to B, B \to C, A \to C}{\scriptstyle \land I}}{A \to C}{\scriptstyle \land E}$$

Observe that for the simulation of GDS, we use only a particular class of inference rules: $\land I, \land E, \to I, \to E, \neg I, \neg E$, which form a subsystem of classical logic, i.e., minimal logic without disjunction. We denote the system as NM. Further note that the natural deduction proof in Example 2 above is not in normal form since it contains redundant steps: without applying $\land I$ and $\land E$ rules, we already have a proof of $A \to C$. **Normalization theorem** for NM states that *any proof of $\varphi$ in* NM *reduces to a normal proof of $\varphi$.* By reducing the above proof, we obtain the following normal proof in Fig. 8:

$$\dfrac{\dfrac{\dfrac{[A]^1 \quad A \to B}{B}{\scriptstyle \to E} \quad B \to C}{C}{\scriptstyle \to E}}{A \to C}{\scriptstyle \to I, 1}$$

**Fig. 8.** Normal proof for Barbara

$$\dfrac{A \to B \quad B \to C}{A \to C}\sqsubset \qquad \dfrac{A \to B \quad B \to \neg C}{A \to \neg C}\vDash$$

**Fig. 9.** Derived rules

The above normal proof provides a derived rule called $\sqsubset$-rule in Fig. 9, and this rule makes explicit the correspondence between the natural deduction proof with $\sqsubset$-rule and the diagrammatic proof for Barbara.

**Lemma 1 ($\sqsubset$ and $\vDash$ rules).** *$\sqsubset$-rule and $\vDash$-rule of Fig. 9 are derived rules in disjunction-free minimal logic.*

**Proposition 2 (Translation of NM).** *Let $\boldsymbol{\mathcal{D}}$ be a set of* EUL*-diagrams which has a model. Let $\alpha$ be a minimal diagram. Any proof of $\alpha^\circ$ from $\boldsymbol{\mathcal{D}}^\circ$ in disjunction-free minimal logic is transformed into a diagrammatic proof of $\alpha$ from $\boldsymbol{\mathcal{D}}$ in* GDS.

*Proof (sketch).* We assume, for simplicity of argument, that $\boldsymbol{\mathcal{D}}$ is a set of minimal diagrams $\boldsymbol{\beta}$. By the normalization theorem for minimal logic, any proof is transformed into a normal proof. Let $\pi$ be such a normal proof of $\alpha^\circ$ from $\boldsymbol{\beta}^\circ$. By the subformula property of normal proofs, $\pi$ consists only of subformulas of $\alpha^\circ$ and $\boldsymbol{\beta}^\circ$, and it is shown that $\pi$ has the following two forms depending on $\alpha^\circ$:

(1) When $\alpha^\circ \equiv s \to t$, $\pi$ has the following form on the left, where $\vdots$ means successive applications of $\to E$-rule:

$$\dfrac{\dfrac{\begin{array}{c}\dfrac{\dfrac{[s]^1 \quad s \to s_1}{s_1}{\to}E \quad s_1 \to s_2}{s_2}{\to}E \\ \vdots \\ s_n\end{array} \quad s_n \to t}{t}{\to}E}{s \to t}{\to}I,1 \qquad\qquad \dfrac{\begin{array}{c}\dfrac{s\to s_1 \quad s_1\to s_2}{s\to s_2}\\ \vdots \\ s\to s_n\end{array} \quad s_n\to t}{s\to t}$$

Then $\pi$ is transformed into the above $\pi'$ on the right by using $\sqsubset$-rule.

(2) When $\alpha^\circ \equiv s \to \neg t$, $\pi$ has the following form on the left, where $\vdots$ means successive applications of $\to E$-rule, and each $\varphi_j$ is an atom or its negation:

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\dfrac{\dfrac{[s]^2 \quad s\to s_1}{s_1}{\to}E \quad s_1\to s_2}{s_2}{\to}E\\ \vdots\\ u\end{array} \quad \begin{array}{c}\dfrac{\dfrac{[t]^1 \quad t\to\varphi_1}{\varphi_1}{\to}E \quad \varphi_1\to\varphi_2}{\varphi_2}{\to}E\\ \vdots \\ \neg u\end{array}}{\bot}\neg E}{\neg t}\neg I,1}{s\to\neg t}{\to}I,2 \qquad \dfrac{\begin{array}{c}\dfrac{s\to s_1\quad s_1\to s_2}{s\to s_2}\\ \vdots\\ s\to u\end{array}\quad \begin{array}{c}\dfrac{t\to\varphi_1\quad \varphi_1\to\varphi_2}{t\to\varphi_2}\\ \vdots\\ t\to\neg u\end{array}}{s\to\neg t}$$

Then $\pi$ is transformed into the above $\pi'$ on the right by using $\sqsubset$- and $\vdash\!\dashv$-rules. In either case, $\pi'$ is easily transformed into a diagrammatic proof of $\alpha$ from $\boldsymbol{\beta}$. ■

Note that when premises and the conclusion are restricted to being minimal diagrams in Proposition 2, a diagrammatic proof obtained from the above translation has the following particular form, cf. Example 2:

**Definition 8 (±-normal form).** A diagrammatic proof $\pi$ is in ±-*normal form* if a unification (+) and a deletion (−) appear alternately in $\pi$.

Thus Proposition 2 gives a normalization theorem of GDS:

**Corollary 1 (±-normal form).** *Let $\boldsymbol{\beta}$ be a set of minimal diagrams which has a model. Let $\alpha$ be a minimal diagram. If $\alpha$ is provable from $\boldsymbol{\beta}$ in* GDS*, then there is a ±-normal diagrammatic proof of $\alpha$ from $\boldsymbol{\beta}$.*

See [11] for a semantic proof of the theorem. This ±-normal form is important because it serves to show the correspondence between diagrammatic proofs and chains of Aristotelian categorical syllogisms. (Cf. [9].)

By applying the construction of canonical diagrammatic proofs given in [11], Proposition 2 is naturally extended to the general case, where the conclusion is not restricted to be minimal:

**Theorem 2 (Translation of NM).** *Let $\boldsymbol{\mathcal{D}}$ be a set of* EUL*-diagrams which has a model. Let $\mathcal{E}$ be an* EUL*-diagram. Any proof of $\mathcal{E}^\circ$ from $\boldsymbol{\mathcal{D}}^\circ$ in disjunction-free minimal logic is transformed into a diagrammatic proof of $\mathcal{E}$ from $\boldsymbol{\mathcal{D}}$ in* GDS.

# 4 Region-Based Framework and Resolution Calculus

## 4.1 Venn Diagrammatic Representation System

We briefly recall syntax and semantics of Venn diagrams. See, e.g., [8,18] for detailed descriptions. We define Venn diagrams in terms of shaded regions. Euler diagrams in the region-based framework, called *Euler diagrams with shading*, are obtained by considering some shaded regions of Venn diagrams as "missing" regions.

A concrete Venn diagram consists of finite numbers of named circles (simple closed curves) on a plane enclosed by a boundary rectangle which satisfies the *partial-overlapping* condition, i.e., all possible intersections of circles must occur. A zone or minimal region is a connected component of the complement of the contour set, which may be shaded. Independently of a concrete plane diagram, an abstract Venn diagram is defined in terms of names of circles as follows: Let $\mathcal{L}$ be the finite set of the names of circles for a diagram $\mathcal{V}$. An **abstract zone** (or **minimal region**) $z$ is defined as $z = (in(z), out(z))$, where $in(z)$ and $out(z)$ are finite subsets of $\mathcal{L}$ such that $in(z) \cap out(z) = \emptyset$ and $in(z) \cup out(z) = \mathcal{L}$. Let $zone(\mathcal{V})$ be the set of zones such that $\{in(z) \mid z \in zone(\mathcal{V})\}$ is the power set $\mathcal{P}(\mathcal{L})$ of $\mathcal{L}$. Then an **abstract Venn diagram** is defined as the set $shade(\mathcal{V})$ of shaded zones which is a subset of $zone(\mathcal{V})$. When $shade(\mathcal{V}) = \emptyset$, $\mathcal{V}$ is called a *primary diagram*. We denote Venn diagrams by $\mathcal{V}, \mathcal{W}, \ldots$.

A **model** of Venn diagrams is a pair $M = (U, I)$ where $U$ is a non-empty set called the universe, and $I$ is an interpretation function which assigns to each circle a subset of $U$. The interpretation function $I$ is naturally extended to interpret zones as follows: For any zone $z = (in(z), out(z))$ the interpretation $I(z)$ of a zone is defined by $I(z) = \bigcap_{X \in in(z)} I(X) \ \cap \ \bigcap_{Y \in out(z)} \overline{I(Y)}$, where $\overline{I(Y)}$ is the complement of $I(Y)$. $M = (U, I)$ is a model of a Venn diagram $\mathcal{V}$, denoted as $M \models \mathcal{V}$, if each shaded zone is interpreted as the empty set, i.e., $\bigcup_{z \in shade(\mathcal{V})} I(z) = \emptyset$. Cf. [8,18].

In a similar way as EUL-diagrams, points (with linking) in Venn diagrams can be considered as special circles. For example, the Venn diagram on the left in Fig. 10, which contains two occurrences of a point $c$ and a linking of them, can be replaced by the diagram on the right:


**Fig. 10.** Points as circles in Venn diagrams

Note that each point expresses the existence of an object, and hence, when a point as a circle is completely shaded, it expresses the contradiction.

### 4.2 Venn Diagrammatic Inference System as Resolution Calculus

In this section, we show that a Venn diagrammatic inference system corresponds to the resolution calculus over clauses. In the context of resolution, we use the "overbar" symbol for negation instead of the usual $\neg$. We denote literals, i.e., either atoms or their negation, by $L, M, N, \ldots$. If $L$ is a literal of the form $\overline{A}$, then $\overline{L}$ denotes the unnegated literal $A$. A **clause** is a finite set of literals, and it is denoted by $x, y, z, \ldots$. When a clause $x$ is $\{L_1, \ldots, L_n\}$, it is sometimes denoted as $L_1 \cdots L_n$. In particular, the empty-clause is denoted by $\Box$. A non-empty set of clauses is called a **clause set**, and it is denoted by $\Gamma, \Delta, \Sigma, \ldots$. We first describe a translation of Venn diagrams into sets of clauses.

**Definition 9 (Translation of Venn diagrams).** Let $\mathcal{V}$ be a Venn diagram whose set of shaded zones $shade(\mathcal{V})$ is $\{z_1, \ldots, z_n\}$.

- Each shaded zone $z_i = (\{A_1, \ldots, A_j\}, \{B_1, \ldots, B_k\})$ is interpreted as a clause $z_i^* = \{A_1, \ldots, A_j, \overline{B_1}, \cdots, \overline{B_k}\}$, which is abbreviated as $A_1 \cdots A_j \overline{B_1} \cdots \overline{B_k}$.
- The Venn diagram $\mathcal{V}$ is translated into the set of clauses $\mathcal{V}^* = \{z_1^*, \ldots, z_n^*\}$. In particular, the completely shaded diagram without any circle is translated as $\{\Box\}$, and each primary diagram is translated as $\emptyset$.

In what follows, we denote by $\boldsymbol{L}$ a set of literals, i.e., a clause, $L_1 \cdots L_n$ for an appropriate $n$. For a clause $z$, we denote by $|z|$ the set of atoms which appear in $z$. For a given clause set $\Gamma$, let $|\Gamma| = \bigcup\{|z| \mid z \in \Gamma\}$.

For example, the Venn diagram on the right in Fig. 10 is translated into a set of clauses $\{ABc, \overline{A}\overline{B}c\}$. For the interpretation of Venn diagrams, a clause, i.e., a zone of a Venn diagram, say $z = \overline{A}\overline{B}C$, is interpreted as the conjunctive formula $\wedge z = \overline{A} \wedge \overline{B} \wedge C$; then a clause set, i.e., a Venn diagram, say $\mathcal{V} = \{ABC, \overline{A}\overline{B}C\}$, is interpreted as the disjunctive normal form (DNF) formula $d(\mathcal{V}) = (A \wedge B \wedge C) \vee (\overline{A} \wedge \overline{B} \wedge C)$. Cf. the semantics of Venn diagrams.

Based on the above translation of Venn diagrams into sets of clauses, inference rules for Venn diagrams are described as the following inference rules over clauses. We call the system VR. In particular, any primary diagram is an axiom. See, e.g. [8,18] for formal descriptions of Venn diagrammatic inference rules.

**Definition 10 (Translation of VR).** Inference rules for Venn diagrams are translated as follows:

Introduction of a circle: A new circle $A$ may be added to a diagram observing the partial-overlapping rule, i.e., each zone splits into two zones with the introduction of $A$. If the zone is shaded, then both corresponding new zones are shaded.

Let $\mathcal{V}$ be a Venn diagram such that $\mathcal{V}^* = \{\boldsymbol{L}_1, \ldots, \boldsymbol{L}_n\}$ and $A \notin |\boldsymbol{L}_i|$:

$$\frac{\{\boldsymbol{L}_1, \ldots, \boldsymbol{L}_n\}}{\{A\boldsymbol{L}_1, \ldots, A\boldsymbol{L}_n,\ \overline{A}\boldsymbol{L}_1, \ldots, \overline{A}\boldsymbol{L}_n\}}\ Intro, A$$

Superposition of diagrams (Combining diagrams): Two diagrams that have the same circles may be combined into a diagram whose semantic information is equivalent to the conjunction of those of the original diagrams. Shaded zones in the combined diagram are shaded in one (or both) of the original diagrams.

Let $\mathcal{V}_1^*$ be $\{\boldsymbol{L}_1, \ldots, \boldsymbol{L}_n\}$ and $\mathcal{V}_2^*$ be $\{\boldsymbol{M}_1, \ldots, \boldsymbol{M}_m\}$ such that $|\mathcal{V}_1^*| = |\mathcal{V}_2^*|$:

$$\frac{\{\boldsymbol{L}_1, \ldots, \boldsymbol{L}_n\} \quad \{\boldsymbol{M}_1, \ldots, \boldsymbol{M}_m\}}{\{\boldsymbol{L}_1, \ldots, \boldsymbol{L}_n\} \cup \{\boldsymbol{M}_1, \ldots, \boldsymbol{M}_m\}} \; Sup$$

Erasure of shaded zones: Any shaded zones may be erased from a diagram.

Let $\mathcal{V}$ be a Venn diagram such that $\mathcal{V}^* = \{\boldsymbol{L}_1, \ldots, \boldsymbol{L}_n, \boldsymbol{L}_{n+1}, \ldots, \boldsymbol{L}_m\}$:

$$\frac{\{\boldsymbol{L}_1, \ldots, \boldsymbol{L}_n, \boldsymbol{L}_{n+1}, \ldots, \boldsymbol{L}_m\}}{\{\boldsymbol{L}_1, \ldots, \boldsymbol{L}_n\}} \; ErS$$

Erasure of a circle: A circle $A$ may be erased from a diagram so that any shading remaining in only a part of a zone should also be erased.

Let $\mathcal{V}$ be a Venn diagram such that $\mathcal{V}^* = \{A\boldsymbol{L}_1, \ldots, A\boldsymbol{L}_n,\ \overline{A}\boldsymbol{M}_1, \ldots, \overline{A}\boldsymbol{M}_m\}$:

$$\frac{\{A\boldsymbol{L}_1, \ldots, A\boldsymbol{L}_n,\ \overline{A}\boldsymbol{M}_1, \ldots, \overline{A}\boldsymbol{M}_m\}}{\{\boldsymbol{L}_i\boldsymbol{M}_j \mid \neg\exists L(L \in \boldsymbol{L}_i \textit{ and } \overline{L} \in \boldsymbol{M}_j)\}} \; Er, A$$

We show that VR is simulated by the resolution calculus. In the literature on automated theorem proving, many refined strategies to construct resolution derivations have been studied (see Chang-Lee [4]). Among them, we mention a naive strategy, called the Davis-Putnam procedure: For any ordering of atoms $A_1, A_2, \ldots, A_m$, a derivation is required to have a sequence of nested resolutions; i.e. first resolutions with respect to $A_1$, then resolutions with respect to $A_2$, etc., concluding with resolutions with respect to $A_m$. With this stipulation, we slightly extend the notion of derivability in our resolution calculus: $\Delta$ is derivable from $\Gamma$ when, for any $x \in \Delta$, $x$ is derivable from $\Gamma$.

For any clause sets $\Gamma$ and $\Delta$, we denote $\Gamma \simeq \Delta$ when they are semantically equivalent, that is, $I(d(\Gamma)) = I(d(\Delta))$ for any model $M = (U, I)$.

**Theorem 3 (Translation of VR).** *Let $\mathcal{V}_1, \ldots, \mathcal{V}_n, \mathcal{V}$ be Venn diagrams. If $\mathcal{V}$ is derivable from $\mathcal{V}_1, \ldots, \mathcal{V}_n$ in* VR*, then there is a Venn diagram $\mathcal{W}$ such that $\mathcal{V}_1^* \cup \cdots \cup \mathcal{V}_n^* \simeq \mathcal{W}^*$ and $\mathcal{V}^*$ is derivable from $\mathcal{W}^*$ in the resolution calculus.*

*Proof.* Note that there is a derivation of $\mathcal{V}$ such that applications of $Er(S)$ rule are delayed, in the sense that successive applications of $Intro$ and $Sup$ rules are followed by successive applications of $Er(S)$ rule. Thus, there is a Venn diagram $\mathcal{W}$ which consists of all circles contained in $\mathcal{V}_1, \ldots, \mathcal{V}_n$ (cf. Shin's maximal diagram [18]). Observe that $Intro, A$ rule is described by the following semantic equations:

$$\boldsymbol{L}_1 \vee \cdots \vee \boldsymbol{L}_n = (\boldsymbol{L}_1 \vee \cdots \vee \boldsymbol{L}_n) \wedge (A \vee \overline{A}) = A\boldsymbol{L}_1 \vee \cdots \vee A\boldsymbol{L}_n \vee \overline{A}\boldsymbol{L}_1 \vee \cdots \vee \overline{A}\boldsymbol{L}_n$$

This means that when $\mathcal{W}_2$ is obtained from $\mathcal{W}_1$ by $Intro$ rule, $\mathcal{W}_1 \simeq \mathcal{W}_2$. Thus, since $Sup$ rule is simply the union operation, we have $\mathcal{V}_1^* \cup \cdots \cup \mathcal{V}_n^* \simeq \mathcal{W}^*$.

Furthermore, note that applications of $ErS$ rule are further delayed, i.e., each $ErS$ rule is applied in the last part of the derivation. We assume without loss of generality that no primary diagram is derived by $Er(S)$ rule, since it is an axiom. Then $Er, A$ rule corresponds to the following applications of resolution with respect to $A$:

$$\frac{A\boldsymbol{L}_i \quad \overline{A}\boldsymbol{M}_j}{\boldsymbol{L}_i\boldsymbol{M}_j}\ A$$

Thus, for any $z^* \in \mathcal{V}^*$, $z^*$ is derivable from $\mathcal{W}^*$ in the resolution calculus. ■

The above proof shows that VR is a resolution calculus over *full* DNF, i.e., a DNF such that each of its atoms appears exactly once in all conjuncts.

The converse of Theorem 4 also holds.

**Theorem 4 (Translation of resolution).** *Let $\mathcal{V}_1, \ldots, \mathcal{V}_n, \mathcal{V}$ be Venn diagrams. If $\mathcal{V}^*$ is derivable from $\mathcal{V}_1^* \cup \cdots \cup \mathcal{V}_n^*$ in the resolution calculus, then $\mathcal{V}$ is derivable from $\mathcal{V}_1, \ldots, \mathcal{V}_n$ in the Venn diagrammatic inference system* VR.

*Proof.* Let us consider the following resolution with respect to $A$ on the left:

$$\frac{AL \quad \overline{A}M}{LM}\ A \qquad\qquad \cfrac{\cfrac{\cfrac{\{AL\}}{\{ALM, AL\overline{M}\}}\ \mathit{Intro}, M \qquad \cfrac{\{\overline{A}M\}}{\{\overline{A}ML, \overline{A}M\overline{L}\}}\ \mathit{Intro}, L}{\{ALM, AL\overline{M}, \overline{A}ML, \overline{A}M\overline{L}\}}\ \mathit{Sup}}{\{LM\}}\ \mathit{Er}, A$$

It is shown by induction (the base case is obtained by using $ErS$ rule) that there are Venn diagrams which correspond to $AL$ and $\overline{A}M$, respectively. Then the resolution is simulated by the above derivation on the right in VR. The same applies to the general case in which $L$ and $M$ are extended to sequences of literals $\boldsymbol{L}$ and $\boldsymbol{M}$. Thus there is a derivation of each zones $z \in \mathcal{V}$ in VR. It is obvious that they are superposed into the Venn diagram $\mathcal{V}$. ■

## 5 Discussion and Future Work

We showed that the Euler diagrammatic inference system GDS, which is formalized in the relation-based framework, corresponds to the natural deduction system for disjunction-free minimal logic (Theorems 1 and 2), and that the Venn diagrammatic inference system VR, which is formalized in the region-based framework, corresponds to the resolution calculus (Theorems 3 and 4). These correspondences highlight both advantages and disadvantages of the two frameworks. On the one hand, inference rules in the relation-based framework provide a natural notion of normal diagrammatic proofs, which neatly characterizes the notion of linguistic chains of syllogisms; but the drawback is that they may cause some complications in their implementation. On the other hand, inference rules in the region-based framework are easy to control so that they are utilized to automated theorem proving; but such resolution-style rules do not provide an appropriate notion of diagrammatic proofs. (See also [17] for our cognitive psychological experiments, which compare Euler and Venn diagrams in actual human reasoning assignments.)

Furthermore, with these correspondences, there arises the possibility of applying well-developed techniques in traditional proof theory within symbolic logic in the field of diagrammatic reasoning. Firstly, it is well-known that the Curry-Howard correspondence between natural deduction and typed $\lambda$-calculus provides correspondences between formulas and types, between proofs and $\lambda$-terms, and between reduction steps of proofs and $\lambda$-terms (see [2]). Under the correspondence between natural deduction and GDS, then, we can explore computational aspects of Euler diagrammatic proofs by defining their syntactic rewriting procedure in GDS. Secondly, in studies on resolution for theorem proving within symbolic logic, many effective strategies are developed (see, e.g. [2,4]). We may apply such strategies for the study of theorem proving using Venn diagrams. Thirdly, complexity of proofs in resolution calculus has been very well studied, e.g. [3]. Such results may be applied to the complexity analysis on Venn diagrammatic proofs.

## References


1. Braine, M.D.S., O'Brien, D.P.: Mental logic. Lawrence Erlbaum, Mahwah (1998)
2. Buss, S.R.: An Introduction to Proof Theory. In: Buss, S.R. (ed.) Handbook Proof Theory. Elsevier, Amsterdam (1998)
3. Buss, S.R.: Propositional Proof Complexity: An Introduction. In: Berger, U., Schwichtenberg, H. (eds.) Computational Logic, pp. 127–178. Springer, Berlin (1999)
4. Chang, C.-L., Lee, R.C.-T.: Symbolic Logic and Mechanical Theorem Proving. Academic Press, New York (1973)
5. Flower, J., Masthoff, J., Stapleton, G.: Generating Readable Proofs: A Heuristic Approach to Theorem Proving With Spider Diagrams. In: Blackwell, A.F., Marriott, K., Shimojima, A. (eds.) Diagrams 2004. LNCS (LNAI), vol. 2980, pp. 166–181. Springer, Heidelberg (2004)
6. Gentzen, G.: Unter suchungen uber das logische Scliessen. Mathematische Zetischrift 39, 176–210, 405–431 (1935); English Translation: Investigations into logical deduction. In: Szabo, M.E. (ed.) The collected Papers of Gerhard Gentzen (1969)
7. Hammer, E., Shin, S.-J.: Euler's visual logic. History and Philosophy of Logic 19, 1–29 (1998)
8. Howse, J., Stapleton, G., Taylor, J.: Spider Diagrams. LMS Journal of Computation and Mathematics 8, 145–194 (2005)
9. Mineshima, K., Okada, M., Sato, Y., Takemura, R.: Diagrammatic Reasoning System with Euler Circles: Theory and Experiment Design. In: Stapleton, G., Howse, J., Lee, J. (eds.) Diagrams 2008. LNCS (LNAI), vol. 5223, pp. 188–205. Springer, Heidelberg (2008)
10. Mineshima, K., Okada, M., Takemura, R.: Conservativity for a hierarchy of Euler and Venn reasoning systems. In: Proceedings of Visual Languages and Logic 2009. CEUR Series, vol. 510, pp. 37–61 (2009)
11. Mineshima, K., Okada, M., Takemura, R.: A Diagrammatic Inference System with Euler Circles (2009) (submitted to a journal)
12. Mineshima, K., Okada, M., Takemura, R.: Two Types of Diagrammatic Inference Systems: Natural Deduction Style and Resolution Style, an extended version of this paper, `http://abelard.flet.keio.ac.jp/person/takemura/index.html`

13. Ono, H.: Logic in Information Science (Joho-kagaku ni okeru Ronri; in Japanese), Nippon-Hyoron-sha (1994)
14. Prawitz, D.: Natural Deduction. Almqvist & Wiksell (1965); Dover (2006)
15. Rips, L.: The psychology of proof. MIT Press, Cambridge (1994)
16. Robinson, J.A.: A Machine-Oriented Logic Based on the Resolution Principle. Journal of the ACM 12(1), 23–41 (1965)
17. Sato, Y., Mineshima, K., Takemura, R.: The efficacy of Euler and Venn diagrams in deductive reasoning: empirical findings. In: Goel, A.K., Jamnik, M., Narayanan, N.H. (eds.) Diagrams 2010. LNCS (LNAI), vol. 6170, pp. 6–22. Springer, Heidelberg (2010)
18. Shin, S.-J.: The Logical Status of Diagrams. Cambridge University Press, Cambridge (1994)
19. Stapleton, G.: A survey of reasoning systems based on Euler diagrams. Proceedings of Euler 2004, Electronic Notes in Theoretical Computer Science 134(1), 127–151 (2005)
20. Stapleton, G., Masthoff, J., Flower, J., Fish, A., Southern, J.: Automated Theorem Proving in Euler Diagram Systems. Journal of Automated Reasoning 39(4), 431–470 (2007)

# Alternative Strategies for Spatial Reasoning with Diagrams

Mike Stieff[1], Mary Hegarty[3], and Bonnie Dixon[2]

[1] Department of Curriculum and Instruction
[2] Department of Chemistry and Biochemistry
University of Maryland
College Park, MD 20742, USA
mstieff@umd.edu, bdixon1@umd.edu
[3] Department of Psychology,
University of California
Santa Barbara, CA 93101, USA
hegarty@psych.ucsb.edu

**Abstract.** This paper examines the nature of problem solving with diagrams on domain-general and domain-specific spatial tasks. Although individuals often solve such tasks using imagistic strategies (e.g. mental rotation, perspective taking), alternative strategies are available. For example, the use of algorithms or heuristics can allow the problem solver to complete these tasks by abstracting spatial or non-spatial information from internal or external representations. Here, we explore the availability of diverse strategies for solving tasks from spatial ability tests and organic chemistry. From an analysis of the potential problem spaces on such tasks, we offer a novel framework for classifying the strategies individuals use to solve spatial problems. Our classification places spatial problem solving strategies in a space defined by three dimensions that characterize the extent to which a strategy (1) recruits spatial versus non-spatial information, (2) relies on internal versus external representations and (3) involves modification of representations.

**Keywords:** mental imagery, analytic strategies, diagrams, internal and external representations.

## 1 Introduction

Problem solving with diagrams in science domains mandates spatial reasoning, that is, inference from spatial information. In most domains, problem solvers must consider the transformation of objects in three-dimensional space. For example, physicists reason about flying balls and spinning gyroscopes, and chemists justify the relationship between molecular structure and chemical reactivity. Because many of the objects, phenomena and concepts under study in science are inaccessible to the human eye, problem solvers frequently report using strategies that include generating, transforming and inspecting mental images using spatial information embedded in domain-specific diagrams. The ubiquity of this approach for reasoning about diagrams in

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 115–127, 2010.

science suggests that one's spatial ability can, to some degree, predict success or failure in learning science and science careers. Indeed, psychometric studies have supported the predictive validity of spatial ability by documenting significant relationships between measures of spatial visualization ability and achievement in chemistry, geology, physics, architecture, engineering and medicine [1, 2, 3, 4, 5]. Furthermore, notable scientists have claimed that they achieved key discoveries from considering mental images of spatial phenomena [6, 7].

Although the ubiquity of diagrams across science domains may predispose problem solvers to engage in mental imagery, strategies applicable to problems that include diagrams are not limited to those that recruit imagistic thinking for two reasons. First, problems that include a diagram in the problem statement can be solved by a problem solver in an analytic fashion. For example, when Schwartz and Black [8] asked people to solve gear problems in which they had to determine which direction a particular gear in a gear chain would move, their gestures indicated that they initially mentally imagined the motion of each of the individual gears; however, on the basis of these simulations, they discovered the simple analytic rule that any two interlocking gears move in opposite directions. Once they had discovered this, participants then switched to a rule-based strategy and no longer reported the use of imagistic strategies to solve the problem.

Second, some scientific diagrams can help to relieve the problem solver from the burden of generating and manipulating an internal analogical mental image. In addition to providing an external representation, diagrams typically abstract from the reality that they represent so that they highlight some spatial aspects of the referent but not others. In chemistry, the wide variety of representations employed by chemists in professional and academic settings have resulted from the invention of unique diagrams for addressing specific conceptual challenges facing the community of chemical scientists. In each case, scientists created a diagram to reason about a phenomenon by representing spatial information in a highly abstracted fashion. For example, Melvin Newman [9] created the Newman projection (Fig. 1), which all chemistry students must learn to use today, to illustrate that any given molecule could assume unique spatial conformations that resulted from rotating specific bonds. Historically, Newman used his new representation, which depicts only some aspects of a molecule in two-dimensions while omitting most of the spatial information about the molecule, to show the scientific community that changes in the spatial conformation of a molecule produces strain on chemical bonds and alters the thermodynamics of a chemical reaction. Examples of the invention or adaptation of diagrams for problem solving, such as the Newman projection, are commonplace in the history of chemistry and often precede major developments in the field [10].

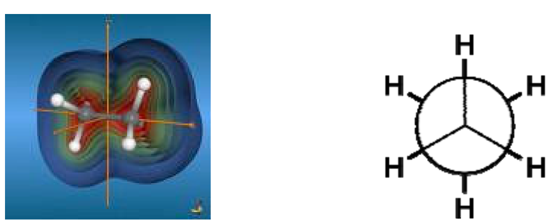



**Fig. 1.** Electron-density map (left) and Newman projection (right) of ethane

Recent analyses of student and expert problem solving in chemistry as well as spatial ability tests have revealed that problem solvers reason about the spatial information represented in both domain-specific scientific diagrams and domain-general spatial representations using a range of strategies. For example, problem solving with chemistry diagrams can involve mental rotation or analytical heuristics, and individuals who initially employ mental rotation can easily learn to use heuristics efficiently and effectively [11, 12]. Likewise, problem solvers often employ analytic strategies on spatial ability tests presumed to mandate strategies that involve the generation and manipulation of mental images. For example, tests of spatial ability typically include tasks such as mental rotation or mental paper folding which are assumed to measure ability to construct and transform mental images. However, there is a history of studies showing that people use a variety of strategies on these tests, including more analytic strategies that do not involve imagery [13, 14, 15, 16]. Although the distinct and interactive roles of alternative strategies for problem solving have been known for some time, a framework that systematically characterizes the features of each strategy remains outstanding and questions regarding the similarities among strategies and the enabling conditions for different strategies remain unanswered. For example, are algorithmic strategies applied exclusively to diagrams (external spatial representations) as opposed to internal spatial representations? Do mental rotation strategies require the holistic manipulation of a mental image? Can problem solvers employ analytic heuristics in tandem with perspective taking?

Here, we propose a framework for defining and distinguishing the range of strategies that problem solvers employ when engaged in spatial reasoning with diagrams. By spatial reasoning with diagrams we mean any mental process that infers new spatial information from information (spatial or non-spatial) illustrated in a diagram in the problem statement. The framework is derived from the analysis of verbal protocols, experiments and field observations of individuals working with diagrams on spatial ability measures and molecular representations in organic chemistry. We have constructed the framework to capture three unique features of spatial reasoning with diagrams: *spatial information use, representation use*, and *modification*. Below, we discuss each dimension of the framework and classify a range of strategies we have observed in each domain according to the framework.

## 2 A Proposed Framework for Characterizing Spatial Reasoning Strategies with Diagrams

We characterize problem solvers' strategies for reasoning spatially using diagrams with a framework that includes three dimensions: (1) what type of representation is used, (2) the degree to which spatial information in a representation is drawn upon and (3) how the relevant representation is manipulated. The first dimension of the framework places any spatial reasoning strategy on a continuum that defines the extent to which the solver relies upon use of internal versus external representations. On one end of the continuum, the problem solver may generate and construct a mental image from the external representation while problem solving. Conversely, the problem solver might rely only on the use of the external representation in the problem statement without attempting to generate an analog mental image. The second dimension of the framework identifies the degree to which spatial information is recruited or disregarded by the strategy. At one end of this dimension, a problem solver might rely heavily on spatial information in the problem statement by carefully considering the spatial relationships depicted in the given diagram. At the other end, a problem solver might focus on non-spatial information in the diagram, such as counting parts of the diagram, to solve the problem. Finally, the framework includes an additional dimension that defines how the problem solver modifies any relevant representation involved in the strategy. This dimension ranges from strategies that involve extensive modification, such as the construction of new external representations while problem solving, to those that involved no modification, such as strategies that involve reading information off an internal or external representation without transforming the representation in any way.

Fig. 2 illustrates the potential interactions between each of these three dimensions. Situating the axes orthogonally demonstrates the power of the framework to capture many strategies involved in spatial reasoning with diagrams. As stated above, we see these axes as representing continua that describe the features of any given strategy. Using the type of representation involved in a strategy as the primary characteristic of any strategy, it is possible to categorize any one strategy as relying primarily on external representations or internal representations, as has been done previously. However, we suggest that this dichotomy is overly simplistic and more a nuanced description of strategies for spatial reasoning with diagrams is needed. For example, strategies that involve transforming the spatial information in a given diagram via rotation of mental images share many features with strategies that employ algorithms to transform spatial information in diagrams.

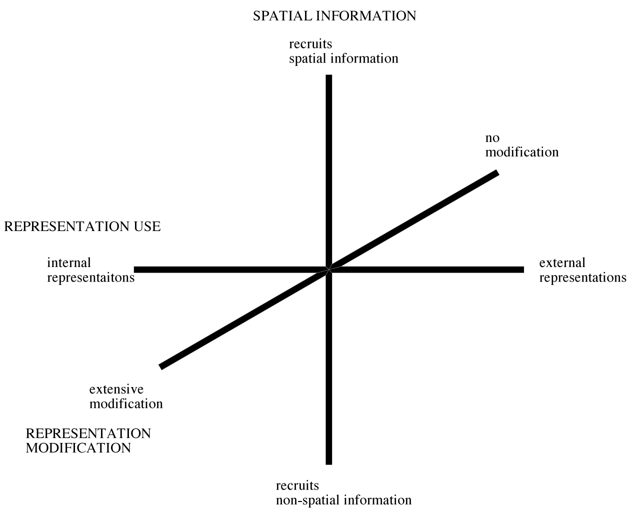



**Fig. 2.** Potential interactions between strategy characteristics

Using this framework, we have identified several common strategies used by problem solvers in chemistry and in the psychology laboratory while spatial reasoning with diagrams. Further, we have characterized the unique features of these strategies according to the three dimensions of the framework, and by aggregating strategies across domains we ultimately propose general classes of strategies for spatial reasoning with diagrams. Here, we review several examples of strategies reported by problem solvers on spatial ability measures and chemistry achievement assessments, and characterize them according to the framework.

## 3 Alternative Strategies for Solving Mental Rotation Tasks on a Spatial Ability Tests

In recent studies Hegarty and colleagues [17, 18], asked students to think aloud while solving items from spatial abilities tests, including the Vandenberg Mental rotations test [19]. This test is based on the classic Shepard and Metzler [20] mental rotation task, which shows that response time to decide whether two objects have the same shape is proportional to the angle of rotation and offers perhaps one of the strongest pieces of evidence for analog mental imagery in the psychological literature. The format of the Vandenberg test is somewhat different in that people are shown a standard on the left and 4 items on the right. Their task is to decide which of the 4 objects on the right have the same shape as the object on the left. This is probably the most commonly used test of spatial ability, possibly because it exhibits robust sex differences [21] and is usually assumed to measure facility in analog mental imagery. The test is made up of 2 sections in which students are given 3 minutes to solve 10 test items. Notably students are not allowed to draw on the test, so strategies that involve manipulating an external representation are not applicable on this test.

In one study students took the first section of each test under the normal timing conditions and then gave a think-aloud protocol while solving the items in the second section. On the basis of these protocols, we identified several strategies that students used in each of the tests. Interestingly, these included both strategies that relied on mental images, which the test purportedly requires problem solvers to use, and strategies that depended on more abstract and even non-spatial information.

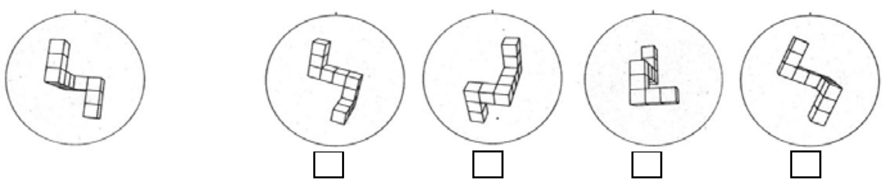


**Fig. 3.** Example of an item from the Vandenberg Mental Rotation test

*Mental rotation.* As expected, mental rotation was the primary strategy reported by several subjects. Consistent with the established features of mental rotation strategies, participants reported generating and manipulating analog mental images of the diagrams presented in each task, as exemplified by the following retrospective protocol:

> "I took the block in the circle... and tried to reposition them in my head, visualizing it... trying to match it to each... um multiple choice quest...question."

*Perspective Taking.* As found in previous studies of mental rotation tasks (e.g., [15], a minority of students reported a perspective taking strategy on these problems. That is, they reported imagining the objects in the problem as stationary, while they moved around the objects to view them from different perspectives. The following protocol is an example:

> “Uh... well... try to imagine I could look from a different perspective... and that's uh... three dimensional...Object... would go around... and try to imagine what it would look like... from different angles...”

Both mental rotation and perspective taking are imagistic strategies, in which students report imagining and transforming a mental image, although we cannot be sure how detailed or vivid their mental images were. Furthermore participants might mentally rotate the shapes all at once (holistic rotation) or piecemeal. In contrast to these imagistic strategies, we also identified several strategies in which participants clearly abstracted a subset of the information shown in the printed diagrams (either spatial or non spatial) given in the problems, and used more analytic heuristics to solve the problems.

*Comparing Arm Axes.* Another common strategy was to abstract the directions of the different segments or “arms” of the figures and compare the relative directions of the arms in the standard figures to that in each of the four answer choices. There were two versions of this strategy. In one version, the directions of all four arms of the figures were noted and compared to the answer choices. In the other version of this strategy, which was previously identified by Geiser [13] participants compared the direction of only the two end arms of the figures, as exemplified by the following protocol.

> “well the first strategy to eliminate... was looking at the end blocks to see if... what direction they were pointing in...And then... often but not always eliminate I...I... one to two...”

This strategy highlights a difference between some items on this paper-and-pencil test and the reaction time task used by Shepard and Metzler [20]. In the Shepard and Metzler task, the foils are always mirror images. However, in the Vandenberg Mental rotation test, 35% of the foils differ from the standard object in shape, and this can be detected by examining the two end arms of the object. For example in the item in Figure 3, it can be seen that in the standard object on the left, the two end arms are parallel to each other, whereas for the two answer choices on the right, the two ends are perpendicular.

The strategies of comparing arm axes differ from the mental rotation and perspective taking strategies listed above in that less spatial information was internally represented while solving the problems. While the imagistic strategies presumably represent the whole object, these strategies only represent some aspects of the objects in the problems, notably schematic spatial information about the relative directions of parts of the object. In these strategies, the information that is recruited to solve the problem was always spatial.

*Counting Cubes.* The final strategy we will discuss here was to count the number of cubes in each segment of the object and compare that to the other objects in the problem. An example of a protocol of someone using this strategy is as follows:

> "But... I tried to like... I tried to cancel out the ones I new couldn't be right by looking at the number of squares... So like here I saw it was like three interrupted by four, interrupted by three all in one field... so like I could quickly... you know count out the ones that was like three and four."

In terms of the dimensions shown in Figure 2, this is an example of a strategy in which, again, only a subset of the information in the external diagram is internally represented. But in contrast to comparing arm axes, the information recruited in this strategy is non-spatial (numbers of cubes). Although spatial information is displayed in the given external representation, the strategy does not rely on this information to generate a resolution. Rather, the strategy involves direct consideration of the subcomponents of the external representations presented in the task. Of course, a corresponding internal numerical representation of some kind is certainly involved in this strategy, but our framework constrains analyses of problem solving to spatial representations (i.e., images or diagrams). Speculating about the nature of the numerical representation is beyond the scope of this paper.

In a follow up study to our verbal protocol study (see [17]), we created "strategy choice" questionnaires that listed the different strategies identified in the protocols and a second group of 37 students (18 male, 19 female) were asked to complete these strategy choice questionnaires after taking a computer administered versions of the Vandenberg Mental Rotation tests. In the strategy surveys, most students reported using a mental imagery strategy (either imagining the rotation of the objects or imagining changing their perspective with respect to the objects), but they typically also reported one of the other strategies so that the strategies used by most student would be classified as "mixed". In fact students reported using reported 3.46 different strategies on average to complete the 20 items on the test. The number of strategies used was not significantly correlated with test score ($r = .18$), but students who reported the strategy of comparing the directions of the two end arms of the figure had higher scores on the test.

In summary, although people often assume that the Vandenberg Mental Rotations strategy measures ability to perform analog transformations on mental images, we found that people use a range of strategies from more imagistic to more abstract and analytic to solve these problems and the majority of solvers used a combination of imagistic and more abstract and analytic strategies to solve the test items.

## 4 Alternative Strategies for Solving Stereochemistry Tasks on Organic Chemistry Achievement Assessments

The study of organic chemistry follows the reactivity of molecules, and just like the shape or structure of a chair defines its use or function for sitting, the shape of a molecule often defines how it reacts. The structure of organic molecules is shown in the two-dimensional plane by using a number of different representations including the dash-wedge notation shown in Fig. 4 and the Newman projection shown in Figure 1. The dash-wedge representation, familiar to chemists, follows the convention that, relative to the plane of the page, groups on wedges represent bonds coming out of the plane whereas groups on dashes represent bonds behind the plane.

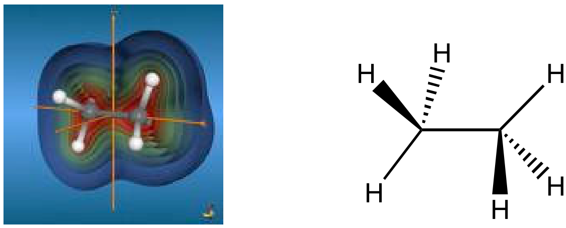



**Fig. 4.** Electron-density map (left) and dash-wedge notation (right) of ethane

Recently, we have examined student use of dash-wedge formulas for solving a class of stereochemistry tasks on organic chemistry achievement assessments [22, 12, 23]. These tasks ask students to consider the spatial relationships between atoms within and between asymmetric molecules. Like the Vandenberg Mental Rotation test, discussed above, typical organic chemistry assessment items regarding stereochemistry ask students to compare pairs of molecules to determine whether the two molecules in the pair are identical or mirror images of one another. Unlike the Vandenberg test, stereochemistry tasks in chemistry often require the student to compare and contrast multiple representations of the same molecule in the problem statement (see Fig. 5). The problem in Fig. 5 asks the student to compare the three structures represented by the given diagrams and determine which two are identical (i.e., show the same molecule). Briefly, the two Newman projections are equivalent to a two-dimensional representation of the dash-wedge perspective formula viewed from the left or right side (90 degree rotation to the right or left of the view shown in the dash-wedge perspective). They also use different conventions to represent depth. For example they use occlusion to indicate which part of the molecule is in front and which is behind, and they do not use the dash-wedge convention. Choice A is equivalent to the diagram on the left as it possesses all of the same internal spatial relationships among the represented atoms.

In one recent analysis of student verbal protocols and classroom problem solving, we asked students to review the task in Figure 4 and solve the problem while speaking aloud without time restrictions. On the basis of these protocols and field observations of the participants working classroom settings, we have identified four distinct strategies that students use to solve the problem. As on the Vandenberg Mental Rotation Test, these strategies range from those that recruit spatial information in mental images to those that involve manipulation of external diagrams.

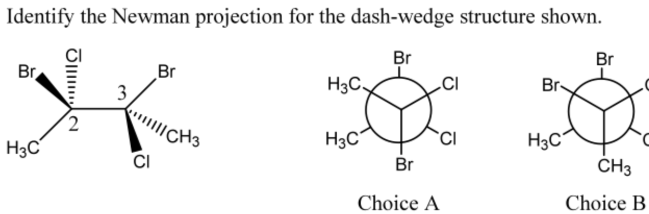



**Fig. 5.** A typical stereochemistry assessment that asks the student to determine which of the two Newman projections (right) is identical to the dash-wedge formula (left)

*Mental Rotation.* Several students were seen to solve the problem by imagining the dash-wedge formula and rotating into alignment with both Choice A and Choice B, as indicated in the following protocol.

> "I'm thinking of the original would be similar to A because of thinking of rotating the back and it seems like they would be similar. The bonds are the same, you can move the molecule around by just rotating the back bonds."

We observed two distinct mental rotation strategies used by students to solve the problem. The first strategy involved rotation of the entire dash-wedge formula by imagining rotating all groups around the central C2-C3 bond, as if participants were imagining handling a physical three-dimensional model of the molecule, until the mental image matched either A or B. We have identified this strategy as a holistic mental rotation strategy since the problem solver reports manipulation of the mental image during the process of solving the problem. Alternatively, we observed problem solvers employ strategies where they did not rotate the entire image to conclude that A was the correct answer. Instead, they engaged in the piecewise rotation of part of the molecule. As indicated in the protocol above, a simple clockwise rotation of the carbon (labeled 2 in Fig. 5) allowed participants to determine that A possessed the same spatial relationships of the attachments to C2 as the original diagram.

*Perspective Taking.* In contrast with the mental rotation strategy, which involved rotating a mental image of the diagram, we observed strategies with which the problem solver chose to imagine the given diagram fixed in space before then imagining moving around the molecule to ascertain information regarding spatial relationships. The following protocol illustrates one student's attempt at perspective taking.

> "It's as if you have to look at it from this way…then you just come over and find out which is the same."

Given that a Newman projections in Fig. 5 depicted the molecule from a different perspective (90 degree rotation to the right or left of the view shown in the dash-wedge perspective) some problem solvers reported solving the problem by imagining moving around to the right or left of the dash-wedge representation to determine which face presented the same spatial relationships shown in the Newman projections. In several cases, the students actually moved their head or entire body to a new perspective relative to the diagram.

The above two strategies relied extensively on inspecting internal mental images for problem solving. Further, each of these strategies involved directly considering the spatial information depicted in the image with extensive modification of the spatial relationships depicted via rotation or perspective taking. In contrast to these imagistic strategies, we also identified strategies in which participants clearly abstracted a subset of the information shown in the printed diagrams (either spatial or non spatial) given in the problems, and used more analytic heuristics to solve the problems.

*Re-representation.* Although the task in Fig. 5 presents the necessary diagrams to the student in the problem statement, we observed that some students construct new Newman projections from the given dash-wedge formula using an algorithm. In these cases, the student would construct a Newman from a perspective on the left side of the structure by insisting that all wedges should be transposed to the right side and all dashes should be transposed to the left side of a self-generated Newman projection. This strategy will always result in a Newman projection that maintains all the correct spatial relationships. The student's self-generated Newman projection was then compared to the two Newman projections given in the problem to determine identity. In essence, this strategy allowed the problem solver to keep track of all the spatial information in the molecule without constructing an internal spatial representation of the information in the problem to compare the relationships between the diagrams.

*Configurational Assignment.* We also noted that problem solvers who had more experience in the discipline display strategies similar to those seen among experts [12]. Using the R/S configuration assignment strategy, some problem solvers were able to assign labels to each of the given diagrams that identified the spatial relationships around specific atoms within each structure. The labels (known as configurational assignments in chemistry) distinguished molecular structures based on unique spatial features. As with the re-representation strategy, students applied the configurational assignments using an algorithm. The strategy can be applied to the dash-wedge formula in Fig. 5 to illustrate that carbons 2 and 3 have the same groups attached, yet each has a unique spatial configuration. Students who recognized this, would label the dash-wedge formula in a manner that highlighted that the Br group is to the right of the Cl group on C2, but to the left of the Cl group in C3. Applying the labels to Choice A and B as well allowed the students to determine that carbons 2 and 3 matched only in the dash-wedge formula and Choice A.

These latter two strategies relied exclusively manipulating external diagrams presented in the problem statement. As with mental rotation and perspective taking, the configurational assignment and re-representation strategies involved direct consideration of the spatial information presented in a diagram. The strategies were distinct from each other, however, in the extent to which they involved transformations of internal representations versus modification of external representations. Whereas mental rotation and perspective taking rely on transformations of internal representations, assigning a configuration involved adding minor labels to each diagram whereas re-representation involved the construction of a new representation of the problem. The latter two strategies could be accomplished without constructing any internal spatial representation (image) of the information in the problem (although they presumably depend on internal representations of rules). All of the strategies documented for the chemistry problem here are high on the "representation modification" dimension of Figure 2; likewise, all involve extensive recruitment of spatial information.

In summary, we observed students to use several unique strategies to solve this chemistry achievement assessment task. While some strategies appeared to rely many on inspecting and manipulating an internal mental images others relied on the simple labeling of external representations. Recently, we have developed a survey to assess students' preference for each strategy on a range of tasks included in organic chemistry. The preliminary results of that work indicate that students begin organic chemistry with a preference for using mental rotation strategies on most tasks, but display a preference for strategies that involve manipulating diagrams when they finish instruction [22].

## 5 Discussion

In the present paper we have explored the plausibility of a new framework for characterizing and classifying problem solving strategies for spatial reasoning with diagrams. Building on prior work that categorizes spatial problem solving strategies as analytic versus imagistic, we have attempted to define strategies according to three unique characteristics: use of internal versus external representations, recruitment of spatial information and modification of representations. Thus, the framework places strategies on a multi-dimensional space that illustrates while some strategies may be discrete many strategies have similar features.

The two types of problems analyzed in detail here also point out that the importance of the three dimensions in Figure 2 may differ for different domains. In the case of the mental rotations problems, the strategies we examined differed primarily in the "spatial information" dimension, with mental rotation and perspective taking relying on all the spatial information given in the external diagrams, In contrast "comparing arm axes" recruits only a subset of the information in the external diagrams and "counting cubes" does not recruit spatial information. In the case of the chemistry problem, the strategies differed primarily in the "representation use" dimension in Figure 2, with mental rotation and perspective taking relying on internal representations and re-representation and configurational assignment relying on modification of external representations.

Analyzing each of the strategies above according to the three dimensions of representational use, representational modification and spatial information reveals that although many strategies share similar characteristics, they are not equivalent. That is, discriminating strategies based on whether they rely primarily on internal representations or external representations does not sufficiently capture the unique aspects of certain strategies. For example, perspective taking and mental rotation both rely on the inspection of internal mental images yet represent distinct approaches to problem solving. Likewise, strategies that involve the modification of an external representation may or may not attend to spatial information in a given diagram.

Of course, our framework has a few notable limitations. First, the framework fails to capture the interactive role of multiple strategies during problem solving. While we have attempted to illustrate the distinct features of unique strategies, it is certainly true that problem solvers may employ several strategies in search of solution. We offer that such hybrid strategies may result from the cumulative application of discrete strategies identified above. Likewise, we are certain that we have not gathered evidence to provide an exhaustive list of strategies that populate the space in Fig. 2. In our own work we have observed several additional strategies that can be represented in this space that are unique to certain tasks both in chemistry and in the psychology laboratory. Finally, we recognize that it is quite possible that the extent to which our axes in Fig. 2 are orthogonal requires empirical support.

Ultimately, our framework suggests that there are distinct and interactive roles for imagistic, diagrammatic and analytic strategies for problem solving in the psychology laboratory as well as the science classroom and that the unique features of each strategy requires further study. For example, strategies that rely primarily on viewing and manipulating an internal mental image of a diagram are frequently reported by novices in domain; however, strategies that involve the modification of non-spatial information in diagrams are reported more frequently as expertise develops [8, 11, 12, 22, 23]. While such distinctions are well documented at the extremes of the expert-novice continuum, little is known about how students develop strategies for reasoning about spatial information in authentic settings or whether specific strategies are used at certain points while learning. Moreover the interaction between a problem-solvers' spatial ability and strategy use might be further examined using the above framework. The availability of alternative strategies for solving spatial tasks such as those above suggest that students who perform exceptionally well on spatial ability tests may be recruiting strategies that allow them to problem solve successfully without using spatial information in the task. Ultimately, our observations of alternative strategy use in both chemistry and the psychology laboratory illustrate that spatial reasoning with diagrams can be accomplished in several ways and requires neither imagistic nor analytic strategies exclusively.

**Acknowledgements.** The authors would like to thank Andrew Stull and Minjung Ryu for contributing to data collection and analysis activities as well as input on the framework. This work was supported in part by the National Science Foundation (DRL-0723313). Any opinions, findings or conclusions expressed in this paper are those of the authors and do not necessarily represent the views of NSF.

## References

1. Casey, M.B., Nuttall, R.L., Pezaris, E.: Mediators of Gender Differences in Mathematics College Entrance Test Scores: A Comparison of Spatial Skills with Internalized Beliefs and Anxieties. Dev. Psychol. 33, 669–680 (1997)
2. Coleman, S.L., Gotch, A.J.: Spatial Perception Skills of Chemistry Students. J. Chem. Educ. 75, 206–209 (1998)
3. Keehner, M.M., Tendick, F., Meng, M.V., Anwar, H.P., Hegarty, M., Stoller, M.L., Duh, Q.: Spatial Ability, Experience, and Skill in Laparoscopic Surgery. Am. J. Surg. 188, 71–75 (2004)
4. Kozhevnikov, M., Motes, M., Hegarty, M.: Spatial Visualization in Physics Problem Solving. Cognitive Sci. 31, 549–579 (2007)

5. Orion, N., Ben-Chaim, D., Kali, Y.: Relationship between Earth-Science Education and Spatial Visualization. J. Geoscience Educ. 45, 129–132 (1997)
6. Rothenberg, A.: Creative Cognitive Processes in Kekule's Discovery of the Structure of the Benzene Molecule. Am. J. Psychol. 108, 419–438 (1995)
7. Watson, J.D.: The Double Helix. Penguin, New York (1968)
8. Schwartz, D.L., Black, J.B.: Shuttling between Depictive Models and Abstract Rules: Induction and Fall-Back. Cognitive Sci. 20, 457–497 (1996)
9. Newman, M.: A Useful Notation for Visualizing Certain Stereospecific Reactions. Record Chem. Prog. 13, 111 (1952)
10. Goodwin, W.M.: Structural Formulas and Explanations in Organic Chemistry. Found. Chem. 10, 117–127 (2008)
11. Stieff, M.: Mental Rotation and Diagrammatic Reasoning in Science. Learn. Instr. 17, 219–234 (2007)
12. Stieff, M., Raje, S.: Expert Analytic and Imagistic Problem Solving Strategies in Advanced Chemistry. Spatial Cognition Comput. (in Press)
13. Geiser, C., Lehmann, W., Eid, M.: Separating "Rotators" from "Non Rotators" in the Mental Rotations Test: A Multivariate Latent Class Analysis. Multivar. Behav, Res. 41, 261–293 (2006)
14. Hegarty, M., Waller, D.: Individual differences in spatial abilities. In: Shah, P., Miyake, A. (eds.) Handbook of Visuospatial Thinking, pp. 121–169. Cambridge University Press, Cambridge (2006)
15. Just, M.A., Carpenter, P.A.: Cognitive Coordinate Systems: Accounts of Mental Rotation and Individual Differences in Spatial Ability. Psychol. Rev. 92, 137–172 (1985)
16. Lohman, D.F.: Spatial Abilities as Traits, Processes, and Knowledge. In: Sternberg, R.J. (ed.) Advances in the Psychology of Human Intelligence, pp. 181–248. Erlbaum, Hillsdale (1988)
17. Hegarty, M.: Components of Spatial Intelligence. In: Ross, B.H. (ed.) The Psychology of Learning and Motivation, vol. 52, Academic Press, San Diego (in Press)
18. Hegarty, M., De Leeuw, K., Bonura, B.: What Do Spatial Ability Tests Really Measure? In: Proceedings of the 49th Meeting of the Psychonomic Society. Psychonomic Society, Chicago (November 2008)
19. Vandenberg, S.G., Kuse, A.R.: Mental Rotations: a Group Test of Three Dimensional Spatial Visualization. Percept. Motor Skill. 47, 599–601 (1978)
20. Shepard, R.N., Metzler, J.: Mental Rotation of Three-Dimensional Objects. Science 171, 701–703 (1971)
21. Voyer, D., Voyer, S., Bryden, M.P.: Magnitude of Sex Differences in Spatial Abilities: A Meta-Analysis and Consideration of Critical Variables. Psychol. Bull. 17, 250–270 (1995)
22. Stieff, M., Ryu, M., Dixon, B.: Students' Use of Multiple Strategies for Spatial Thinking in Chemistry (manuscript submitted for publication) (2010)
23. Stieff, M.: When Is a Molecule Three-Dimensional? A Task-Specific Role for Imagistic Reasoning in Advanced Chemistry. (manuscript submitted for publication) (2010)

# Relating Two Image-Based Diagrammatic Reasoning Architectures

Michael Anderson[1] and George Furnas[2]

[1] Department of Computer Science, University of Hartford, West Hartford, CT 06117 USA
anderson@hartford.edu
[2] School of Information and Department of Computer Science and Engineering, University of Michigan, Ann Arbor, MI 48104 USA
furnas@umich.edu

**Abstract.** To advance the understanding of different diagrammatic reasoning architectures that reason directly with images, we examine the relationship between Anderson's Inter-Diagrammatic Reasoning (IDR) architecture and Furnas' BITPICT architecture using the technique of cross-implementation. Implementing substantial functionality of each in the other, and noting what is easy and what is difficult, yields insights into the two architectures and such systems in general.

**Keywords:** Diagrammatic Reasoning, Depictive Representations, IDR, BITPICT.

## 1 Introduction

Among the numerous diagrammatic reasoning architectures, relatively few reason directly with images. These include Lindsay's ARCHIMEDES[1][2], Jamnik's DIAMOND [3], Furnas' BITPICT[4][5][6][7][8], and Anderson's Inter-Diagrammatic Reasoning (IDR) architecture[9][10][11][12]. As an important aspect of any field is the clear delineation of the work within it, we investigate the relationship between the two of these that are arguably most similar, BITPICT and IDR. Both attempt to reason *solely* with diagrams, with exploration sharply focused on the exclusive use of depictive diagrammatic representations to bring our understanding of their computational power on par with other, better-understood more sentential representation methods.

BITPICT and IDR also differentiate themselves from the other architectures listed above in that they are domain-independent. Where the other architectures have focused on the use of diagrams in mathematical reasoning, BITPICT and IDR have been used in a wide variety of domains in various applications. IDR for example has been used for game playing, heuristic development, learning from diagrams, and most recently, formulating a polynomial graphical solution for deadlock avoidance [13]. BITPICT has been used for graphical simulation of physical and abstract processes, for game play and puzzle solving, novel shape-rich human-computer interactions, and topological analyses, and has recently been incorporated into a general purpose, SOAR-based spatial reasoning architecture [14][15]. BITPICT has been shown to be Turing equivalent [16] and IDR, situated within Lisp, can surely claim the same.

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 128–143, 2010.

BITPICT and IDR differ in their approaches to reasoning directly with diagrams. BITPICT reasons from diagram to diagram where IDR reasons over collections of related diagrams. BITPICT commits to a particular control structure, a graphical rule-based production system, where IDR makes no such commitment, simply offering a small set of graphical primitives. To clarify the significance of these similarities and differences we performed a cross-implementation exercise. After describing the BITPICT and IDR architectures, we show how IDR operators can be used to implement BITPICT rules, how BITPICT rules can be used to implement IDR operators, and, finally, discuss what was learned.

## 2 Furnas' BITPICT Architecture

The BITPICT architecture explores direct inferencing from image to image, functioning like a visual expert system shell with exclusively graphical production rules that modify a pixellated, graphically represented knowledge base. The left hand side (LHS) of each rule contains a small pixel pattern ("bitpict"), which, when found in an image, is replaced by a comparable bitpict pattern from the rule's Right Hand Side (RHS). The rules fire repeatedly until there are no more matches, in this way performing a computation that "deduces" the consequence of the original picture under the action of the rules.

For example, Fig. 1 shows a BITPICT naive physics simulation. The leftmost image, Fig. 1a, is comprised of "balls" and "ramps" and serves as the starting point. Two simple BITPICT rules (Fig. 2) govern the balls' behavior. In Rule 1, a ball with nothing below it falls down one pixel. In Rule 2, a ball touching any small section of ramp slides down and to the side. After the exhaustive application of these graphical rewrite rules, the balls come to rest (Fig. 1c). (An additional rule could disperse balls as they pile up.) Throughout the simulation, the balls behave expectedly – falling, sliding

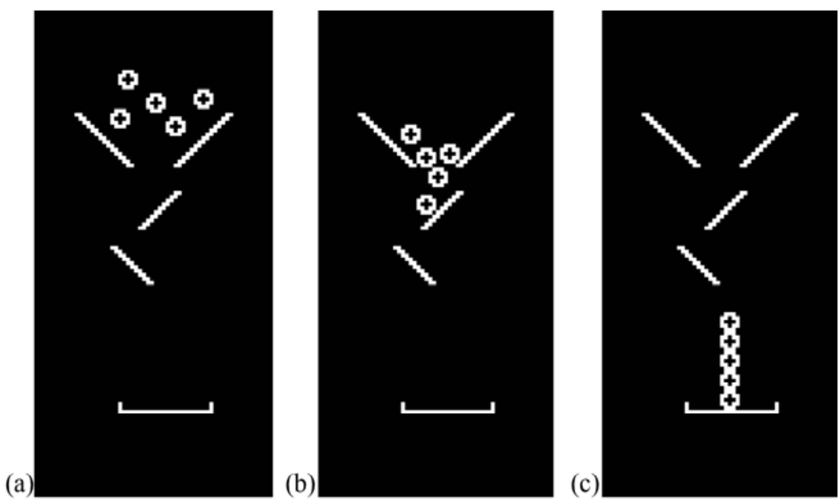



**Fig. 1.** BITPICT example: Balls and Ramps

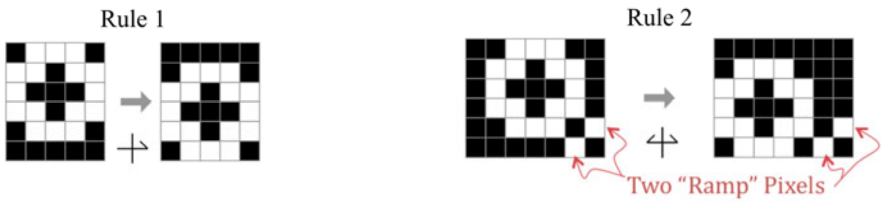



**Fig. 2.** BITPICT rules for the Balls and Ramps example. In Rule 1, a "ball" falls one pixel. In Rule 2, a ball in contact with a ramp, slides down and to the side. The glyph annotations below the rule arrows indicate the geometric transformations allowed in the match: *xy*-translations only for Rule 1 and *xy*-translations plus reflection around the y-axis for Rule 2.

down ramps, waiting their turn while other balls fall through openings, and even emptying from the hopper faster if its opening is drawn wider.

The underlying rule engine can allow geometric transformations in the matching. In Fig. 2, both rules permit *xy*-translation (matching at any location), and Rule 2 permits reflection around the *y*-axis, allowing the balls to slide either right or left on ramps. The underlying rule engine also takes care of conflict resolution, needed whenever there is more than one match in a picture. BITPICT uses a serial execution model – only one rule fires at a time – and an explicit, configurable conflict resolution procedure is invoked to choose a winner amongst all current matches. The dominant factor is typically rule priority, but may also include probabilistic weighting, and the age of the match. In general BITPICT uses many gray-scales and colors for its pixel values (not just black and white as in Fig. 1), but they are treated as nominal scale: only equality/inequality is defined (all that is needed for matching); they are given no additional ordered or arithmetic properties. As we will see, the special responsibilities of the underlying engine, to take care of geometric transformations and resolve conflicts, and the nominal scale of the pixel values all have implications for the cross-realization of the two systems.

## 3 Anderson's Inter-diagrammatic Reasoning Architecture

*Inter-Diagrammatic Reasoning* (IDR) defines diagrams as *tessellations* (tilings of finite subsets of two-dimensional space). Individual *tesserae* (tiles) are atomic and take their values from an *I, J, K*-valued subtractive CMY (Cyan, Magenta, Yellow) color scale. Intuitively, these CMY color scale values (denoted $v_{i,j,k}$) correspond to a discrete set of transparent color filters where $i$ is the cyan contribution to a filter's color, $j$ is the magenta contribution, and $k$ is the yellow contribution. Overlaid, these filters combine to create other color filters from a minimum of $v_{0,0,0}$ to a maximum of $v_{I-1,J-1,K-1}$. A tessellation whose tesserae are all $v_{0,0,0}$-valued is the *null tessellation* (denoted $\varnothing$); a tessellation whose tesserae are all $v_{I-1,J-1,K-1}$-valued is the *maximum tessellation* (denoted $\infty$).

IDR leverages the spatial and temporal coherence often exhibited by groups of related diagrams for computational purposes. Like diagrams are combined in ways that produce new like diagrams that infer information implicit in the original diagrams. The following IDR primitives take two diagrams, $d_1$ and $d_2$, of equal dimension and tessellation and return a new diagram where each tessera has a value $v_{i, j, k}$ that is some function of the values of the two corresponding tesserae, $v_{i1, j1, k1}$ and $v_{i2, j2, k2}$ , in the operands.

- OR, denoted $d_1 \vee d_2$, returns the maximum of each pair of corresponding tesserae, defined as $v_{max(i1, i2), max(j1, j2), max(k1, k2)}$.
- AND, denoted $d_1 \wedge d_2$, returns the minimum of each pair of corresponding tesserae, defined as $v_{min(i1, i2), min(j1, j2), min(k1, k2)}$.
- OVERLAY, denoted $d_1 + d_2$, returns the sum of each pair of corresponding tesserae, defined as $v_{min(i1+i2, I-1), min(j1+j2, J-1), min(k1+k2, K-1)}$.
- PEEL, denoted $d_1 - d_2$, returns the difference of each pair of corresponding tesserae, defined as $v_{max(i1-i2, 0), max(j1-j2, 0), max(k1-k2, 0)}$.

The following unary operators, binary operators, and functions complete the set of basic tools for the process of IDR:

- NOT, denoted $\neg d$, is a one place operator that returns the value of $\infty$ - *d*, where $\infty$ (the black diagram) denotes a diagram equal in tessellation to *d* containing all BLACK-valued tesserae.
- NULL, denoted $\eta(d)$, is a one place Boolean function taking a single diagram that returns **TRUE** if $d = \emptyset$, else it returns **FALSE**.
- ACCUMULATE, denoted $\alpha(d, ds, o)$, is a three place function taking an initial diagram, *d*, a set of diagrams of equal dimension and tessellation, *ds*, and the name of a binary diagrammatic operator, *o*, that returns a new diagram which is the accumulation $((d \text{ o } d_1) \text{ o } d_2) \text{ o } ...)$, of the results of successively applying *o* to *d* and each diagram in *ds*.
- MAP, denoted $\mu(f, ds_1 , ... , ds_n)$, is an *n+1* place function taking an *n*-place function, *f,* and *n* sets (of equal cardinality) of diagrams of equal dimension and tessellation, $ds_i$, that returns the set of values resulting from the application of *f* to each corresponding *n* diagrams in $ds_1, ... , ds_n$.
- FILTER, denoted $\phi(f, ds)$, is a two place function taking a Boolean function, *f,* and a set of diagrams of equal dimension and tessellation, *ds*, that returns a new set of diagrams comprised of all diagrams in *ds* for which *f* returns **TRUE**.

## 4 IDR Realization of Furnas' BITPICT System

Inter-diagrammatic reasoning operators can be used to realize Furnas' BITPICT system. The graphical rewrite rules are fully instantiated into sets of *n* ordered triples of diagrams, $\{(LHS_1, Mask_1, RHS_1), ... , (LHS_n, Mask_n, RHS_n)\}$. In the unusual event that the rule allows no transformations (not even translations), there will be only one triple in the set. Otherwise the set contains a triple for each transformed version of the nominal rule (at most all possible *x*-translations × possible *y*-translations × 2 vertical reflections × 2 horizontal reflections × 4 rotations). Each $LHS_i$ is a diagram that corresponds to a particular transformed instance of the LHS of the rule being realized; each $RHS_i$ is the corresponding RHS. Each $Mask_i$ is a diagram that is completely foreground color (white, as only black and white bitpicts will be considered) except for

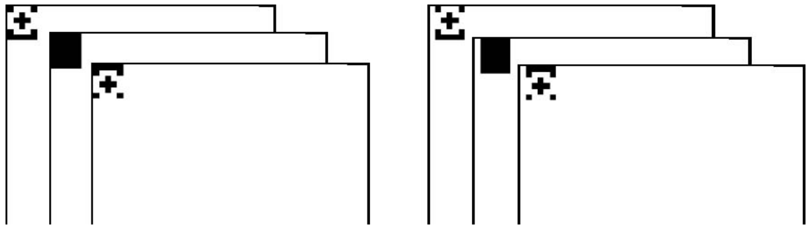


**Fig. 3.** BITPICT Balls and Ramps example in IDR. Depicted are two of the ordered triples of diagram produced for the realization of graphical rewrite Rule 1 of Fig. 2. The left column, from top to bottom represents the triple ($LHS_1$, $Mask_1$, $RHS_1$) at location (0, 0). The right column, from top to bottom, represents the triple ($LHS_2$, $Mask_2$, $RHS_2$) at location (1, 0). A set of these triples is required for each *x,y* location to represent the *x,y* translation of the corresponding BITPICT rule.

the portion of the diagram that is currently being matched which is background-colored (black, in the example), permitting each $Mask_i$ to mask away all but the pertinent part of the image to which the rules are being applied. This pertinent part is compared with $LHS_i$ and, if it is equal to it, it is replaced with $RHS_i$. In the case where rules do not conflict, each rule set, then, is simply passed through with each $LHS_i$ being compared to the masked image to see whether or not its pixels match to the image's pixels and, when they do, those pixels are replaced with those in the corresponding $RHS_i$. When no more matches can be found, processing stops.

To accomplish this using IDR operators, each $Mask_i$ is ANDed with the image in question producing an image that is completely foreground color in the areas not currently of concern while leaving the pertinent area unchanged. The resulting image is then compared for equality with $LHS_i$ by ORing the diagrammatic difference of it and $LHS_i$ with the diagrammatic difference of $LHS_i$ and it. If the resulting image is NULL, the diagrams are equal and the matching part is removed from the image by ANDing a negated version of $Mask_i$ to it. The result is ORed with $RHS_i$ to complete the replacement.

Given that *rule* denotes the set of triples for a given rule and *d* denotes the image to which the rule is to be applied, the following returns the set of all triples whose LHS matches the image in its corresponding position:

$$matches := \phi( \lambda(x)\ (x.LHS = x.Mask \wedge d),\ rule). \qquad (1)$$

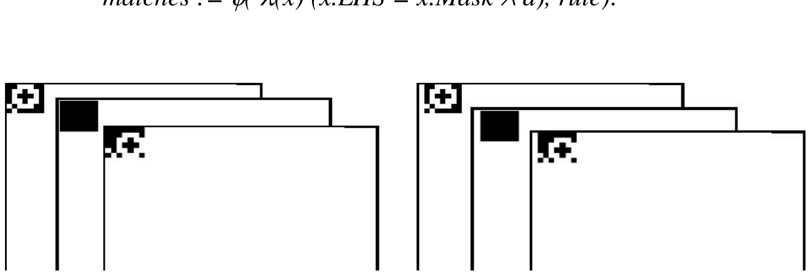


**Fig. 4.** BITPICT Balls and Ramps example in IDR (cont). Depicted are two of the ordered triples of diagram produced for the realization of graphical rewrite Rule 2 of Fig 2.

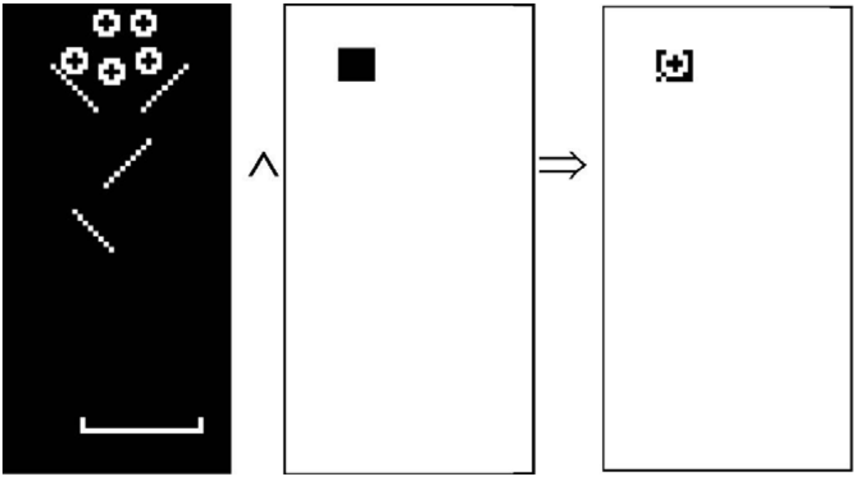


**Fig. 5.** BITPICT Balls and Ramps example in IDR (cont). Mask is ANDed with image to isolate the currently pertinent portion of the image to which the rewrite rules are being applied.

Given *MasksOf* and *RHSOf* are functions that return the set of Masks and RHSs, respectively, of the set of triples passed to them, the following then uses these sets to replace each appropriate portion of *d* with its corresponding RHS:

$$d := \alpha(\neg\alpha(\varnothing, MasksOf(matches), \vee) \wedge d, RHSOf(matches), \vee). \tag{2}$$

Less formally, as *matches* is the set of all rule sets whose LHS matched the image, the masks of these rules are accumulated onto a single diagram. This diagram is then

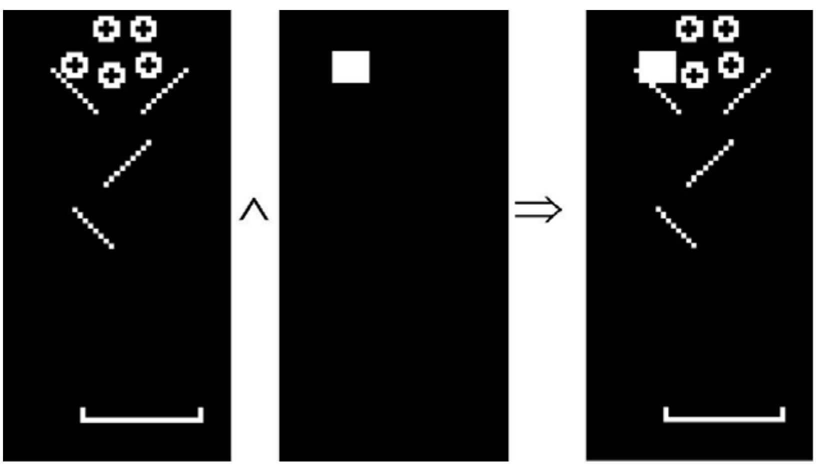


**Fig. 6.** BITPICT Balls and Ramps example in IDR (cont). Negation of a mask is ANDed with image to remove the portion of the image that is to be replaced.

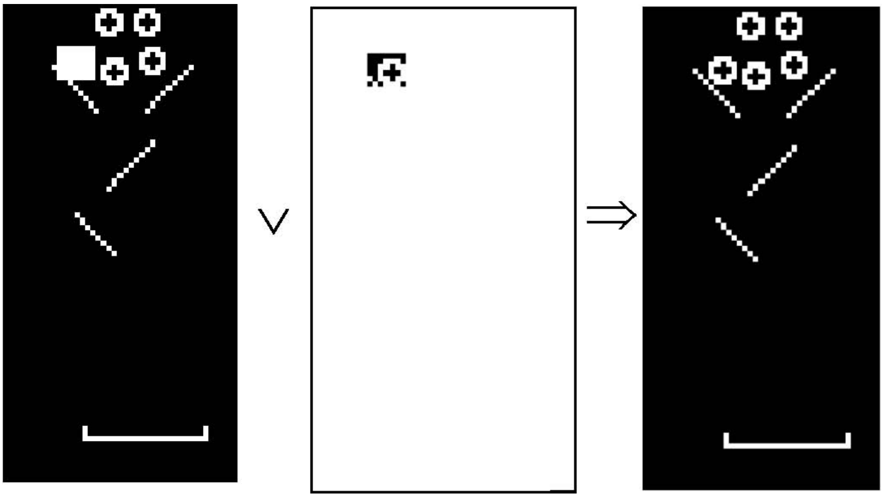



**Fig. 7.** BITPICT Balls and Ramps example in IDR (cont). RHS of rewrite rule is ORed with image from previous figure, updating image to reflect rule firing.

negated and used to remove portions of the image that are to be replaced. The RHSs of all these matched rules are then overlaid upon this modified image created the final result.

To accommodate cases where rules may conflict, such conflict can be recognized by finding rule sets in matches whose masks are overlap. Sets of conflicting rule sets are removed from matches, reduced to a single rule set by some conflict resolution routine, and then inserted back into matches before application. When rules are prioritized, the rules sets in matches can be ordered by some priority measure and applied iteratively (as opposed to the parallel fashion in which they are applied above) to the image in this order.

Figs. 3 - 7 delineate this process for the "balls and ramps" example previously discussed. Fig. 3 shows two of the ordered triples of diagrams produced for the realization of graphical rewrite Rule 1 of Fig. 2. Fig. 4 shows the ordered triples of diagrams produced for the realization of graphical rewrite Rule 2 of Fig. 2. Fig. 5 shows the process of using the Mask to isolate the currently pertinent portion of the image to which the rewrite rules are being applied (leftmost image of Fig. 1). The result of this process is then checked for equality with the current LHS. Since in this example this is the case, the negation of the Mask is used to remove the pertinent portion of the image as depicted in Fig. 6. Finally, the image is updated by ORing the RHS with the result of the previous operation as depicted in Fig. 7.

## 5 BITPICT Realization of Anderson's IDR System

Much basic diagrammatic functionality in IDR has a direct implementation in BITPICT. As we will see, however, complexities arise from some high-level language infrastructure which IDR gets from its embedding in Lisp. While one could perhaps

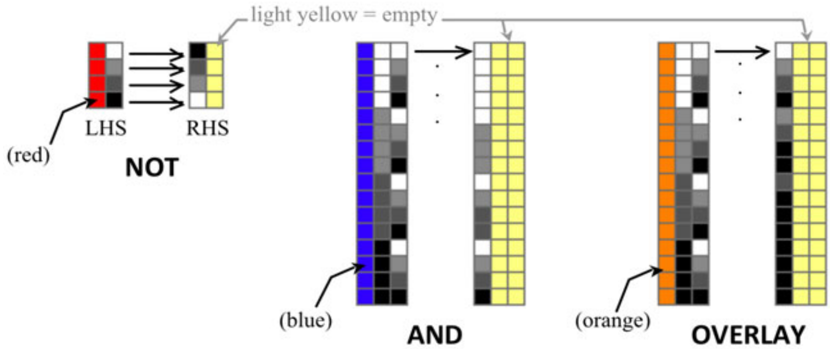



**Fig. 8.** BITPICT rules for NOT, AND and OVERLAY for an IDR system with four grayscale levels. Each row in the figure illustrates the action of a different rule, one for each row of the function's "truth table". Thus there are four rules for NOT and sixteen each for AND and OVERLAY. Similar rules are easily defined for OR and PEEL.

add corresponding high-level constructs to BITPICT (e.g., see [16]), at the end of this section we will instead sketch ways some of that functionality might be obtained with purely diagrammatic constructs in BITPICT.

Presenting the BITPICT versions in a static black and white medium is nearly impossible (like typical SIGGRAPH animation work). The rules use many colors and their workings are highly dynamic, making myriad changes that can only be suggested in a few static pictures. Here we will use only four levels of gray in the IDR model and at least some colors will be annotated, but the reader is strongly encouraged to obtain full color versions of the figures.[1] Also, BITPICT now allows multiple layers of 2D planes – useful for scratch computation in general, and for handling sets of pictures in IDR. To illustrate the use of the multiple layers, we will restrict ourselves to one-dimensional, columnar IDR "diagrams", placed side-by-side in "layers". Throughout this exercise we will assume all diagrams have a given fixed dimension, and a rectangular tessellation, corresponding to pixels.

### 5.1 BITPICT Implementation of the Basic IDR Diagram Operators

The basic IDR diagram operators perform independent pixel-wise operations between two pixel diagrams and can be implemented nearly trivially with pixel rewrites between layers. The only difficulty is that pixel values in the current BITPICT definition are nominal. Since there is no preexisting machinery for ordinal operations on pixel values, like greater-than, less-than, max, and min, nor machinery for quantitative operations like addition and subtraction, these must all be implemented by hand, with an explicit rule for each possible combination of inputs. Thus a binary rule on $n$ gray levels would require $n^2$ rules.

The BITPICT implementation uses prefix notation -- when we want to use an operator to combine two diagrams, we will use three consecutive layers: Operator

[1] Animations of Figs. 8-13 are available in the University of Michigan DeepBlue archive, accessible via -- `http://hdl.handle.net/2027.42/65116`

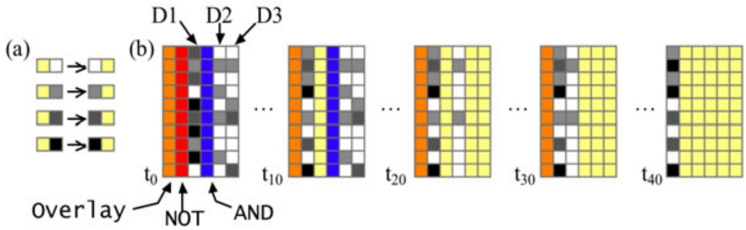



**Fig. 9.** Expression evaluation. Four rules (a) shift values left, trading places with empty space (light yellow). These, combined with the operator rules of Figure 8, yield the ability to evaluate arbitrary expressions. In (b) three one-dimensional "diagrams" (each 1x10 pixels) are combined according to the expression OVERLAY(NOT(D1), AND(D2, D3)). From time $t_0$ to $t_{10}$, the NOT is evaluated in place. From time $t_{10}$ to $t_{20}$, the AND is evaluated. From time $t_{20}$ to $t_{30}$ the results of the AND are shifted left. Finally, from $t_{30}$ to $t_{40}$, the OVERLAY is evaluated.

$\text{OperandDiagram}_1$ $\text{OperandDiagram}_2$. The operator layer is filled uniformly with a special "operator-name" color (e.g., red for NOT, green for OR, blue for AND, orange for OVERLAY, purple for PEEL). Fig. 8 illustrates operations on diagrams using the 4 rules for the NOT operation and the 16 rules each needed to define the AND and OVERLAY operations, for our four gray levels (white=0, light-gray=1, dark-gray=2, black=3="infinity").

The rules use a kind of reduction strategy. That is, the result overwrites the original operator-name layer, and leaves a special "empty" color (light yellow) in the place of

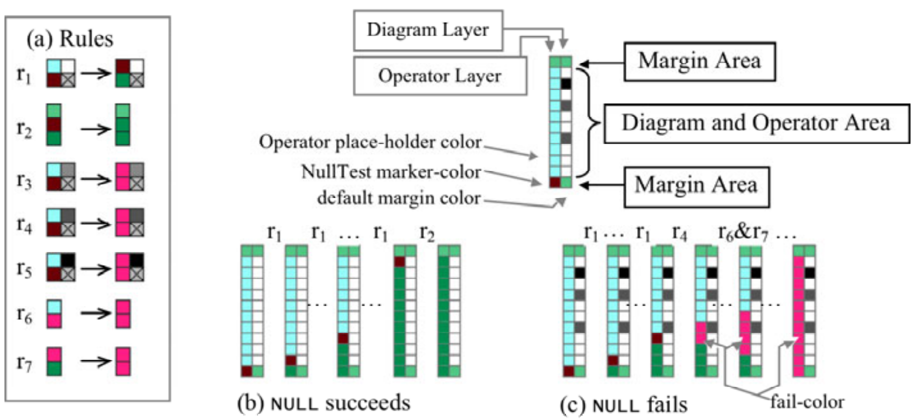



**Fig. 10.** The NULL predicate. In the seven rules in (a), the first column is an operator layer (column), the second if it exists, is a diagram layer. The first rule moves the marker-color (brown) upwards alongside white (zeros), leaving a trail of success (green) behind. If it reaches the top margin, the marker is colored green as well, and the whole layer now marked green signifies that the following diagram layer satisfies the predicate, as in (b). The next three rules detect non-zero at the leading edge of the progressing marker, seeding the "fail" color (pink). The last two, high priority, rules propagate the fail-color up and down, halting the predicate test and marking the following diagram as "failing to satisfy the predicate," as in (c).

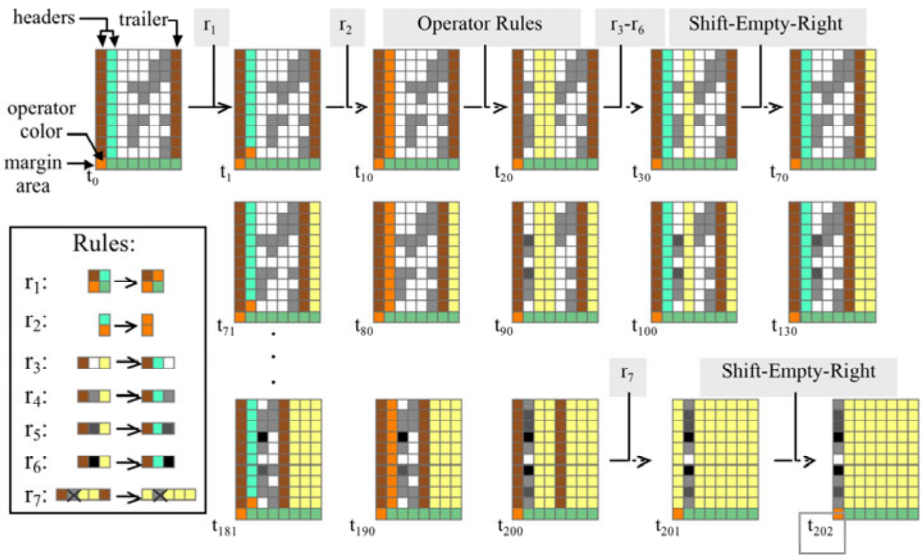



**Fig. 11.** The ACCUMULATE function. The situation at $t_0$ shows the two header columns, a column of *d*, five columns of *ds*, and the trailer column. The bottom row at each time is one pixel worth of marginal scratch area, used to hold the operator color for reseeding the second trailer column with the operator, per $r_1$ and $r_2$. Rules $r_3$ to $r_6$ have high priority and pre-empt the usual Shift-Empty-Right rules to restore the second header column. At $t_{200}$, rule $r_7$ matches to clear up the headers and trailer. (The x'ed out pixel of $r_7$ is a "don't-care", ignored in the match.)

the original operand layers. Other simple rules percolate these empty colors to the back end (right) of the workspace, bringing result values to the front (left) for further evaluation. Fig. 9 shows these percolation rules (a) and their dynamic operation in nested expressions (b).

The Boolean predicate, NULL, is more complicated – its output is not a diagram but a Boolean value, one which depends on every pixel in the diagram (all must be 0). To handle the non-diagrammatic Boolean results we make use of extra scratch space in the "margins" of the diagrams – in this case some pixels in rows above and below the "layers" (columns) of the one-dimensional "diagrams". The strategy for detecting all-zeros uses a layer, a light turquoise column, in front of the "diagram" to be tested, and begins with a special mark (a dark brown pixel) at the foot of that column. Pixel re-writes propagate that dark color one pixel up the column *iff* the adjacent diagram pixel is zero. As it rises, the mark leaves a trail of green (success) behind and only reaches the top if the whole diagram-column is zero. If the rising brown mark reaches a non-zero neighbor in the diagram column, a red (fail) color is seeded and with high priority propagates up and down the column. (Note that the NULL operator requires *n* rules for *n*-gray levels, to detect the failure conditions.)

### 5.2 BITPICT Implementation of the Basic IDR Set Operators

The functions ACCUMULATE, MAP and FILTER are even more complicated because they take as arguments other functions and arbitrary sized sets of diagrams. For

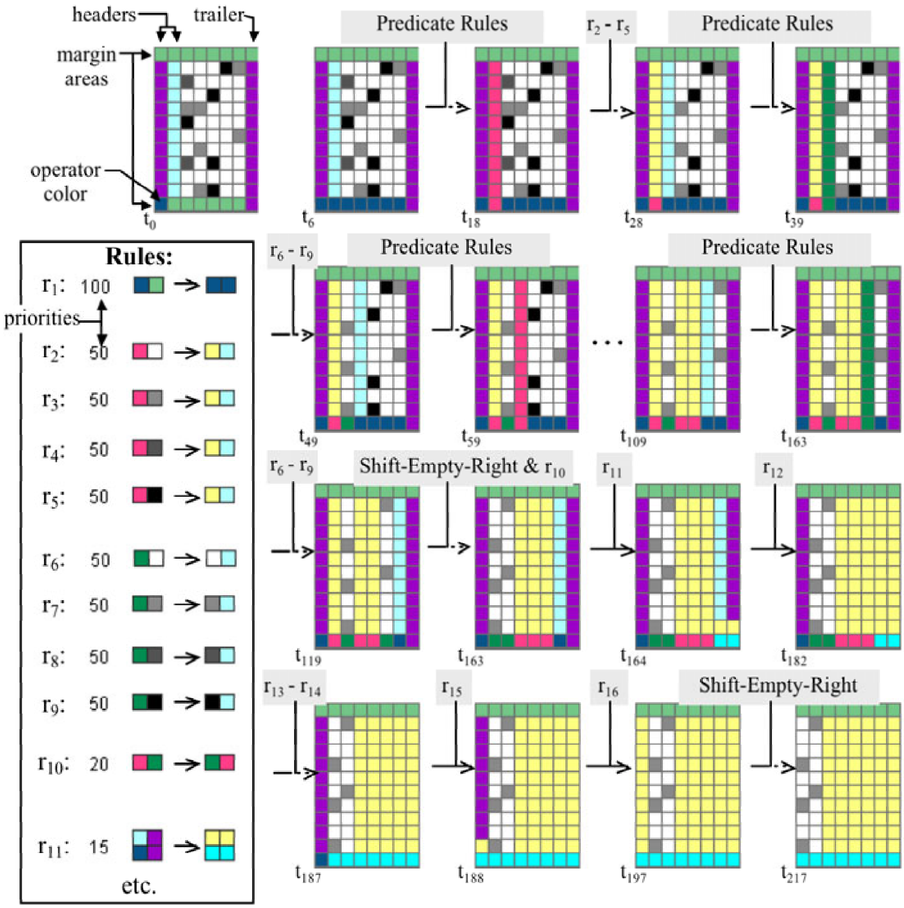



**Fig. 12.** The FILTER function applying a pre-defined LessThanTwo (LT2) predicate to a set of five diagrams. At $t_0$ there are Header and Trailer layers, with the second header layer containing only a OperatorPlaceHolder color. By $t_6$ the seed color of the LT2 operator has propagated from the lower left margin corner across the bottom of the whole set. Then the standard predicate handling rules (Fig. 10), propagate the operator mark up next to the first diagram layer and result in the layer marked with its success or, as in this first diagram, failure ($t_{18}$). At $t_{28}$ the failing diagram has been erased, and all is set up to evaluate the next diagram. This diagram satisfies LT2, is marked green at $t_{39}$, and $r_6$-$r_9$ preserve the diagram while moving the place-holder column forward. This evaluation process sweeps through the diagram layers leaving only satisfying diagrams. It stalls when it reaches the trailer, whereupon priority 20, shift-emptyspace-right rules (Fig. 9a) move the surviving diagrams left. Finally rule $r_{11}$ responds to the stalled sweep situation, seeding clean up processes like those in Fig. 11.

now we will assume that the functions being invoked have been predefined case by case in BITPICT rules (in the manner of Fig. 8). Fig. 12 shows a bitpict algorithm for ACCUMULATE, executing ACCUMULATE(*d*, *ds*, OVERLAY) for one 1x10 diagram, *d*, and a set, *ds*, of five 1x10 diagrams. To work with the arbitrary set, *ds*, of diagrams, we concatenate them all right behind *d*, bracketing them with two header columns at the front, and one trailer column at the end. The first and last columns are of a special ACCUMULATE color, with the front header column having an operator-color pixel below it in the margin indicating the operation which is to be accumulated. The second header column is a kind of regenerating operator column, as will be seen. In execution, first a rule copies the operator pixel color into the base of the second header column. Another rule, with high priority, propagates the operator color up the column. Then the usual operator execution rules evaluate this operator on the adjacent two data columns. Before the shift-empty-space-right rules get to fire, however, other high-priority rules fire and move the results column right and re-establish the operator placeholder column. Then the shift-empty-right rules fire and the remaining layers are compacted back to the left. At this point the setup is as at the beginning, except that the first pair of columns has been merged by the operator. This process then iterates, with the operator pixel recopying itself into the operator column, etc., until the last pair of columns has been merged. At this point a very high priority rule, $r_7$, recognizes the situation (matching when the header and trailer are only three apart with two empty-spaces at the end), and pre-empts the usual restoration of the operator column. Instead it replaces the header and trailer with empty-space. Then shift-empty-right rules move the result to its canonical, left-most position.

To implement the FILTER operator with a predefined monadic predicate, like NULL, we simply reset failed columns to empty-space (achievable with only a few rewrite rules). If we then also reset the predicate columns to empty-space, all the non-failed (null) columns will congregate to the left, and we end up with a set of all and only the diagrams for which the predicate succeeded. (See Fig. 12).

An implementation of a general MAP function (applying a predefined *n*-ary function) is possible using an auxiliary scratch area (see Fig. 13). The basic strategy concatenates the *n* sets of diagrams with dividers and uses a scratch area of comparable size below. The diagrams immediately following the dividers are peeled off into the scratch area, and collected. There the *n*-ary function operates and the result is copied back up to the main work area, and with a little cleanup, the process is ready to iterate on the remaining argument sets, until all is complete.

The set operations in ACCUMULATE, MAP and FILTER required effort to scope the sets being used, organize the sequential scanning through the diagram layers and clean up afterwards. The number of gray levels affected the FILTER mechanism but none of the others set operators.

Handling functions and predicates not explicitly defined by BITPICT rules presents a special challenge. IDR achieves this using the argument passing, lambda expressions, and defun capabilities of Lisp. Similar external machinery is possible for BITPICT (e.g., see [16]). Such machinery does not, however, exist per se in the primitives of pixel rewrites, any more than it does in the tape and table of a Turing machine. Consequently, its implementation in BITPICT is as much work as trying to

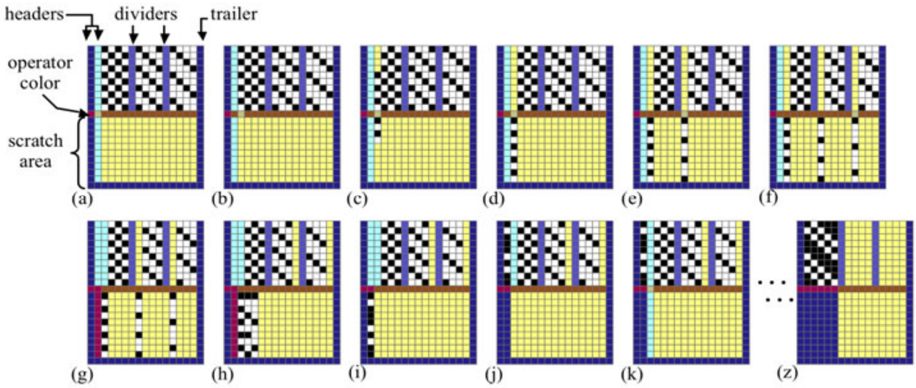



**Fig. 13.** The MAP function on a predefined 3-ary *max* function. (a) The three sets of diagram arguments are concatenate separated by dividers, above an equal sized scratch area of empty-space. (b) A small "particle" moves along the separating horizontal line until, reaching the first diagram after any divider, its progress is pre-empted by rules which drop the diagram column above the divider down, (c), one pixel at a time, into the space below, (d). The particle continues, copying the first diagram of each set into the scratch space, (e)&(f). Then the maroon operator-color is flooded into left end of the scratch space, and a turquoise OperatorPlace-Holder color is copied into the adjacent empty-space in the top section, (g). Shift-empty-space-right rules (Fig. 9a) collect the operands in the scratch area, and the diagram sets above, on the left, (h). The 3-ary *max* function operates the scratch area, (i), with shifted up to the original area, leaving a widened first header behind, (j). A few rules reset for the next cycle, (k), and the process iterates, evaluating the function on the remaining diagrams in each set, collecting the results in the front area of the original work area, (z).

bridge the comparable gap between a Turing machine (or some conventional machine language) and a Lisp compiler. The exercise would feel like a mix of abstract computer science and physical computer engineering. In an elaboration of the strategy in Fig. 13, encapsulation would be done by spatial separation and values would be passed by spatial movement. In a large scratch area outside the main workspace, the function definition would appear, basically as a sequence of layers of operators, constant diagrams, and dummy variable diagrams (each of a distinct special color). The diagrams in the main workspace that were to be passed as arguments would be connected by correspondingly colored wire-like paths to the corresponding dummy variable slots in the function definition. Algorithms we have for moving pixel patterns along arbitrary paths would move the copies of the input argument diagrams along the wire-like paths and plug them into the dummy argument slots of a scratch copy of the function, which would execute in place with results shunted back by a wire to the original calling location. All this would be done, for example, each time the MAP function tried to apply the defined function to a new diagram layer.

## 6 Discussion

As we performed this exercise, several commonalities between the systems became apparent.

Both systems operate on pixellated depictive representations, performing computations that respond to graphical structure in the image in complex ways to make inferences and solve problems. Along the way, both make use of intermediate graphical constructions [8] (e.g., the rule-masked-diagrams of Figs. 5 and 6 for IDR; the auxiliary workspace of Fig. 13 for BITPICT). We believe such graphical constructions to be likely in any depictive system. More specific to the architectures involved, we noted how easy it was for BITPCT to implement the basic nesting structure of Lisp's functional programming by simply using prefix notation. This is presumably because Lisp relies on a linear data structure – the list – whose 1D spatial localization organizes the computation. BITPICT simply exploited one of its spatial dimensions to do the same (though it had to push things from side to side to accomplish operations Lisp does more easily with its linked lists.) The larger lesson is that spatial structure itself can be exploited to organize computation and inference.

A variety of differences were also clear. Some simply arise because IDR works within Lisp's procedurally enhanced functional programming environment, while BITPICT uses a low-level rule-based production system architecture. Thus any looping execution in IDR would have to be achieved in BITPICT by capitalizing on the built-in match/rewrite loop of its underlying engine. Any conditional branching IDR might do in Lisp would, in BITPICT, have to be based on (graphically instantiated) logical state variables that its rewrites could later process, as would any other of Lisp's familiar high-level language capabilities.

Each system had certain built in graphical strengths, suggestive for the other. IDR was designed to make quantitative use of pixel values. BITPICT, with only nominal-scale values, required a combinatorial explosion of enumerated cases. Future enhancement of BITPICT to handle quantitative values explicitly could enable not just IDR-like functionality, but exploration of fuzzy-match and image processing capabilities. BITPICT's focus on nominal pixel values has, however, led it to explore the use of pixels to represent more complex types of spatially localized logical state for use in computation. Where IDR had built-in quantitative pixel values, BITPICT had built in geometric transformations that IDR had to explicitly enumerate. An IDR augmented with primitives for geometric transformations of diagrams could enable not just BIPTICT functionality, but enhance reasoning about problems with various built in symmetries.

IDR can be characterized as working *between* diagrams, with global operations independently processing corresponding pixels in sets of diagrams in ways that suggest an ease of parallelization. BITPICT, in contrast, is continually having some pixels affect their neighbors *within* a diagram, and uses a sequential execution and conflict resolution model that, while it allows highly optimized search and rewrite [6] with RETE-like processing not available to IDR, raises interesting challenges for parallelization. The distinction that IDR operates between diagrams where BITPICT operates within them is somewhat blurred, however, when one notes that BITPICT realizes IDR by making IDR's between-diagram operations become within-diagram operations, for higher dimensional diagrams. That is, a set of $k$-dimensional diagrams for IDR between-diagram operations becomes a $k+1$-dimensional diagram – the stack of IDR's diagrams – for internal operations in BITPICT, using $k+1$-dimensional rewrite rules.

The similarities and differences between IDR and BITPICT has led us to a deeper understanding of each, and pointed to directions for enhancing them, perhaps leading to convergence, or even to hybrid systems. As an important aspect of any field is the clear delineation of the work within it, it is our hope that this effort serves as encouragement to other efforts to consolidate related diagrammatic reasoning systems.

**Acknowledgements.** We wish to thank Dr. Robert Lindsay the University of Michigan his for insightful comments on an early draft of this paper and his continuing support of our work. Further, we wish to thank Richard Reilly of Fordham University for his help in delineating the differences between the two systems. This material is based upon work supported by the National Science Foundation under grant number IIS-9820368.

## References

1. Lindsay, R.K.: Knowing about diagrams. In: Anderson, M., McCartney, R. (eds.) Learning from Diagrams. Special Issue on Diagrammatic Reasoning of Journal of Machine Vision and Graphics, vol. 6(1) (1997)
2. Lindsay, R.K.: Using diagrams to understand geometry. Computational Intelligence 14, 222–256 (1998)
3. Jamnik, M.: Mathematical Reasoning with Diagrams. CSLI Publications, Stanford (2001)
4. Furnas, G.: Formal models for imaginal deduction. In: Proceedings of the Twelfth Annual Conference of the Cognitive Science Society, Cambridge, Mass., July 25-28, pp. 662–669. Lawrence Erlbaum, Hillsdale (1990)
5. Furnas, G.: Reasoning with Diagrams Only. In: Narayanan, N. (ed.) Proceedings of AAAI Spring Symposium on Reasoning with Diagrammatic Representations. AAAI Press, Menlo Park (1992)
6. Furnas, G., Qu, Y.: Shape Manipulation using Pixel Rewrites (at Visual Computation 2002). In: Proceedings of the Distributed Multimedia Systems 2002, San Francisco, CA, September 26-29, pp. 630–639 (2002)
7. Furnas, G., Qu, Y.: Using Pixel Rewrites for Shape-Rich Interaction. In: Proceedings of the Human Factors in Computing Systems CHI 2003 Conference, pp. 369–376. ACM, New York (2003)
8. Furnas, G., Qu, Y., Shrivastava, S., Peters, G.: The use of intermediate graphical constructions in problem solving with dynamic, pixel-level diagrams. In: Anderson, M., Cheng, P., Haarslev, V. (eds.) Diagrams 2000. LNCS (LNAI), vol. 1889, pp. 314–329. Springer, Heidelberg (2000)
9. Anderson, M., McCartney, R.: Inter-diagrammatic reasoning. In: Proc. IJCAI 1995, Montreal, Canada, pp. 878–884 (1995)
10. Anderson, M., McCartney, R.: Diagrammatic reasoning and cases. In: Proc. AAAI 1996, Portland, OR, pp. 1004–1009 (1996)
11. Anderson, M.: Toward diagram processing: A diagrammatic information system. In: Proc. AAAI 1999, Orlando, FL, pp. 394–401 (1999)
12. Anderson, M., McCartney, R.: Diagram processing: computing with diagrams. Artif. Intell. 145(1-2), 181–226 (2003)

13. El-Kafrawy, P.: Graphical Deadlock Avoidance. In: International Conference on Advances in Computing, Control, and Telecommunication Technologies (2009)
14. Lathrop, S.: Extending Cognitive Architectures with Spatial and Visual Imagery Mechanisms, Ph.D. Dissertation in Computer Science and Engineering, University of Michigan (2008)
15. Lathrop, S., Laird, J.: Extending Cognitive Architectures with Mental Imagery. In: Proceedings of the Second Conference on Artificial General Intelligence (AGI 2009) Arlington, Virginia, March 6-9 (2009)
16. Yamamoto, K.: Visulan: A Visual Programming Language for Self-Changing Bitmaps. In: Proc. of International Conference on Visual Information Systems, pp. 88–96. Victoria Univ. of Tech. cooperation with IEEE, Melbourne (1996)

# A Spatial Search Framework for Executing Perceptions and Actions in Diagrammatic Reasoning*

Bonny Banerjee and B. Chandrasekaran

Lab for AI Research, Department of Computer Science & Engineering
The Ohio State University, Columbus, OH 43210, USA
Banerjee.28@osu.edu, Chandra@cse.ohio-state.edu

**Abstract.** Diagrammatic reasoning (DR) requires perceiving information from a diagram and modifying/creating objects in a diagram according to problem solving needs. The perceptions and actions in most DR systems are hand-coded for the specific application. The absence of a general framework for executing perceptions/actions poses as a major hindrance to using them opportunistically. Our goal is to develop a framework for executing a wide variety of specified perceptions and actions across tasks/domains without human intervention. We observe that the domain/task-specific perceptions/actions can be transformed into domain/task-independent spatial problems. In our framework, a human problem solver specifies a spatial problem as a quantified constraint satisfaction problem (QCSP) in the real domain using an open-ended vocabulary of properties, relations and actions involving three types of spatial objects – points, curves, regions. Traditional approaches solve such QCSPs by computing the equivalent quantifier-free algebraic expression, the complexity of which is inherently doubly exponential. In this paper, we investigate a domain-independent framework of spatial search for solving $2D$ spatial problems specified as QCSPs. The framework searches for the solution in the space of the diagram instead of in the space of algebraic equations/inequalities. We prove the correctness of our approach and show that it is more efficient than cylindrical algebraic decomposition, a well-known algebraic approach, by executing perceptions/actions in two army applications.

## 1 Introduction

Reasoning with diagrams requires the use of symbolic knowledge representations and inferences as well as visually perceiving information from a diagram and creating/modifying spatial objects satisfying certain constraints. Complex problems are solved opportunistically as different subtasks are solved by the component – symbolic or visual – that is more suitable for it. The focus in this

* This research was partially supported by participation in the Advanced Decision Architectures Collaborative Technology Alliance sponsored by the U.S. Army Research Laboratory under Cooperative Agreement DAAD19-01-2-0009.

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 144–159, 2010.

paper is on the visual component that extracts (or perceives) information from a diagram and creates/modifies (or acts on) a diagram to satisfy given informational constraints. Accomplishing these perceptions and actions require solving domain-independent spatial problems. For example, an army commander, while planning strategic operations, might need to know the portions of a path prone to ambush (a.k.a. $RiskyPortionsofPath$). The visual component has to solve appropriate spatial problems – compute the set of points $q$ on a curve (the path) $c$ such that $q$ is within a specified distance (the firepower range) $d$ from some point $p$ and the line segment $\{p, q\}$ intersects a region $r$ (i.e., $p$ is behind $r$ with respect to $q$) – to deliver the required information. A schematic diagrammatic reasoning (DR) architecture from [1] is shown in Fig. 1.

In the last couple of decades, numerous DR systems have been built for different applications in different domains, such as, SKETCHY [2] for graph understanding in thermodynamics and economics, REDRAW [3] for structural analysis in civil engineering, ARCHIMEDES [4] for proving theorems in Euclidean geometry, DIAMOND [5] for proving certain mathematical theorems by induction, MAGI [6] for symmetry detection, and so on. All of these DR systems require perceiving from and/or acting on diagrams, where each perception/action requires solving a domain-independent spatial problem. How were these spatial problems solved in the DR systems?

Typically, the human developing a DR system identifies a priori the problem solving steps including a set of perceptions and actions, and hand-codes algorithms for solving each of them. If the problem solving steps need to be altered in future and as a result, a new perception arises, the developer has to write a new algorithm for solving it. Clearly, this is inconvenient and does not allow fast and easy experimentation with different problem solving strategies for the same problem. These drawbacks are further magnified when the goal is to build a general-purpose DR system where a very large variety of perceptions and actions are possible which is not feasible to ascertain a priori, and develop and store algorithms for. Hence, our goal is to investigate:

1. A language for a human to specify a variety of spatial problems, and
2. A general domain-independent framework for efficiently solving spatial problems, specified in the above language, without human intervention and presenting the solution in visual form whenever possible.

**Definition 1** ***Diagram.*** *(from [1]) A diagram $\mathcal{D}$ is a set of labeled $2D$ spatial objects $\{\mathcal{O}_1, \mathcal{O}_2, ...\mathcal{O}_l\}$. A spatial object $\mathcal{O}$ is a 3-tuple $< \mathcal{L}, \mathcal{T}, \mathcal{E} >$ where $\mathcal{L}$ is a label, $\mathcal{T}$ is a type (point, curve or region), and $\mathcal{E}$ is its spatial extent. The spatial extent of an object is the set of points constituent of the object. Further, diagrammatic image, $\mathcal{I}(\mathcal{D})$, is the set of points constituent of all objects in $\mathcal{D}$.*

We are interested only in monochromatic diagrams with no intensity variation. A perception is a mapping from a diagram $\mathcal{D}$ to a set of booleans $\{True, False\}$ or real numbers $\Re$. An action is a mapping from $\mathcal{D}$ to a new set of spatial objects or diagram $\mathcal{D}'$, where $\mathcal{I}(\mathcal{D}) \neq \mathcal{I}(\mathcal{D}')$.

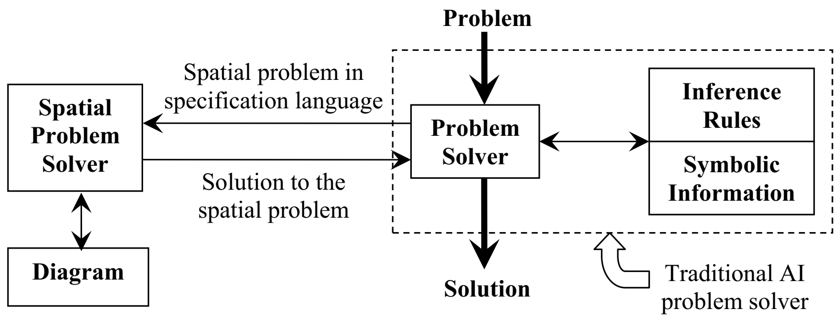



**Fig. 1.** A schematic diagrammatic reasoning architecture

In the next section, we briefly visit the specification language (borrowed from our earlier work) and discuss the challenges that arise in solving spatial problems from such specification. We have identified a small vocabulary of predicates which are hand-coded internally as procedures (as opposed to declarative sentences). A spatial problem is specified in the specification language using a fixed set of operators and the predicates in the vocabulary. In section 3, a general domain-independent framework of spatial search – the main focus of this paper – is described along with its correctness analysis. In this framework, unlike algebraic approaches, the solution is searched for in the space of the diagram instead of in the space of algebraic equations/inequalities. Enhancements are proposed to improve complexity. The enhanced algorithm is shown to be more efficient than cylindrical algebraic decomposition (CAD), a well-known algebraic approach, in executing perceptions/actions for two army applications.

## 2 Specification Language

The specification language [7] is a high-level language that is extensible, human-usable, and expressive enough to describe a wide variety of $2D$ spatial problems. The problems will be accepted as input by the spatial problem solver (SPS) and solved without human intervention. The language recognizes boolean operators $\{\wedge, \vee, \neg\}$, arithmetic operators $\{+, -, \times, \div\}$, relational operators $\{<, >, =, \neq\}$, and quantifiers $\{\exists, \forall\}$. The brackets () are used to express precedence while the brackets {} are used for a set. The domain is real plane, $\Re^2$.

### 2.1 Vocabulary

From the literature [1,7,8,9,10], we have extracted a minimal vocabulary of properties, relations and actions based on their wide usage in expressing spatial problems in different domains. The vocabulary is not claimed to be complete and addition of new properties/relations/actions is allowed when a problem cannot be easily expressed using the existing ones. Currently, the properties, relations and actions in our vocabulary are as follows.

***Properties.*** Associated with each type of object are a few properties – location $\mathcal{E}(p)$ of a point $p$, locations $\mathcal{E}_x(p)$ and $\mathcal{E}_y(p)$ of $p$ along x- and y-axis respectively; spatial extent $\mathcal{E}(c)$ and length $Length(c)$ of a curve $c$; and spatial extent $\mathcal{E}(r)$, area $Area(r)$ and periphery $Periphery(r)$ of a region $r$, where the periphery of a region refers to its boundary curve.

***Relations.*** The vocabulary contains the relations: $Distance(p_1, p_2)$, $Angle(p_1, p_2, p_3)$, $Leftof(p_1, p_2)$, $Rightof(p_1, p_2)$, $Above(p_1, p_2)$, $Below(p_1, p_2)$, $On(p, c)$ and $Inside(p, r)$, where $p$, $p_1$, $p_2$, $p_3$ are points, $c$ a curve and $r$ a region.

***Actions.*** There are four actions for modifying objects. $Translate(\mathcal{O}, t_x, t_y)$ returns a translation of object $\mathcal{O}$ for $t_x$ units along $x$-axis and $t_y$ units along $y$-axis, $Rotate(\mathcal{O}, c, \theta)$ returns a rotation of the object $\mathcal{O}$ with respect to point $c$ as center for $\theta$ degrees in the anti-clockwise direction, $Reflect(\mathcal{O}, \{a, b\})$ returns a reflection of object $\mathcal{O}$ with respect to the line segment $\{a, b\}$, $Scale(\mathcal{O}, c, s_x, s_y)$ returns a scaling of the object $\mathcal{O}$ with respect to point $c$ for $s_x$ units along $x$-axis and $s_y$ units along $y$-axis.

### 2.2 The Language

We will use a functional constraint logic programming language (first-order predicate logic) as the specification language [7].

***Quantified Constraint Satisfaction Problem.*** An instance of a constraint satisfaction problem (CSP) consists of a tuple $< \mathcal{V}, D, \mathcal{C} >$ where $\mathcal{V}$ is a finite set of variables, $D$ is a domain, and $\mathcal{C}= \{\mathcal{C}_1, ...\mathcal{C}_k\}$ is a set of constraints. A constraint $\mathcal{C}_i$ consists of a pair $< \mathcal{S}_i, \mathcal{R}_i >$ where $\mathcal{S}_i$ is a list of $m_i$ variables and $\mathcal{R}_i$ is a $m_i$-ary relation over the domain $D$. The question is to decide whether or not there is an assignment mapping each variable to a domain element such that all the constraints are satisfied. All of the variables in a CSP can be thought of as being implicitly existentially quantified.

A useful generalization of the CSP is the quantified constraint satisfaction problem (QCSP), where variables may be both existentially and universally quantified. An instance of the QCSP consists of a quantified formula in first-order logic, which consists of an ordered list of variables with associated quantifiers along with a set of constraints. A QCSP can be expressed as follows:

$$\phi(v_1, ...v_m) \equiv Q(x_n, ...x_1)\phi'(v_1, ...v_m, x_1, ...x_n)$$
$$Q(x_n, ...x_1) \equiv Q_n x_n, ...Q_1 x_1$$

where $Q_i \in \{\forall, \exists\}$, $\{x_1, ...x_n\}$ is the set of quantified variables, $\{v_1, ...v_m\}$ is the set of free variables, $\mathcal{V}= \{v_1, ...v_m, x_1, ...x_n\}$, and $\phi'$ is a quantifier-free expression called the matrix. Such representation of a quantified expression $\phi$, where it is written as a sequence of quantifiers followed by the matrix, is referred to as prenex form.

**Definition 2** ***Spatial Problem.*** *A spatial problem is a QCSP where a variable (quantified or free) can only be of type point, and the domain is* $\Re^2$.

Thus, a spatial problem $\phi$ is a mapping from a diagram $\mathcal{D}$ satisfying a logical combination of constraints $\mathcal{C}$ to a set of booleans $\{True, False\}$ or real numbers $\Re$ or spatial objects $\mathcal{D}'$, i.e.,

$$\phi : \mathcal{D} \xrightarrow{\mathcal{C}} \{True, False\} \cup \Re \cup \mathcal{D}'$$

Solving a spatial problem requires solving a QCSP. Algebraic approaches to solving a QCSP eliminate quantifiers and solve algebraic equations/inequalities to arrive at the most simplified expression. The computational bottleneck is real quantifier elimination (QE) which is inherently doubly exponential in time complexity even when there is only one free variable and all polynomials in the quantified input are linear [11]. The most general and elaborately implemented method for real QE is CAD [12], complexity of which is $(sd)^{O(1)^{k-1}}$ where $s$ is the number of polynomials, their maximum degree is $d$ and coefficients are real, and $k$ is the number of blocks of quantified variables. For large real-world problems, it soon becomes too time consuming to solve a QCSP.

## 3 Spatial Problem Solver (SPS)

We propose a general framework of spatial search for efficiently solving, without human intervention, a wide variety of spatial problems expressed as QCSPs and presenting the solution in visual form. The solution is searched for in the space of the diagram instead of in the space of algebraic equations/inequalities.

***Decision and Function problems.*** In the specification language, a spatial problem $\phi$ is expressed as a QCSP where $\mathcal{V}$ consists of variables of type point, $D = \Re^2$, and $\mathcal{C}$ is the set of constraints to be staisfied. Solving a spatial problem involves:

1. When there are no free variables in $\mathcal{V}$, deciding whether there exists a mapping from $\mathcal{V}$ to $D$ satisfying $\mathcal{C}$.
2. When there are free variables in $\mathcal{V}$, computing the mapping from $\mathcal{V}$ to $D$ satisfying $\mathcal{C}$.

The first case constitutes a decision problem that yields a $True/False$ solution. The second case constitutes a function problem that yields a spatial object described by the free variables as the solution. For example, given a curve $c$ and two points $p$, $q$, the spatial problem $BehindCurve(q, c, p)$ is defined as deciding whether or not $q$ is behind $c$ with respect to $p$. This might be specified as deciding whether or not the curve $c$ and line segment $\{p, q\}$ intersect. Thus,

$$BehindCurve(q, c, p) \equiv Intersect(c, \{p, q\})$$

For particular instances of $q$, $p$, $c$, the solution to this problem is $True/False$ – a decision problem. For particular instances of $p$, $c$, and generalized coordinates of $q$ i.e., $q \leftarrow (x, y)$, the solution to the same problem is a region object – a function problem. While a decision problem merely requires checking whether or not a given instance of an object satisfies the constraints or not, a function problem requires computing all instances that satisfy the constraints.

Since there are only point variables, the solution to a function problem can be obtained by solving the corresponding decision problem for every point in the diagram. Only the points for which the decision problem evaluates to $True$ are included in the solution set. Thus, theoretically, if a decision problem can be solved, function problems involving that decision problem can also be solved.

Since the number of points that constitute any diagram is infinite, checking whether or not a decision problem outputs $True$ for each one of them cannot be accomplished in finite time. However, if the space in a diagram is discretized into a finite number of pixel-like elements or *pels* where each pel corresponds to a point, the task can be accomplished in finite time. This reduction in time complexity comes at the cost of precision. The precision of the solution will be dependent on the maximum resolution of the diagram.[1] However, if the resolution can be varied in a problem-dependent manner such that only relevant parts of the diagram are viewed at high resolution, a significant amount of computation time can be saved without compromising precision significantly.

### 3.1 Underlying Representation and Operators

***Pel-object data structure.*** The space in a diagram is discretized by imposing an array of square pels at the image resolution. Each pel corresponds to a point and is indexed and maintains a list of all spatial objects that it is occupied by. Each object maintains a list of all pels that it occupies. This can be conceptualized as a $2D$ table with the two dimensions corresponding to the objects and the pels, and each entry in the table containing a 1 or 0 depending on whether that pel and object belong to each other or not. Thus, given a pel and an object, their relation can be retrieved in constant time. Size of this table is $N \times l$ where $N$ is the number of pels at image resolution and $l$ the number of objects.

***Implementation of predicates.*** $\mathcal{E}(p)$ returns the x- and y-indices of the pel occupied by point $p$. $\mathcal{E}(c)$ returns the sequence of indices of pels occupied by curve $c$ from one end to another. $Length(c)$ returns the number of pels in $\mathcal{E}(c)$. $\mathcal{E}(r)$ returns indices of all pels occupied by region $r$. $Area(r)$ returns the number of pels in $\mathcal{E}(r)$. $Periphery(r)$ returns the sequence of indices of all pels occupied by the boundary curve of $r$ starting from some pel on the boundary. $Distance(p_1, p_2)$ returns the Euclidean distance between $\mathcal{E}(p_1)$ and $\mathcal{E}(p_2)$. $Angle(p_1, p_2, p_3)$ returns the angle at $\mathcal{E}(p_2)$ from $\mathcal{E}(p_1)$ and to $\mathcal{E}(p_3)$. $Leftof(p_1, p_2)$ returns $True$ if $\mathcal{E}_x(p_1) \leq \mathcal{E}_x(p_2)$, otherwise $False$. $Rightof(p_1, p_2)$ returns $True$ if $\mathcal{E}_x(p_1) \geq \mathcal{E}_x(p_2)$, otherwise $False$. $Above(p_1, p_2)$ returns $True$ if $\mathcal{E}_y(p_1) \geq \mathcal{E}_y(p_2)$, otherwise $False$. $Below(p_1, p_2)$ returns $True$ if $\mathcal{E}_y(p_1) \leq \mathcal{E}_y(p_2)$, otherwise $False$. $On(p, c)$ returns $True$ if the data structure contains 1 for $\mathcal{E}(p)$ and $c$, otherwise $False$. $Inside(p, r)$ returns $True$ if the data structure contains 1 for $\mathcal{E}(p)$ and $r$, otherwise $False$. $Translate(\mathcal{O}, t_x, t_y)$ returns the set of pels in $\mathcal{E}(\mathcal{O})$ after adding $t_x$ and $t_y$ to the x- and y-indices of each pel. $Rotate(\mathcal{O}, p, \theta)$

[1] This is true for algebraic approaches as well where the complexity depends on $s$ and $d$ (see section 2.2). To keep $d$ low, spatial objects are approximated piecewise-linearly. To keep $s$ low, objects cannot be approximated too closely, thereby losing precision.

returns the set of pels in $\mathcal{E}(\mathcal{O})$ after rotating each pel's indices with respect to $p$ for $\theta$ degrees in anticlockwise direction. $Reflect(\mathcal{O}, \{a, b\})$ returns the set of pels in $\mathcal{E}(\mathcal{O})$ after reflecting each pel's indices with respect to line segment $\{a, b\}$. $Scale(\mathcal{O}, p, s_x, s_y)$ returns the set of pels in $\mathcal{E}(\mathcal{O})$ after scaling each pel's indices with respect to $p$ for $s_x$ and $s_y$ units along x- and $y$-axis. Each time a new object is introduced/modified/deleted, the data structure is updated.

***The three spatial search operators.*** Whether a spatial problem is decision or function is interpreted from the problem specification by the SPS – if the specification contains free variables, it is a function problem, else a decision problem. The SPS also knows how to search when the different kinds of quantifiers occur – using three operators $\{F_\exists, F_\forall, F\}$.

In a decision problem, when an existential quantifier occurs in the specification, i.e. $\phi \equiv \exists x, \phi'(x)$, the SPS searches all pels in the diagram until it finds one that evaluates $\phi'$ to $True$, when it halts and outputs $True$. If no such pel is found, the SPS outputs $False$. The pels being searched correspond to the existentially quantified point variable. Given a diagram $D$, a quantifier free expression $\phi'(x)$ such that $\phi \equiv \exists x, \phi'(x)$, and the existentially quantified variable $x$, the operator $F_\exists$ computes the solution to $\phi$, as follows:

$F_\exists$ : 1. for $i \leftarrow$ each pel in $D$
2. if $\phi'(i) = True$,
3. return $True$;
4. return $False$;

The application of $F_\exists$ is written as $F_\exists \; o \; (D, \phi'(x), \{\}, \exists x)$ where the empty set $\{\}$ indicates no free variables. A function problem requires two searches – for each pel in the diagram, the SPS needs to solve the decision problem. The pels for which the decision problem evaluates to $True$ constitute the solution. The operator $F$ collects all those pels from a diagram that satisfy the decision problem. If $\phi(v) \equiv \exists x, \phi'(v, x)$ is a function problem and $v$ is the free variable, $F$ computes the solution to $\phi$, as follows:

$F$ : 1. $S \leftarrow \{\}$;
2. for $i \leftarrow$ each pel in $D$
3. if $F_\exists \; o \; (D, \phi'(i, x), \{\}, \exists x) = True$,
4. $S \leftarrow S \cup \{i\}$;
5. return $S$;

This is written as $F \; o \; F_\exists \; o \; (D, \phi'(v, x), \{v\}, \exists x)$.

Similarly, for a decision problem with a universal quantifier, i.e. $\phi \equiv \forall x, \phi'(x)$, $F_\forall$ computes the solution to $\phi$ as follows:

$F_\forall$ : 1. for $i \leftarrow$ each pel in $D$
2. if $\phi'(i) = False$,
3. return $False$;
4. return $True$;

This is written as $F_\forall \; o \; (D, \phi'(x), \{\}, \forall x)$.

When there is more than one quantified variable in a decision problem $\phi \equiv Q(x_n, ...x_1)\phi'(x_1, ...x_n)$, it is first expressed in prenex form to extract $\phi'(x_1, ...x_n)$. Then the problem is solved by the successive application of the operators, $F_\exists$ or $F_\forall$, corresponding to the quantifiers, $\exists$ or $\forall$. For example, a decision problem $\phi \equiv \exists x_2 \forall x_1, \phi'(x_1, x_2)$ is solved as $F_\exists \ o \ F_\forall \ o \ (D, \phi'(x_1, x_2), \{\}, \exists x_2 \forall x_1)$. The corresponding function problem is solved as $F \ o \ F_\exists \ o \ F_\forall \ o \ (D, \phi'(v, x_1, x_2), \{v\}, \exists x_2 \forall x_1)$.

In a $2D$ diagram, there can be at most one free variable in any problem, but a number of quantified variables in arbitrary order. In general, a decision problem is of the form $\phi \equiv Q_n x_n, ...Q_1 x_1 \phi'(x_1 ...x_n)$, and is solved as:

$$F_{Q_n} \ o \ ... \ F_{Q_1} \ o \ (D, \phi'(x_1, ...x_{n-1}, x_n), \{\}, Q(x_n, ...x_1))$$

Similarly, a function problem in DR is in general of the form $\phi(v) \equiv Q_n x_n, ...Q_1 x_1$ $\phi'(v, x_1 ...x_n)$ where $v$ is the free variable, and is solved as:

$$F \ o \ F_{Q_n} \ o \ ... \ F_{Q_1} \ o \ (D, \phi'(v, x_1, ...x_n), \{v\}, Q(x_n, ...x_1))$$

The three operators automate the process of spatial problem solving. The time complexity of this naive approach is $O(N^k)$ where $N$ is the number of pels in the diagram and $k$ is the total number of free and quantified variables.

### 3.2 Correctness Analysis

By induction, we will show that the decision problem $\phi \equiv Q(x_n, ...x_1)\phi'(x_1, ...x_n)$, where $Q(x_n, ...x_1) \equiv Q_n x_n ...Q_1 x_1$, $Q_i \in \{\exists, \forall\}$, can be solved by the application of the two operators, $F_\exists$ and $F_\forall$, in the following sequence:

$$F_{Q_n} \ o \ ... \ F_{Q_1} \ o \ (D, \phi'(x_1, ...x_n), \{\}, Q(x_n, ...x_1))$$

Using that, we will show that the function problem $\phi \equiv Q(x_n, ...x_1)\phi'(v, x_1, ...x_n)$, where $v$ is a free variable, can be solved by the application of the three operators, $F$, $F_\exists$ and $F_\forall$, in the following sequence:

$$F \ o \ F_{Q_n} \ o \ ... \ F_{Q_1} \ o \ (D, \phi'(v, x_1, ...x_n), \{v\}, Q(x_n, ...x_1))$$

For $n = 1$, the decision problem $\phi \equiv \exists x_1, \phi'(x_1)$ or $\phi \equiv \forall x_1, \phi'(x_1)$. We have shown that these problems can be solved by $F_\exists \ o \ (D, \phi'(x_1), \{\}, \exists x_1)$ and $F_\forall \ o \ (D, \phi'(x_1), \{\}, \forall x_1)$ respectively.

For $n = 2$, the decision problem $\phi$ can have four cases: $\phi \equiv \exists x_2 \exists x_1, \phi'(x_1, x_2)$, $\phi \equiv \forall x_2 \exists x_1, \phi'(x_1, x_2)$, etc. The first case is solved by $F_\exists \ o \ F_\exists \ o \ (D, \phi'(x_1, x_2), \{\}, \exists x_2 \exists x_1)$ and similarly for the other three cases.

Let us assume that the proposition holds for $n = m$. Therefore, the decision problem $\phi \equiv Q(x_m, ...x_1)\phi'(x_1, ...x_m)$ can be solved as:

$$F_{Q_m} \ o \ ... \ F_{Q_1} \ o \ (D, \phi'(x_1, ...x_m), \{\}, Q(x_m, ...x_1))$$

Consider the proposition for $n = m + 1$. The decision problem

$\phi \equiv Q(x_{m+1}, ...x_1)\phi'(x_1, ...x_{m+1})$ can have two cases:
$\phi \equiv \exists x_{m+1} Q(x_m, ...x_1)\phi'(x_1, ...x_{m+1})$ or $\phi \equiv \forall x_{m+1} Q(x_m, ...x_1)\phi'(x_1, ...x_{m+1})$.

The first case is solved as $F_\exists \ o \ F_{Q_m} \ o \ ... \ F_{Q_1} \ o \ (D, \phi'(x_1, ...x_{m+1}), \{\}, \exists x_{m+1} Q(x_m, ...x_1))$. Similarly for the other case. Thus, given that the proposition holds for $n = m$, it also holds for $n = m + 1$. But it holds for $n = 1, 2$. Hence it holds for $n = 3, 4, ...$. The proof follows.

Trivially, a function problem $\phi(v) \equiv Q(x_n, ...x_1)\phi'(v, x_1, ...x_n)$ is solved as $F \ o \ F_{Q_n} \ o \ ... \ F_{Q_1} \ o \ (D, \phi'(v, x_1, ...x_n), \{v\}, Q(x_n, ...x_1))$. Thus, the proposition holds for function problems as well.

### 3.3 Enhancing the Efficiency of Spatial Search

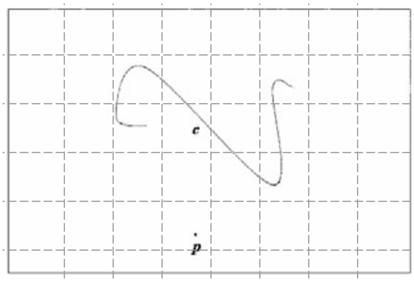
It is impressive, to say the least, how the human visual system executes spatial search so efficiently for such a wide variety of spatial problems. We believe, in addition to parallel computation, a problem guides the visual system to facilitate search by restricting computation to relevant parts of the space. For example, when computing the region behind a curve $c$ with respect to a point $p$ (see Fig. 2), certain areas require more computation than others due to constraints in the problem and not merely due to the configuration of objects in the diagram. In this problem, the space that contain the boundaries of the behind region receives more computation time. The rest of the space, either clearly behind or clearly not, receives less time. In general, a vast majority of time is restricted to computing the precise boundary of the solution.

To implement this strategy, we maintain the pel-object data structure at multiple resolutions. Let $\mathcal{P}^{(i)}$ denote a pel at the $i^{th}$ resolution. Let there be $s$ resolutions, $i = 0$ being the highest resolution, $d \times d$ be the number of $\mathcal{P}^{(i)}$s in one $\mathcal{P}^{(i+1)}$ (i.e. factor of change in resolution). Then, $\mathcal{P}^{(i)}$ contains $d^i \times d^i$ $\mathcal{P}^{(0)}$s. Each $\mathcal{P}^{(i)}$ is indexed by the indices of the four $\mathcal{P}^{(0)}$s at its corners.

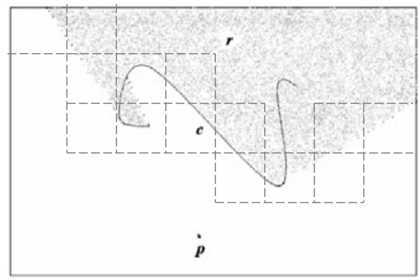
Given a spatial problem, the diagram is processed starting from the lowest resolution. $\mathcal{P}^{(i)}$ is said to satisfy constraints $\mathcal{C}$ when all four of its corner $\mathcal{P}^{(0)}$s satisfy $\mathcal{C}$. $\mathcal{P}^{(i)}$ is said to not satisfy $\mathcal{C}$ when all four of its corner $\mathcal{P}^{(0)}$s do not satisfy $\mathcal{C}$. $\mathcal{P}^{(i)}$ is said to partially satisfy $\mathcal{C}$ when at least one of its corner $\mathcal{P}^{(0)}$s satisfies $\mathcal{C}$ and at least one does not. Thus, $\mathcal{P}^{(i)}$, when checked for satisfaction of constraints, belongs to one of three classes – satisfies, does not satisfy, or partially satisfies. Pels belonging to the last class are further investigated at the next higher resolution and checked for satisfaction of constraints thereby classifying them into three classes. This procedure continues until the maximum resolution is reached. It is trivial to see that, in this implementation, computation time is restricted to space in the diagram that requires it and not wasted by processing the entire diagram uniformly.[2] See Fig. 2 for example.

***Computational complexity.*** Since the amount of space explored is problem-dependent, it is difficult to ascertain the absolute reduction in computational costs. However, an average case analysis will provide an idea of the efficiency

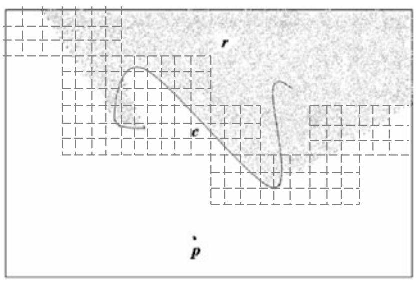
[2] A data-structure called *quadtree* is used to sample space hierarchically using pels of varying sizes based on the density of objects – higher the density, smaller the pel. Efficient spatial search requires a problem-dependent exploration of space as implemented by our strategy and not by quadtree.



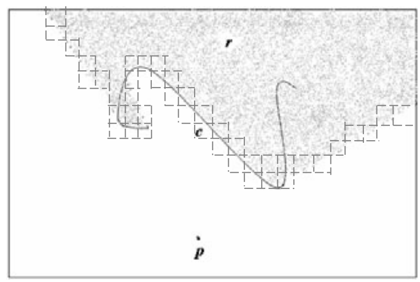

(a) An array is imposed on the diagram.




(b) The partially behind array elements.




(c) Subdividing partially behind array elements into smaller elements.




(d) The partially behind array elements at the higher resolution.

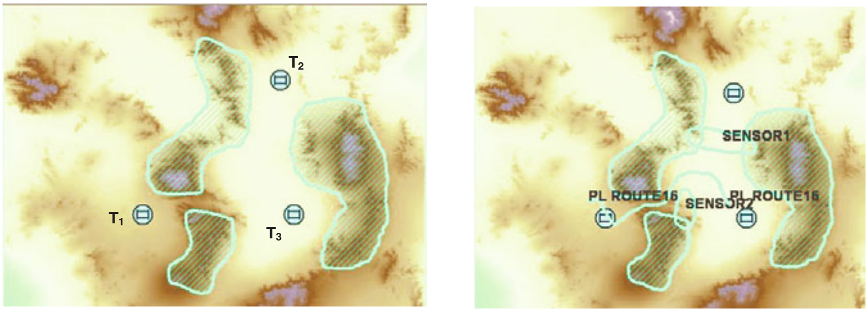
**Fig. 2.** Solving the $BehindCurve(p, c)$ problem by spatial search

achieved. Two kinds of computational costs are involved – a one time cost of filling in the multi-resolution data structure and a processing cost incurred due to checking every pel for satisfaction of the decision problem. The goal in this analysis is to count the total number of pels updated and processed by the naive spatial search strategy and compare it with the same for the resolution-based enhanced strategy. Let $n$ be the number of pels at the lowest resolution, $N$ be the number of pels at the highest resolution, $m$ be the average fraction of pels that require further investigation at any step, and $k$ be the total number of variables in the specification of the spatial problem.

Therefore, $N = nd^{2s}$. The multi-resolution data structure contains a total of $O(Nl)$ pels. In the naive spatial search, the SPS processes $N^k$ pels. In the multi-resolution approach, at the $i^{th}$ resolution, SPS processes $n^k(md^2)^{ik}$ pels. Thus, the total number of pels processed in $s$ resolutions is $N^k m^{sk}$ – a reduction by $m^{sk}$ $(m \leq 1, s, k \geq 1)$ compared to the naive approach. For example, in the $BehindCurve$ problem (Fig. 2), using parameters $m \leftarrow 0.5$, $d \leftarrow 3$, $s \leftarrow 4$, $n \leftarrow 12 \times 12$, $N \leftarrow 1024 \times 1024$, $k \leftarrow 2$, processing time reduces by 99.61%.

## 4 Experimental Results

In this section, we illustrate how a human problem solver can exploit the SPS to execute perceptions and actions without human intervention as needed for




(a) Terrain, impassable regions, and sighted tanks.

(b) A path from the only plausible homotopy class.

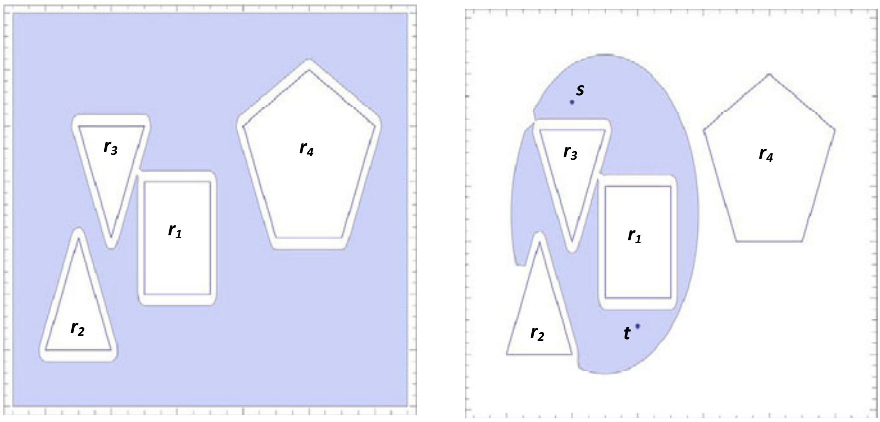
**Fig. 3.** A simplified entity re-identification scenario

DR. Two applications will be considered – entity re-identification and ambush analysis – that are deemed very important in the military domain. Problems in military domain involve a wide variety of objects with arbitrary properties and relations, and hence, help to illustrate the expressiveness of the specification language and the efficiency and generality of the spatial search strategy. Comparison of computation time between our SPS and the CAD algorithm [12] is provided for an instance of each spatial problem.

### 4.1 Entity Re-identification

The entity re-identification problem is a core task in the US Army's All-Source Analysis System (ASAS). ASAS receives a new report about sighting of an entity $T_3$ of type $T$ (e.g. tanks). The task is to decide if the new sighting is the same as any of the entities in its database of earlier sightings, or an entirely new entity. Reasoning has to dynamically integrate information from different sources – database of sightings, mobility of vehicles, sensor reports, terrain and map information – to make the decision.

Fig. 3(a) shows the terrain of interest – mountainous with the closed regions marking impassable areas for entities of type $T$ (e.g., tanks). Let $T_3$ be an entity newly sighted at time $t_3$ located at point $p_3$ while $T_1$, $T_2$ are the two entities that were located at points $p_1$, $p_2$ when last sighted at times $t_1$, $t_2$ respectively. $T_1$ and $T_2$ were retrieved from the database as having the potential to be $T_3$ based on their partial identity information. Also, in the area of interest, there are three enemy regions or obstacles $\{r_1, r_2, r_3\}$ with a given firepower/sight range $d$ of the enemy. Reasoning proceeds as follows. If $T_1$ can reach $p_3$ within the time $t_3 - t_1$, then $T_3$ might be $T_1$. Similarly for $T_2$. Since each mountainous region (or obstacle) is a hiding place for enemies with a firepower range $d$, the existence of an entity shows that it most probably did not traverse through a territory within the firepower range. Further, there might be sensor fields that report to the database when they sense entities. If no entity was sensed between the times $t_1$ and $t_3$, then $T_1$ could not have followed a path that passed through




(a) The unshaded polygons are obstacles. The shaded region is the safe region.

(b) Paths lying in the safe region and less than a given length between two points.

**Fig. 4.** A simplified scenario to illustrate the performance of the proposed SPS

that sensor field. Such constraints have to be taken into account while reasoning. All information might not be available in the database at once. In what follows is a simple scenario and a discussion of the spatial problems as they occur.

The problem solver (e.g., a commander) wants to know whether there exists a contiguous safe region containing the points $p_1$ and $p_3$, and specifies:

$$SafeRegion(q, \{r_1, ...r_n\}, d) \equiv \forall a, \neg(\vee_{i\leftarrow 1}^{n} Inside(a, r_i)) \vee Distance(q, a) \geq d$$

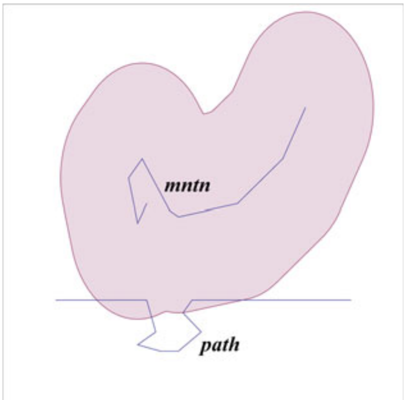
where $q \leftarrow (x, y)$. In order to compare the actual computation times, we constructed a very simple diagram consisting of four polygonal regions depicting obstacles (see Fig. 4(a)), where $r_1 \leftarrow \{(10, 10), (30, 10), (30, 30), (10, 30)\}$, $r_2 \leftarrow \{(-20, 0), (0, 0), (-10, 20)\}$, $r_3 \leftarrow \{(0, 20), (10, 40), (-10, 40)\}$, $r_4 \leftarrow \{(50, 20), (70, 20), (80, 40), (60, 50), (40, 40)\}$, and $d \leftarrow 2$. See Table 1.[3]

Next the problem solver wants to know whether there exists a path between points $p_1$ and $p_3$ safely avoiding the obstacles and enemy firepower range, and whether that path can be traversed in time $t_3 - t_1$. Let $v$ be the velocity of the sighted entity – a piece of symbolic knowledge available from the database. Then, the maximum length of path traversable in the given time is $L = v \times (t_3 - t_1)$. Let $l \ll L$ be a rational number. Then, the problem of path existence between two points $s$ and $t$ such that the path lies inside a region $r$ and is less than a given length $l$ can be specified as:

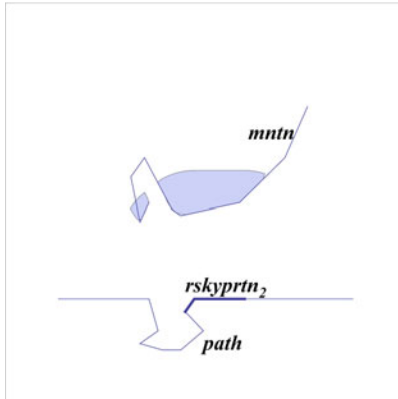
$$PathExists(s, t, r, l) \equiv \exists q, Inside(q, r) \wedge Distance(s, q) + Distance(q, t) \leq l$$

In Fig. 4, $s \leftarrow (0, 45)$, $t \leftarrow (20, 5)$, $r \leftarrow SafeRegion((x, y), \{r_1, r_2, r_3, r_4\}, 2)$, $l \leftarrow \sqrt{1010}$. The region consisting of all paths that satisfy the constraints is computed from the function problem

[3] SPS refers to objects by their labels and acquires their locations from the pels they occupy. But, for the CAD algorithm, numerical coordinates need to be provided.




(a) The shaded region is the risky region prone to ambush due to enemies hiding at *mntn*. The risky portions of *path* are those inside the risky region.




(b) Troops traveling on $rskyprtn_2$ (in bold) might be ambushed as they are within firepower range from enemies hiding in the shaded region.

**Fig. 5.** A simplified ambush analysis scenario to illustrate the performance of SPS

$$FindPath(q, s, t, r, l) \equiv Inside(q, r) \wedge Distance(s, q) + Distance(q, t) \leq l$$

where $q \leftarrow (x, y)$, and shown in Fig. 4(b).

From the results, the problem solver infers that $T_3$ might be $T_1$. Next he repeats the same for entities $T_3$ and $T_2$, and finds that $T_3$ might be $T_2$ as well. The sensor database informs that there are two sensor fields – `SENSOR1`, `SENSOR2` – in the area of interest but there has been no report from them of any passing vehicle. Problem solver wants to verify whether any of the paths passes through any of the sensor fields. He specifies the problem $IntersectRegions(r_1, r_2)$ to compute the intersection of two regions $r_1$, $r_2$ as:

$$IntersectRegions(r_1, r_2) \equiv \exists q, Inside(q, r_1) \wedge Inside(q, r_2)$$

He computes the problem $IntersectRegions(paths_{13}, s_1)$ where $paths_{13} \leftarrow FindPath(q, p_1, p_3, r, l)$ and $s_1$ is the region covered by `SENSOR1`. In our scenario, the solution is $True$. Next the problem solver wants to know whether there exists a path between points $p_1$ and $p_3$ safely avoiding the obstacles and enemy firepower range such that it can be traversed in time $t_3 - t_1$. He computes $PathExists(p_1, p_3, r_{13}, l)$, where $r_{13} \leftarrow paths_{13} - s_1$, which returns $True$. The inference follows that $T_3$ might be $T_2$. The same reasoning is repeated for $T_3$ and $T_2$; $Intersect(paths_{23}, s_2)$ returns $True$ while $PathExists(p_2, p_3, r_{23}, l)$ returns $False$. The inference follows that $T_3$ cannot be $T_1$. Hence, $T_3$ is $T_2$.

### 4.2 Ambush Analysis

There are two main factors – range of firepower and sight – that determine the area covered by a military unit. Presence of terrain features, such as mountains, limit these factors and allow units to hide from opponents. These hidden units not only enjoy the advantage of concealing their resources and intentions from the opponents but can also attack the opponents catching them unawares if they are traveling along a path that is within the sight and firepower range of the hidden units, thereby ambushing them. Thus, it is of utmost importance for any military unit to determine the areas or portions of a path prone to ambush before traversing them. In this section, given a curve or region as a hiding place and the firepower and sight ranges, we show how the regions and portions of path prone to ambush is efficiently computed by the proposed SPS.

Given a curve $c$ and the range $d$, the problem $RiskyRegion(q, c, d)$ is defined as the set of all points covered by that range from $c$, and specified as:

$$RiskyRegion(q, c, d) \equiv \exists a, On(a, c) \wedge Distance(a, q) \leq d$$

where $q \leftarrow (x, y)$. In order to compare the actual computation times required to solve the problem, we constructed a very simple diagram consisting of two curves, $path$ and $mntn$, where $path \leftarrow \{(-25, -10), (-5, -10), (-3, -15), (-7, -17), (-2, -18), (2, -18), (7, -15), (3, -12), (5, -10), (40, -10)\}, mntn \leftarrow \{(-5, 5), (-7, 2), (-9, 9), (-6, 12), (0, 4), (2, 3), (15, 5), (25, 12), (30, 20)\}$. The solution to $RiskyRegion(q, mntn, 15)$ is shown in Fig. 5(a) where $mntn$ is an obstacle for hiding (e.g., mountain range). $RiskyRegion(q, r, d)$ is specified by replacing $On(p, c)$ with $Inside(p, r)$.

Again, given a curve $c_1$ as a path, a curve $c_2$ for hiding, and a firepower range $d$, the problem $RiskyPortionsofPath(q, c_1, c_2, d)$ is defined as parts of $c_1$ covered by that range from $c_2$. Thus,

$$RiskyPortionsofPath(q, c_1, c_2, d)$$
$$\equiv On(q, c_1) \wedge \exists p, On(p, c_2) \wedge Distance(p, q) \leq d$$

where $q \leftarrow (x, y)$. Solution to $RiskyPortionsofPath(q, path, mntn, 15)$ is shown in Fig. 5(a). If the hiding place is a region $r$ instead of the curve $c_2$, the problem can be specified by replacing $On(p, c_2)$ with $Inside(p, r)$.

The region behind $c_2$ where the enemies might be hiding is the set of all points that are behind $c_2$ with respect to each point on the risky portions of curve $c_1$. However, enemies might be hiding not anywhere behind a mountain but within a distance from where they can ambush the friendly units. An important problem is $BehindCurvewrtRiskyPath(q, c, c_2, d)$ where $d$ is the distance from where the enemies can ambush them.

$$BehindCurvewrtRiskyPath(q, c, c_2, d)$$
$$\equiv \exists a, On(a, c) \wedge BehindCurve(q, c_2, a) \wedge Distance(a, q) \leq d$$

where $q \leftarrow (x, y)$. Solution to $BehindCurvewrtRiskyPath(q, rskyprtn_2, mntn, 20)$ is shown in Fig. 5(b). If the hiding place is a region $r$ instead of $c_2$, the problem can be specified by replacing $On(p, c_2)$ with $Inside(p, r)$. A comparison between the CAD algorithm and our proposed SPS of actual computation times for problems relevant to entity re-identification and ambush analysis is in Table 1.

**Table 1.** Comparison of computation times (in seconds) between the CAD algorithm and our proposed SPS for instances of spatial problems discussed. A 2.8 GHz PC with 4 GB RAM, 5356 MB virtual memory, 32-bit operating system was used, and implemented in *Mathematica*. SPS maximum resolution was 128 × 128, comparable to CAD's piecewise linear approximation. Time required for object approximation and filling data structures were not included. Below, $q \leftarrow (x, y)$ and "$OOM$" refers to out of memory.

| **Spatial Problem** | **SPS** | **CAD** |
|---|---|---|
| $SafeRegion(q, \{r_1, r_2, r_3, r_4\}, 2)$ | 3.0 | 5.5 |
| $PathExists(s, t, r, \sqrt{1010})$ | 1.2 | $OOM$ |
| $IntersectRegions(paths_{13}, s_1)$ | 1.5 | $OOM$ |
| $CurveInsideRegion(c, r)$ | 5.7 | 175.01 |
| $RiskyRegion(q, mntn, 15)$ | 0.5 | 0.11 |
| $RiskyPortionsofPath(q, path, mntn, 15)$ | 0.1 | 0.48 |
| $BehindCurve(q, mntn, (5, -10))$ | 0.6 | 0.71 |
| $BehindCurvewrtRiskyPath(q, rskyprtn_2, mntn, 20)$ | 7.4 | 15.33 |

## 5 Conclusion

Our goal was to build a general framework for executing perceptions/actions such that a human can use them opportunistically for DR. These perceptions/actions were transformed into domain/task-independent $2D$ spatial problems and specified as QCSPs in the real domain. Traditional algebraic approaches for solving such QCSPs are inherently doubly exponential in time complexity. We proposed a general framework of spatial problem solving where a small vocabulary of predicates are implemented as procedures, using which a spatial problem is specified in first-order logic. Our SPS searches for the solution in the space of the diagram instead of in the space of algebraic equations/inequalities. We proved the correctness of our approach and showed it to be more efficient than CAD in executing perceptions/actions for two army applications.

## References

1. Chandrasekaran, B., Kurup, U., Banerjee, B., Josephson, J.R., Winkler, R.: An architecture for problem solving with diagrams. In: Blackwell, A.F., Marriott, K., Shimojima, A. (eds.) Diagrams 2004. LNCS (LNAI), vol. 2980, pp. 151–165. Springer, Heidelberg (2004)
2. Pisan, Y.: A visual routines based model of graph understanding. In: Proc. 17th Annual Conf. Cognitive Science Society. Erlbaum, Pittsburgh (1995)
3. Tessler, S., Iwasaki, Y., Law, K.: Qualitative structural analysis using diagrammatic reasoning. In: Proc. 14th Intl. Joint Conf. AI, Montreal, pp. 885–893 (1995)
4. Lindsay, R.K.: Using diagrams to understand geometry. Computational Intelligence 14(2), 238–272 (1998)
5. Jamnik, M.: Mathematical Reasoning with Diagrams: From Intuition to Automation. CSLI Press/Stanford University, CA (2001)

6. Ferguson, R.W.: Magi: Analogy-based encoding using symmetry and regularity. In: Proc. 16th Annual Conf. Cognitive Science Society, Atlanta, GA, pp. 283–288 (1994)
7. Banerjee, B.: Spatial problem solving for diagrammatic reasoning. PhD thesis, Dept. of Computer Science & Engineering, The Ohio State University, Columbus (2007)
8. Egenhofer, M.J., Franzosa, R.D.: Point set topological relations. Intl. Journal of Geographical Information Systems 5, 161–174 (1991)
9. Frank, A.U.: Qualitative spatial reasoning about distances and directions in geographic space. Journal of Visual Languages and Computing 3, 343–371 (1992)
10. Cohn, A.G., Bennett, B., Gooday, J.M., Gotts, N.: RCC: A calculus for region based qualitative spatial reasoning. GeoInformatica 1, 275–316 (1997)
11. Brown, C.W., Davenport, J.H.: The complexity of quantifier elimination and cylindrical algebraic decomposition. In: Proc. Intl. Symp. Symbolic and Algebraic Computation, pp. 54–60. ACM, New York (2007)
12. Collins, G.E., Hong, H.: Partial cylindrical algebraic decomposition for quantifier elimination. Journal of Symbolic Computation 12(3), 299–328 (1991)

# Toward a Physics of Equations

David Landy

Department of Psychology, University of Richmond, Richmond, VA 23173
dlandy@richmond.edu

**Abstract.** Papers on diagrammatic reasoning often begin by dividing marks on paper into two basic classes: diagrams and sentences. While endorsing the perspective that a reasoning episode can be diagrammatic or sentential, I will give an overview of recent evidence suggesting that apparently symbolic expressions in algebra and arithmetic are frequently treated as diagrammatic or even pictorial depictions of objects and events—events that occur not in the content of the expression, but within the notation itself. This evidence suggests that algebra is sometimes less a matter of rules and abstract syntax, and more a matter of constraints on the physical behavior and part-whole structure of notational *things*: an idiosyncratic notational physics, whose laws constrain the structure of proofs. These considerations suggest that whether some marks are a diagram depends on exactly how a user engages them.

**Keywords:** mathematical cognition, experimental psychology, reasoning.

## 1 Introduction

Our understanding of diagrams often begins from a division of external representation schemes into diagrams and sentences [1,2]. Pictures, blueprints, and maps serve as prototypes of diagrams; the default sentential representation is spoken language. Although modern mathematical expressions superficially resemble quintessential diagrams in that they are typically set off in their own physical space, and use two-dimensional physical space (e.g., in subscripts and superscripts), expressions exhibit many properties typical to sentential schemes. The central contention of this paper is that mathematical forms can be profitably viewed both as sentences and diagrams, depending on how they are used to support reasoning.

It is widely agreed that the content domain of a representation scheme does not determine whether or not it is diagrammatic. As an example, statements of propositional logic may be expressed either with words or with Euler diagrams. A more common perspective is that the relationship between form and content fixes the status of a representation scheme [1,2]. On these accounts, diagrams depict content-level relationships through a homomorphism to physical structures. In sentential systems, physical relationships such as ordering may have a homomorphic relationship to the abstract grammatical structure of a proposition, but not to meaning in the depicted content. Whether a system is sentential or diagrammatic has entirely to do with the system's relationship to its intended content; the user of the system does not contribute to the distinction.

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 160–166, 2010.

In this paper, I suggest that categorizing representations in this way glosses the actual psychological processes employed by reasoners in solving problems. In particular, in some cases reasoners treat an external representation *as though* it was depicting something that it normatively isn't. This paper provides an overview of recent evidence collected by myself and others, demonstrating that low-level perceptual features of mathematical expressions have a substantial impact on reasoning. Previously, I have argued that reasoning with notations involves the development of specialized perceptual mechanisms [3,4]. Here, I develop an alternative interpretation (see also [5]): Learning mathematical rules involves learning a kind of commonsense physics—the physics of mathematical objects. That is, people often apply to mathematical forms reasoning processes which they typically apply to physical objects undergoing various kinds of change and motion. On this latter interpretation, although there is no homomorphism between the form of a mathematical expression and its normative content, there is an iconic relationship between the surface form of an expression and the representation of symbols as physical objects. The result is a diagrammatic relationship between the physical structure of an expression, and the expression as conceived by the reasoner.

## 2 Mathematical Expressions as Physical Objects

People know a lot about physical objects. We have a fairly good understanding of how objects move, collide, bend, and break [6]. We also have rich mechanisms for recognizing object boundaries—segmenting visual scenes into objects and their parts.

Infants exhibit knowledge of and interest in the way that objects move, change, appear, and vanish [7-9]. As children explore their environments, they also develop an understanding of which features cue object boundaries [7,10]. By the time they are adults, human reasoners have a rich and developed ontology of different object types, with different kinds of properties. In the same manner that children may initially apply general principles of object segmentation and motion, and over time learn appropriate particular rules for particular kinds of objects [7,8], reasoners learning mathematical systems may adapt general segmentation and dynamic event processes to suit the structure of mathematical expressions. Causally potent experience with objects and affordances shapes children's understanding of specialized situations and objects [8-10]; in a similar manner, causally potent experience with mathematical computations may lead to the incorporation into general physical understandings of constraints particularly suitable to mathematics.

At the very least, people occasionally talk about notations as though they were objects: in Britain, for instance, improper fractions such as $\frac{17}{5}$ are often called "top heavy," suggesting a metaphor to an object standing upright in gravity. Talk of equations as "balanced" suggests similar implicitly gravitational considerations. People often talk of equation solution in terms of motion. Pilot work in my lab found that when asked to describe how to solve generic linear equations, approximately 10-15% of subjects spontaneously described the process of isolating the variable being solved for by using the word "move," and an additional 10% used language suggesting motion. If such descriptions are not purely metaphorical, but indicate processes used in actual online computation, one straightforward hypothesis is that by and large, physical models of mathematics map symbols into material forms by using object segmentation to implement grammatical rules, and understand axioms using representation systems which apply to dynamic events.

## 2.1 Material Approaches to Formal Grammars

What are the objects that populate the world of algebraic notations? A natural guess is that the structural part-whole segmentation mirrors the formal grammar: objects are expressions, whose parts are connectives and sub-expressions. For example, the expression $9+6\times 7$ is one object, made up of three parts: $9$, $+$, and $6\times 7$. The last part is then itself a compound object, made up of $6$, $\times$, and $7$.

Visual grouping principles require very little training to account for mathematical behaviors. This is because the visual structure of algebraic notations already largely aligns spatial and syntactic proximity[11]. For example, consider the expression

$$3x^2 + \frac{x+\sqrt{4+7x}}{(3+y)x} \ . \qquad (1)$$

The division sign forms a vertical barrier paralleling the syntactic separation into numerator and denominator. Parentheses form a perceptual region, visually grouping the terms within. The overbar in the radical also creates a visually connected region (and is itself the vestige of an obsolete grouping system [12]). Exponents are placed very close to their bases, and omission of the multiplication sign causes products to be spaced more closely than sums. However, some mathematically meaningful segments are not handled appropriately by domain-general grouping principles. In order to correctly group simple arithmetic expressions in uniformly spaced typefaces, such as $3+5\times 4=23$, it is necessary to visually group terms surrounding multiplications preferentially over additions, and those preferentially over equals signs.

An amodal account of expression parsing that relies strictly on rules expressed formally to determine structure provides no particular predictions about the physical requirements of a mathematical system, beyond that the symbols be clearly readable, and close enough that the next symbol can be seen before the previous one is forgotten [13]. However, if the implementation of the rules of interpretation comprises learning idiosyncratic grouping and segmentation principles layered over the usual grouping processes that apply to physical scenes, then a basic biconditional prediction follows: physical features that affect object segmentation should influence the computation of formal syntax, and physical features that influence formal syntax should influence segmentation.

Substantial evidence supports the former of these two conclusions. Kirshner [11] demonstrated that students learning a novel arithmetic notation incorporated spatial proximity into syntactic operations. Subjects were better able to respect the order of operations while performing arithmetic expressions, when the novel notation contained spatial proximity relations similar to that of typical algebraic expressions (that is, when higher precedence operations were closer). Landy & Goldstone [3] demonstrated that the effect was not limited either to spacing, or to features present in standard notations. A wide variety of grouping principles affect both perceptual grouping and mathematical competence (see Figure 1).

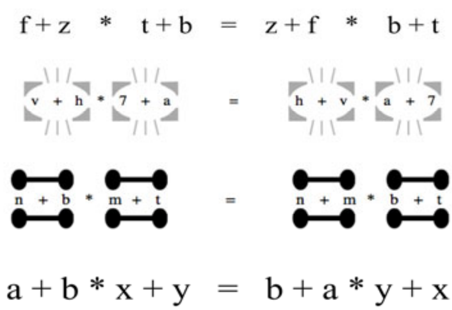



**Fig. 1.** Sample stimuli from Landy & Goldstone [3] illustrating the effect of (from top to bottom) physical space, common region, connectedness, and alphabetic proximity. In each case, participants were biased to see visually grouped objects as syntactically bound.

The implication that mathematical relations should influence spatial perception and grouping of mathematically relevant objects is relatively unexplored (though see [14]). The conceptualization of formal learning presented here makes emphatically the prediction that learning syntax should affect object segmentation.

### 2.2 Mathematical Rules as Constraints on Physical Change

Once one has found the objects, one must understand how they behave. To be useful, the laws of dynamics must, of course, by and large guarantee mathematically valid results. Many valid manipulations can be accomplished by assuming that expressions are semi-rigid physical forms, with parts that move continuously and that can be created and destroyed in specific kinds of ways and circumstances.

Let's consider two ways to solve linear equations, one using a sentential approach, the other a material. Table 1 presents one derivation of the solution to $y\times 3+2=8$, using Euclid's axioms. These axioms specify a family of equations—including each of the equations in Table 1—that have the same solution. The key to this method is to apply the rules to find a member of the family whose answer is obvious.

**Table 1.** A sentential approach to equation solving

| Statement | Justification |
|---|---|
| $y\times 3+2=8$ | Given |
| $y\times 3+2-2=8-2$ | Apply Axiom of addition |
| $y\times 3=8-2$ | Arithmetic Simplification |
| $\frac{y\times 3}{3}=\frac{8-2}{3}$ | Apply axiom of division |
| $y=\frac{8-2}{3}$ | Arithmetic simplification |

The sentential approach treats the derivation as a sequence of separate statements from the same set (in the case of linear equations, equations with a fixed solution). A short proof or derivation is similar to a paragraph: it is a sequence of separate statements that follow closely upon each other, but which consist themselves of wholly separate words and phrases. Each sentence is a separate thing.

Alternatively, one can see the proof structure as a narrative of transformation, in which one or a few physical objects undergo a succession of alterations. Consider the proof shown in Table 2. It is identical to Table 1, except that two steps have been collapsed. However, the justification is quite different. Here the solver conceptualizes a single equation, undergoing physical transformations. It is unambiguous that there is a single equation, which appears in different forms in the three proof lines as a result of the changes it has undergone (the 2 has “moved rightward” and “changed sign”). Note that this is a kind of motion specific to mathematics; when an object crosses an equation boundary, it must transform (by changing sign).

Consistent with the idea that people sometimes solve problems by treating notations as though they represented motion, Landy & Goldstone [15] found that people solving linear equations were systematically affected by the simultaneous perception of actual motion (see Figure 2). When irrelevant dots in the background behind a problem moved in the same direction that the terms would be moved in the motion-based strategy, error rates were lower than when the dot motion was opposite to that implied by the equation. Furthermore, this effect grew larger with increased mathematical experience, and was strongest on problems that were most familiar (those involving addition and multiplications, rather than subtractions and divisions), suggesting that experience leads to increased use of motion-based conceptions in notations. This is consistent with the hypothesis that situated experience with the physical contingencies of mathematical proofs drives the construction, in reasoners, of physical representations of formal systems.

Although we initially described Table 1 using an sentential approach, there is also a readily available material interpretation. Rather than seeing the lines as separate statements, with derivational links justified by the application of rules (the most formally sound perspective), one can see the lines of the proof as dynamic events happening, again, to a single object. In this case, the events would not be motion events, but instead involve creation and destruction of terms. So between lines three and four of Table 1, a “divided by 3” is introduced on both sides of the equation. This language mirrors the usual justification for the axiom itself (“likes done to likes yield likes”); however, in the material approach, the introduced term is not semantically de-referenced. We do not take away two-thirds of each of two piles of stones; we instead insert a pair of symbols “/3.” Thus, the justification is intrinsically tied to the notation

**Table 2.** A material approach to equation solving

| Statement | Justification |
|---|---|
| $y \times 3 + 2 = 8$ | Given |
| $y \times 3 = 8 - 2$ | Move 2 rightward (and change sign) |
| $y = \dfrac{8-2}{3}$ | Move 3 rightward (and change sign) |

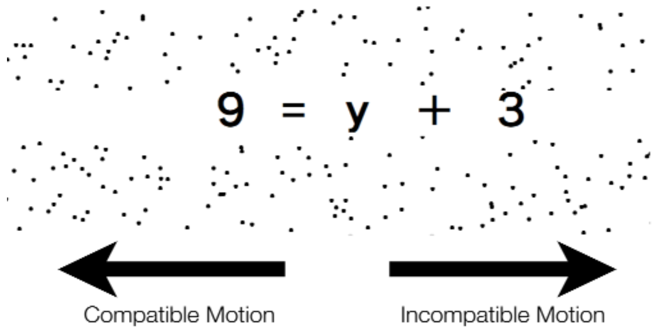



**Fig. 2.** Sample stimulus from [15]. In the stimulus dots moved quasi-randomly. For this problem, compatible motion is leftward motion; incompatible motion is rightward.

display, rather than to an underlying situation model. Although I know of no direct evidence that people in fact engage in *this* kind of symbolic-material reification, the material approach predicts that people do. This approach therefore suggests particular phenomena. For instance, semantic and visual facts that prime creation and cancellation should encourage corresponding computational processes and vice versa.

## 3 Discussion

So what kind of representations are mathematical expressions? When taken to be representations of underlying situations or abstract facts, an abstract syntax mediates physical form and meaning. At least some of the time, however, it appears that the actual mechanisms involved in notation manipulation treat notations *as though* they were literal depictions of physical objects. In these cases, the physical properties of expressions are iconic representations of literal physical objects, and are therefore quintessentially diagrammatic rather than sentential. These considerations suggest a dual treatment of mathematical forms. As depictions of objects, forms can be segmented, manipulated, created, destroyed, and simply observed. As referential symbol tokens, forms can be unpacked into general, meaningful statements. Together, these two systems yield powerful, domain-general syntactic computation.

Not every sentential representation is pictorial in the sense discussed here. The physical interpretation of formal languages makes them different from written natural language. Although proximity may well play a role in sentential understanding, there is no reason to suspect that dynamic transformations of notations play a significant role in sentence understanding. The dual interpretability of mathematical systems may constitute a basic virtue of our modern symbolic systems.

All of this can be summarized as a fairly trivial point about diagrams: diagrams often work by letting one kind of thinking you're good at stand in for a another kind of thinking you're bad at, as when Venn diagrams allow one to come to conclusions about set relations by thinking about spatial relations. Similarly, mathematical expressions written in our modern notation often let you come to conclusions about formal statements and proofs by thinking, instead, about objects in space.

## References

1. Chandrasekaran, B.: What makes a bunch of marks a diagrammatic representation, and another bunch a sentential representation? In: Proceedings of the AAAI Spring Symposium (2005)
2. Stenning, K.: Seeing reason: Image and language in learning to think. Oxford University Press, Oxford (2002)
3. Landy, D., Goldstone, R.L.: How abstract is symbolic thought? Journals of Experimental Psychology: Learning, Memory, and Cognition 23, 720–733 (2007)
4. Goldstone, R.L., Landy, D., Son, J.Y.: The education of perception Topics in Cognitive Science (in Press)
5. Landy, D., Goldstone, R.L.: Pushing symbols: Perceived motion in algebraic problem solving (submitted)
6. Proffitt, D.R.: Naïve Physics. The MIT Encyclopedia of the Cognitive Sciences (1999)
7. Wilcox, T.: Object individuation: Infants' use of shape, size, pattern, and color. Cognition 72, 125–166 (1999)
8. Baillargeon, R.: How do infants learn about the physical world? Current Directions in Psychological Science 3(5), 133–140 (1994)
9. Needham, A., Barrett, T., Peterman, K.: A pick-me-up for infants' exploratory skills: Early simulated experiences reaching for objects using 'sticky mittens' enhances young infants' object exploration skills. Infant Behavior and Development 25, 279–295 (2002)
10. Needham, A.: Improvements in object exploration skills facilitate the development of object segregation in early infancy. Journal of Cognition and Development 1, 131–156 (2000)
11. Kirshner, D.: The visual syntax of algebra. Journal for Research in Mathematics Education 20, 274–287 (1989)
12. Cajori, F.: A History of Mathematical Notations. Dover Publications, New York (1993)
13. Haugeland, J.: Artificial Intelligence: the Very Idea. MIT Press, Cambridge (1989)
14. Jansen, A.R., Marriot, K., Yelland, G.W.: Comprehension of algebraic expressions used by experienced users of mathematics. Quarterly Journal of Experimental Psychology 56A, 3–30 (2003)
15. Landy, D., Goldstone, R.L.: Pushing symbols. In: The 31st Annual Conference of the Cognitive Science Society, Amsterdam (2009)

# Usability of Accessible Bar Charts

Cagatay Goncu, Kim Marriott, and John Hurst

Clayton School of IT, Monash University
cagatay.goncu@infotech.monash.edu.au,
kim.marriott@infotech.monash.edu.au,
john.hurst@infotech.monash.edu.au

**Abstract.** Bar charts are one of the most commonly used diagram types. *Tactile* diagrams are a widely used technique for presenting graphics to people who are blind. We explored how to present bar charts using a tactile presentation. Our user study used blind participants and evaluated both user preferences and performance. We found that providing grid lines and values in a tactile diagram was preferred to a direct transcription. In addition, presenting the data as a tactile table was preferred to a tactile chart. Both of these approaches reduced the error rate, and presentation as a table had performance benefits. We also investigated the comparative usability of: a tactile presentation, an *audio* description of the bar chart, and a *tactile/audio* presentation in which a tactile diagram is overlaid on a touch-sensitive device which provides audio feedback on demand. We found that tactile was the most preferred while audio was the least.

## 1 Introduction

Bar charts are one of the most common kinds of diagrams (see Figure 1). They are widely used in educational material and in newspaper articles to present multi-dimensional information. However, not everyone can see a bar chart. Here we investigate how to present bar charts to people who are blind.

One common technique used to present diagrams to people who are blind is to use a tactile presentation which allows the user to *feel* the diagram. The main focus of this paper is how to present bar charts using tactile presentations. We conducted two user studies using blind participants and evaluated both user preferences and user performance. The paper has two main contributions.

The first contribution is to investigate the effectiveness of different styles of layout for tactile (and tactile/audio) presentation of bar charts. The most common approach is to use a direct tactile transcription of the original visual representation. However, like Challis and Edwards [1], we believe that good layout for tactile presentation of charts and diagrams may have significant differences to that for visual presentation because of different characteristics of the vision and haptic sensory systems.

We first investigated whether readability can be improved by adding gridlines and/or by adding explicit Braille values at the top of the bars (as some transcription guidelines [2,3] suggest). We then made some major modifications to the

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 167–181, 2010.

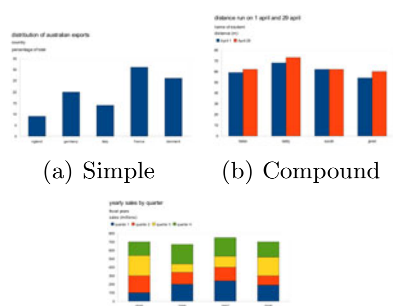
vertical charts. We conjectured that tactile bar charts would be more readable when presented using horizontal bars rather than vertical bars since this is better suited to a top-down, left-to-right reading of the diagram. We also investigated whether a tactile table presentation of the data might be more preferable to a tactile bar chart presentation. This was motivated by user studies suggesting that depending on the task, sighted people performed better on tables in terms of time and accuracy [4].

A limitation of a pure tactile presentation is that text must be presented as Braille which takes up considerable space and is not accessible to a significant number of users. Furthermore, it can be difficult to use easily distinguishable textures when translating a diagram which heavily utilizes patterns and colour to differentiate between kinds of diagram elements. Tactile diagrams are not the only possible method for presenting a bar chart to a blind person. Audio description of the diagram and its content is another common technique. This, of course, has the disadvantage that the 2-D nature of the original diagram is lost. More recently, touch sensitive devices like the IVEO [5] and Tactile Talking Tablet (TTT) [6] have been used to provide a combination of audio and tactile which potentially overcomes the limitations of a pure tactile or a pure audio presentation. These devices allow a tactile diagram to be overlaid on top of a pressure-sensitive screen. During reading the user presses on the tactile diagram to obtain audio feedback.

Our second contribution is to describe a preliminary usability study to compare the usability of the three different presentation media: tactile, audio and tactile/audio for presenting bar charts. To the best of our knowledge, there has been no formal user study to compare these techniques for any kind of diagram.

## 2 Related Work

A number of psychological studies which provide a basis for tactile diagram design guidelines have been conducted [7,8,9]. These studies indicated the limitations of the tactual senses and the importance of tactile symbol choice, element separation, and texture selection. For example, in [10], the authors reported that texture outlines, height, and separation of symbols were the important factors to distinguish different textures, and affected the performance of users while reading tactile maps. In a study which evaluated tactile design guidelines based on the above for a tactile music notation [1], the error rates of the participants decreased when the design guidelines were used.



(a) Simple (b) Compound

(c) Stacked

**Fig. 1.** Kinds of bar charts

Four different designs of tactile line graphs with gridlines (extended value marks on both axis) were investigated by [11]. The authors evaluated no grid, grid-on-graph, grid overlay, and grid underlay configurations, and found that grid-on-graph and grid underlay ones were the most effective ones in terms of speed and accuracy, but the no grid ones were the most preferred. In a similar study [12], the authors investigated three different formats of line graphs, no-grid, box-no-grid (duplicated axes on either side of the graph), and box-grid (graph was superimposed on a regular grid with duplicate axes). They reported that the box-grid format was more effective in terms of errors, but not in time. However, questions asking for the trend were answered more quickly with no-grid graphs than the others. Participants preferred the box-grid format.

There has been considerable research into so called *multi-modal interfaces* which utilize a mixture of speech, non-speech audio, touch, and force feedback to present diagrams to blind people. Most research in this area has focused on the design of new devices and interfaces and then a user study of the new device. There are few comparative user studies. The two most relevant studies are [13,14]. These compared two force-feedback devices for bar chart exploration. Speech and non-speech sound was also used to support the interaction. It was found that using audio improved the performance of the participants. This multi-modal system was compared with traditional tactile diagrams in the second study. They found that the performance on correct answers was increased in the multi-modal interface, but time performance was not improved. They conjectured that the main reason for this outcome was the unfamiliarity with the multi-modal system. In addition, they stated that the single point of contact of the multi-modal system had a disadvantage over a tactile diagram where the participants can touch the diagram with multiple fingers.

In another study, a prototype system using a graphics tablet, and a VTPlayer tactile mouse was evaluated [15]. The users explored the virtual bar chart on the graphics tablet by using the stylus. Based on the position of the stylus, the two tactile arrays of Braille cells on the mouse, which was held by the other hand, were activated. The activation of the pins in these cells were determined by the pixel values pointed by the stylus. Speech audio feedback was also provided by clicking the button on the stylus. It was reported that the participants preferred the prototype system to the tactile only presentation.

Tactile/audio diagrams were also used in mathematics testing to present mathematical graphs [16]. The TTT was used with different tactile overlays that included tactile diagrams. The authors reported that the participants using TTT improved the performance in terms of correct answers.

## 3 Experiment 1

The visual and haptic sensory subsystems have different characteristics. The visual subsystem has a wide area of perception that provides parallel continuous information acquisition and it also has a narrow focus of attention frame which can provide detailed information. The haptic subsystem can provide most of the same information as the visual subsystem such as shape, size, texture, and position [17]. However, the extent of the perceptual field is less than the visual field and limited to the hands. Moreover, since perception requires movement of the fingers and hands, information acquisition is less parallel than vision [17].

This suggests that good layout of tactile diagrams may be quite different to good layout of visual diagrams. In particular it suggests that it may be useful to explicitly provide information in the tactile representation when it requires parallel wide focus processing to extract the same information from the visual representation. This provides some support for providing grid lines and value labels in tactile bar charts (which have been suggested in some transcription guidelines [2,3]) since this will obviate the need for frequent back and forth movements to the axis lines to understand the bar chart values. Further support for this is provided by the study into grid lines for tactile presentation of line graphs [12]. Our first user study investigated this hypothesis:

H1 Adding grid lines, and value labels to a tactile bar chart layout will improve readability of the diagram and will be preferred by users.

We investigated this for simple, compound and stacked bar charts. We looked at user preferences and user performance in terms of error and time.

### 3.1 Method

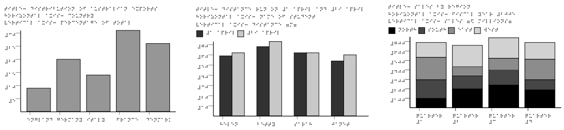
**Participants:** 12 participants, 5 born blind and 7 late blind, between the ages of 19 to 60 completed the experiment. They all had previously read a bar chart. Participants were recruited by advertising the study on two email lists for print-disabled people in Australia by a posting from Vision Australia.

**Materials and design:** The starting point was the three kinds of visual bar charts shown in Figure 1. We created 9 variations of these bar charts that preserved the number of bars and the information but which varied in the data values, such as country name and percentage.

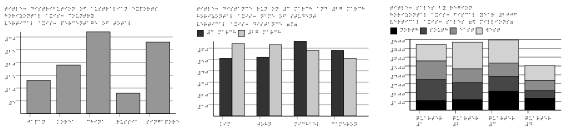
For each bar chart kind (simple, compound, stacked) we created tactile diagrams in three chart styles. The first style, *direct*, was a direct transcription of the standard visual representation. It was similar to the original chart except that colours were replaced by textures and the labels were replaced by Grade 1 Braille text. The second style, *grid*, had grid lines while the third, *grid+value*, had both grid-lines and a Braille numeric value on the top of each bar giving its precise value. In the *grid+value* style of the stacked bar chart, we put the values of individual bars inside the bars. But, we did not provide labels for bars with small values, because the Braille labels would not fit in that area. We also included the total sum at the top of each stacked bar. An example of the three styles for each kind of bar chart is shown in Figure 2.

The tactile bar charts were created with the Tactile Chart Tool [18]. We used Braille paper as the presentation medium and used a Tiger embosser to emboss the charts.

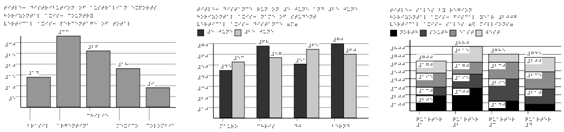
For each tactile chart we developed three questions designed to explore the efficacy of three major tasks on bar charts: detail-on-demand (*DoD*), *browse* and



(a) Direct transcription



(b) With grid lines



(c) With grid lines and values

**Fig. 2.** Tactile chart styles used in Experiment 1. Note that in tactile diagrams the darker the area, the higher it is raised.

*search* [19]. In detail-on-demand users must find information about a particular element and its properties. In a search task they find a particular element on the layout based on a property, while browsing requires users to gain an overview of the entire chart. Sample questions used for the simple bar chart (which showed the percentage of Australian exports to different countries) were:

- What is the percentage of Australian exports to Italy? (*DoD*)
- Which countries are the top two importers from Australia? (*search*)
- Which countries are the goods exported greater than %18? (*browse*)

For the compound bar chart (which showed the distance run by different people on two different dates) example questions were:

- What is the distance run by Sarah on 29th of April? (*DoD*)
- Who run the maximum distance on the 1st of April? (*search*)
- Do they generally run further on the 1st or the 29th of April? (*browse*)

For the stacked bar chart (which showed a company's sales in four quarters for four years) example questions were:

- What is the amount of sales at Quarter 2 in 2006? (*DoD*)
- In which quarter do the sales reach its minimum in 2007? (*search*)
- What is the trend of the sales at Quarter 3? (*browse*)

To correct for the order of presentation we created three counterbalanced versions of the experiment which ensured that each chart style (direct, grid, grid+value) was shown an equal number of times as the first, second and third for each chart kind (simple, compound, stacked). A second possible confounding factor was the exact choice of data and question. We did not think this would affect preference or error data but thought it might (slightly) affect question answering time. We did not correct for this for the first six participants but for the last six participants we modified the three counterbalanced versions to correct for this. We used the same data and questions for the first chart in the three versions, and for the second chart (but different to the first chart) and for the third chart (but different to the preceding charts).

**Procedure:** Experiments were performed in a small room either at Vision Australia or at Monash University. The procedure was as follows.

First, participants were asked to sign a consent form which had previously been sent by email to them and which they were given a tactile version on the day. This also provided a short description of how the experiment would be conducted and what type of information would be collected. They were then asked how long they had been blind. Then we presented the participants with a sample tactile bar chart to familiarize them with the presentation medium and to give them an idea of the kind of charts they would see during the study. They were asked if they were familiar with bar charts.

Then the following process was repeated for simple bar charts, then compound and finally stacked. For each kind of bar chart, the participant was shown a tactile diagram for each of the three styles one at a time. The participant was asked to explore the diagram and let us know when he/she was ready to answer questions about it. They were then asked to answer the three questions for that chart. The time taken to explore and answer each question were recorded as were their answers. After reading and answering questions for the three styles (direct, grid, grid+value), the participants were asked to rank them from the least preferred to the most preferred. During this process, participants were invited to give comments and explain the features that influenced their rankings.

### 3.2 Data Analysis and Results

**User preferences:** As Table 1 shows, participants much preferred the style with grid lines and bar values and their second preference was for the style with grid lines while the direct transcription was the least preferred. Since the small values in the contingency table are not suitable for a chi-square analysis, we use multinomial distribution to calculate the significance levels under the null hypothesis where preferences for the styles are equally likely. Selecting a particular style of layout as the first preference is significant for all kinds of bar charts, $P(Y \geq 8|n = 12, p = .33) = .056$; and as the third preference is significant for simple and compound bar charts, $P_{simple}(Y \geq 9|n = 12, p = .33) < .03$, $P_{compound}(Y \geq 8|n = 12, p = .33) = .056$, but not for stacked bar charts (Y = number of preferences). This provides support for Hypothesis H1. Those participants who preferred the direct translation appeared to do so because it was less cluttered. Representative comments were:

- Direct-transcription versions are less cluttered, but not accurate.
- Gridlines helped not to lose orientation on the layout.

**Table 1.** Chart style preferences in Experiment 1: first choice count (second choice count)

| Kind of chart | Direct | Grid | Grid+Value |
|---|---|---|---|
| Simple | 2 (1) | 2 (8) | 8 (3) |
| Compound | 2 (2) | 2 (9) | 8 (1) |
| Stacked | 2 (3) | 2 (9) | 8 (0) |

**Table 2.** Error rate of chart styles in Experiment 1: total number of errors

| Task | Simple | | | Compound | | | Stacked | | |
|---|---|---|---|---|---|---|---|---|---|
| | Direct | Grid | Grid+Value | Direct | Grid | Grid+Value | Direct | Grid | Grid+Value |
| DoD | 3 | 3 | 1 | 4 | 2 | 0 | 11 | 11 | 3 |
| Search | 0 | 0 | 0 | 1 | 1 | 0 | 5 | 2 | 3 |
| Browse | 2 | 3 | 2 | 0 | 0 | 0 | 6 | 5 | 5 |
| Total | 5 | 6 | 3 | 5 | 3 | 0 | 22 | 18 | 11 |

- Have difficulty to understand the purpose of the value texts on top of the bars in with-gridlines-values charts.

**Error rate:** We next analyzed the error rates. For value judgments we allowed some imprecision in the answer for all styles: an answer was incorrect if it corresponded to a value more than half a centimetre above or below the correct value on the tactile layout. The results are shown in Table 2.

We performed a within-subject ANOVA analysis on the total number of errors for different kinds of bar chart (simple, compound, stacked) versus styles of bar chart (direct, grid, grid+value). We found that $F_{kind}(2, 22) = 21.80, p < .01$, $F_{style}(2, 22) = 8.07, p < .01$, providing strong support for a relation between errors and kinds, and errors and styles.

Additionally, we performed a logistic regression analysis and pairwise test on the total number of errors, and found that the results were significant for direct and grid+value pair at a level of $p < .01$, and less significant for grid and grid+value pair at a level of $p = .05$. These results suggest that one would expect for the hypothesis, that there would be less errors in grid+value charts than grid and direct styles. It is surprising that participants made mistakes for grid+value style. The comments suggest that some participants made these mistakes because they could not understand the purpose of the value labels.

The relatively high number of errors for the stacked bar chart kind is significant when compared to other kinds of bar charts. This is also in correlation with the results for sighted people [20]. Here part of the problem seems to be lack of familiarity with stacked bar charts. This meant that some participants did not realize that the value of a sub-bar in a stacked bar is obtained by subtracting the value associated with the bottom of the sub-bar from the top, instead these participants simply gave the value associated with the top of the sub-bar. Another issue was difficulty in distinguishing and remembering the textures associated with the different kinds of sub-bars. Illustrative comments include:

- Stacked bar charts are too complicated, and not sure how to interpret them.
- Stacked bar charts might be separated into more than one chart to be more comprehensible.

**Table 3.** Time spent on charts in Experiment 1: average time (standard deviation)

| Task | Simple | | | Compound | | | Stacked | | |
|---|---|---|---|---|---|---|---|---|---|
| | Direct | Grid | Grid+Value | Direct | Grid | Grid+Value | Direct | Grid | Grid+Value |
| Exploration | 62.83 (27.23) | 54.33 (17.44) | 90.00 (42.47) | 64.33 (45.88) | 84.67 (32.89) | 105.00 (64.16) | 153.83 (109.35) | 94.67 (88.90) | 102.83 (42.02) |
| DoD | 18.33 (10.31) | 24.17 (9.39) | 23.17 (33.47) | 26.50 (9.25) | 32.33 (16.63) | 17.00 (8.22) | 36.50 (15.54) | 37.83 (25.42) | 28.50 (25.56) |
| Search | 16.17 (10.48) | 16.33 (8.14) | 16.83 (13.35) | 21.50 (9.48) | 34.50 (33.77) | 30.83 (23.54) | 30.50 (9.09) | 49.67 (15.51) | 61.67 (33.73) |
| Browse | 22.17 (15.63) | 24.17 (14.27) | 25.67 (8.94) | 17.67 (14.36) | 28.17 (12.24) | 43.50 (45.15) | 61.67 (41.05) | 41.83 (23.07) | 50.50 (21.27) |
| Total | 119.50 (38.07) | 119.00 (37.36) | 155.67 (63.74) | 130.00 (44.91) | 179.67 (48.93) | 196.33 (120.54) | 282.50 (137.83) | 224.00 (101.73) | 243.50 (91.98) |

**Time:** Finally, we analyzed the time spent initially exploring the diagrams and then answering each question for each diagram. The times for the last six participants are given in Table 3.

We performed a within-subject ANOVA analyses for the times spent on different tasks and only found significance for a relation to chart kinds (simple, compound, stacked) in explore, search, browse, and total time except DoD, $F_{explore}(2, 10) = 4.91, p = .03$, $F_{search}(2, 10) = 7.98, p < .01$, $F_{browse}(2, 10) = 8.25, p < .01$, $F_{total}(2, 10) = 5.62, p < .02$. The data suggest that the stacked bar chart was more difficult to read and perform the tasks than the single or compound bar chart.

The results do not support our hypothesis that adding grid line and values would improve performance in terms of time. While not statistically significant, the trend is that adding values and grid lines tend to slow performance. This is similar to the finding in [12] that adding a grid to a tactile line graph does not lead to faster times. We expect that this is because additional elements may clutter the diagram making it slower to read and also because some participants even when given a value still double check this value using the scale on the $y$-axis.

## 4 Experiment 2

One interesting observation from Experiment 1 was that participants who were born blind had a reading strategy which was based on systematically exploring the entire layout so as to memorize all elements in the layout. In contrast, the late-blind participants tended to move their fingers more around the diagrams to find the tactile elements, navigating through the diagram in a way that seemed more similar to that of a sighted reader. This observation suggests that, at least for people who were born blind, it may be useful to provide a layout which facilitates this initial exploration.

Since Braille text is read left-to-right from the top of a page to the bottom, Braille readers are very used to reading in this way. We therefore thought that a blind reader might prefer a layout which facilitates a systematic left-to-right, top down reading of the bar chart. This leads us to hypothesize:

H2 A horizontal layout for a bar chart in which the independent axis of the chart is placed on the vertical axis will be more readable and will be preferred by users to the vertical layout.

The main reason for presenting data in a bar chart rather than a table is that the visual representation allows the reader to immediately "see" the magnitude of data values and trends in the data. We wished to investigate if similar benefits held for blind readers of tactile bar charts. Although the findings in comparison between graph and table representations are inconclusive, it was reported that tables had advantages over graphs for specific tasks [4,21]. Thus, we felt that a tactile table would allow readers to better understand the size of a value and to more readily compare values and to identify trends. Our hypothesis was:

H3 A tactile table will be more readable and will be preferred by users to presenting the same information in a tactile bar chart.

The results from the previous experiment indicated that clutter and difficulty of distinguishing textures are issues in readability of tactile diagrams. This suggests that a tactile/audio presentation which uses an uncluttered tactile presentation without Braille labels but uses audio to provide information provided in the labels and information about the kind of bar and the bar chart value may be preferred by blind people since it will provide the benefits of the *grid+value* style with a less cluttered diagram. Another benefit of a tactile/audio presentation is that we can provide an audio overview explaining the kind of bar chart and number of values in it at the top of the chart to facilitate understanding (as suggested by participants in Experiment 1). For completeness we also wished to compare with an audio only presentation of the data presented as a table. Our hypotheses were that:

H4 A tactile/audio presentation of a bar chart will be more readable and will be preferred by users to a tactile presentation.

H5 A tactile/audio and a tactile presentation of a bar chart will be more readable and will be preferred by users to an audio only presentation.

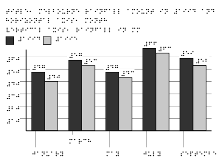
We used the Dell Latitude XT which has a capacitive touch-screen to provide the tactile/audio presentation where audio would be initiated naturally when feeling the tactile diagram rather than requiring explicit finger pressing.

**Participants:** 6 participants, 1 born blind and 5 late blind, between the ages of 25 to 50 completed the experiment. They all had previously read a bar chart and all had participated in the previous experiment, so had substantial recent experience in reading tactile bar charts. They also had some experience using JAWS [1] to read tables.

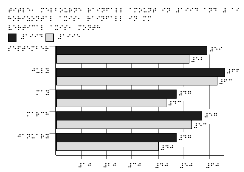
**Materials and design:** The experiment had two sub-experiments: Part A investigated Hypotheses H2 and H3, and Part B investigated Hypotheses H4 and H5.

The material for Part A was three different tactile versions for three different data sets each with two independent variables and one dependent variable and the same number of elements. The first version, *vertical*, was a compound bar chart with vertical bars, the second, *horizontal*, was a compound bar chart with horizontal bars, while the third, *table*, was a tactile table. In the table, rows

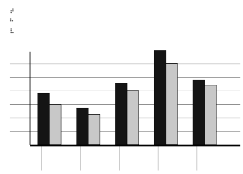
[1] www.freedomscientific.com



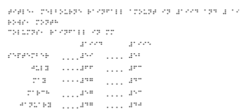
(a) Vertical



(b) Horizontal



(c) Tactile/audio



(d) Table

**Fig. 3.** Tactile bar charts in Experiment 2

and columns were aligned horizontally, and vertically, respectively. There were also three dots between each label on rows to guide the participants. Both bar charts contained grid lines and bar values reflecting the results of Experiment 1. Examples of the three versions are shown in Figure 3 (a, b, c). For each of the data sets we created three questions similar to those used in Experiment 1 with the compound bar chart.

In order to remove the possible confounding factor of presentation order and the exact choice of data and question we created three versions of the experimental material in the same experimental design as for the last six participants in Experiment 1.

The materials for part B were three different media constructed from three data sets each with two independent variables and one dependent variable and the same number of elements. For each data set we created three questions similar to those used in Experiment 1 and used the same counterbalanced experimental design as Part A.

The first presentation medium, *tactile*, was a tactile compound bar chart with vertical bars, grid lines and Braille values on the bars. The second medium, *tactile/audio*, was a tactile compound bar chart overlaid on the Dell Latitude XT tablet PC. At the top of this chart there were Braille labels 'C' for chart title, 'H' for horizontal axis title, and 'V' for vertical axis title. The axes had no Braille values but just the tick marks. The values of each bar were spoken when the users touched at the top border of the bars. Additionally, when the users were on the bars of different series, the series name and the independent value were spoken. An example is shown in Figure 3 (d). The third medium, *audio*, was a computer mediated audio presentation of a table containing the data. We used JAWS without script extensions to read and navigate a HTML table since most participants would be familiar with this program.

We implemented the software for the Dell Latitude XT that used its multi-touch feature to map the points touched on the screen to the graphic elements on the tactile layout. Both TTT and IVEO use a *resistive* touch-screen which supports an interaction style in which the user must explicitly press on part of the diagram to obtain audio feedback. Latitude XT is equipped with NTrig DuoSense dual-mode digitizer which supports both pen and touch input using *capacitive* sensors that are based on proximity. Perhaps surprisingly, we have found that a capacitive touch-screen can be overlaid with a tactile diagram and still sense the position of fingers when the user touches the diagram.[2] While capacitive touch-screens can support the *press* mode of interaction, they also allow a novel *touch* mode of interaction in which audio feedback is more continuous and based on finger position. As part of our user study the software provided the *touch* interaction mode where the computer read the label of a tactile element as long as the user touched the element. When the user lifted his/her finger up or moved out of the element, the computer stopped reading the label.

We used Windows 7 Build 7000 Beta [3] operating system which was the only operating system that supported the use of multi-touch features of the digitizer. For speech synthesising, we used the FreeTTS library [22].

**Procedure:** Experiments were performed in a small room either at Vision Australia or at Monash University. The procedure was similar to Experiment 1.

First participants were asked to sign a consent form which had previously been sent by email to them and which they were given a tactile version on the day. This also provided a short description of how the experiment would be conducted and what type of information would be collected. They were then asked how long they had been blind.

Then the following process was repeated for Part A and Part B. In each part the participants were shown one chart at a time for that part, and asked to explore the chart and let us know when they were ready to answer the questions about it. They were then asked to answer the three questions for that chart. The time taken to explore and answer each question were recorded as were their answers. After reading and answering questions for a chart, the participants were asked to rank the presentation from the least preferred to the most preferred. During this process, participants were invited to give comments and explain the features that influenced their rankings.

### 4.1 Data Analysis and Results

**Part A:** The user preferences for Part A are shown in Table 4. We found that most participants ranked the table version as their most preferred chart. They reported that it was because the table version was easy to understand, and had only the necessary information on it.

We used a multinomial distribution to calculate the significance levels for different versions. For selecting a particular version as the first, second, or third preference all significance levels were $P(Y \geq 4|n = 6, p = .33) = .3$. Hence, there was some support for Hypothesis H3, but not for H2. On the contrary, the participants preferred the vertical version to the horizontal version. We conjecture

[2] This is surprising since capacitive touch-screens rely on the naked human finger acting as a capacitor. We have realized that the Braille paper has still some degree of conductivity.

[3] http://www.microsoft.com/windows/windows-7/

**Table 4.** Preferences in Experiment 2: first choice count (second choice count)

(a) Versions

| Version of chart | Preferences |
|---|---|
| Vertical | 1 (4) |
| Horizontal | 1 (1) |
| Table | 4 (1) |

(b) Media

| Presentation medium | Preferences |
|---|---|
| Tactile | 4 (1) |
| Tactile/Audio | 1 (4) |
| Audio | 1 (1) |

**Table 5.** Time spent on charts in Experiment 2: average time (standard deviation)

| Task | Part A | | | Part B | | |
|---|---|---|---|---|---|---|
| | Vertical | Horizontal | Table | Tactile | Tactile/Audio | Audio |
| Exploration | 133.17 (91.86) | 139.50 (83.28) | 66.50 (46.92) | 66.67 (18.46) | 153.67 (99.48) | 134.67 (139.41) |
| DoD | 18.83 (3.71) | 31.83 (34.88) | 11.50 (7.23) | 15.00 (7.95) | 34.20 (26.35) | 12.33 (4.59) |
| Search | 31.00 (26.33) | 28.67 (14.24) | 22.83 (24.83) | 17.50 (16.75) | 38.60 (31.86) | 33.67 (16.79) |
| Browse | 24.00 (7.24) | 22.50 (18.43) | 22.33 (14.77) | 9.67 (5.20) | 26.00 (18.10) | 20.17 (3.76) |
| Total | 207.00 (100.26) | 222.50 (136.70) | 123.17 (71.91) | 108.83 (31.99) | 263.20 (156.55) | 165.67 (56.29) |

that the low ranking for the horizontal version was probably the result of participant's unfamiliarity with horizontal layout. Some of the participants reported that they could give faster answers if they had more experience on horizontal layout version.

Representative comments were

- I am not sure how to read this type of chart (HORIZONTAL), but I guess I can still understand it.
- Table is easy to understand.

Because both the vertical and horizontal versions provided gridlines and values we did not expect many errors. This was true: the only two user errors were by one participant for the horizontal version (probably as a result of unfamiliarity.) Thus, all three presentation media led to very few errors.

However there did appear to be differences in time taken with the different versions. As seen from Table 5, performance was faster with the table version than with either bar chart versions because the initial exploration took less time. We performed a within-subject ANOVA test and found that exploration and total time values were significant, $F_{explore}(2, 10) = 9.95, p < .01$, and $F_{total}(2, 10) = 7.10, p = .01$. This provides some further support for Hypothesis H3. There was not a significant difference in timing between the horizontal and vertical bar chart versions, but the trend was the horizontal versions took slightly more time.

**Part B:** The user preferences for Part B are shown in Table 4. We used multinomial distribution to calculate the significance levels for different media. For selecting a particular medium as the first, second, or third preference all significance levels were $P(Y \geq 4|n = 6, p = .33) = .3$. Since most of the participants preferred the tactile medium, we did not have any support for Hypothesis H4. On the other hand, they preferred the audio medium the least which provides weak support for Hypothesis H5.

It is interesting to see that almost all participants preferred the tactile medium to the tactile/audio. We believe this may have been for two reasons.

First was lack of familiarity with tactile/audio medium. Indeed we note that the one participant who was familiar with tactile/audio medium (they had previously used the IVEO system) was the only participant who ranked tactile/audio first.

The second reason was a lack of robustness of the software while presenting the tactile/audio medium. Although the software handles multiple touch inputs, it is not always stable due to the beta versions of both the operating system and the digitizer driver which occasionally stopped responding when a participant put his hands on the screen. To avoid this problem we asked participants to avoid putting their both hands on the screen as much as possible and try to use only one finger for interaction.

We observed that for the tactile medium, the participants first read the Braille texts before exploring other tactile elements on the layout. Participants reported that the most important advantage of the tactile chart was that all the information were on the layout, and unlike the tactile/audio and audio media they did not need to make guesses and try to find the "hidden" information.

The participants tried to do the same kind of exploration on tactile/audio diagrams. They got confused at first since there was no Braille text on the layout, but only tactile elements whose information would be spoken by the speech synthesizer. When they realized this, they started exploring the charts starting from the top left to left bottom, and then to right bottom where the chart title, axis titles, dependent, and independent values were spoken by the computer. They spent more time around the title and the axis areas than the other parts of the diagram. Most of the positive comments about the tactile/audio presentation were that it allowed them to feel a physical material and also gave accurate information which eliminated guessing without introducing any tactile clutter on the layout.

Since the audio diagram was read by JAWS, it was difficult to understand for users who had not used JAWS frequently. The main issue was the navigation within the audio description. They used the arrow keys keyboard, but since JAWS has a number of additional shortcuts for reading rows, columns individually, experienced participants found it easier to navigate the table data.

Participants made few errors with any of the presentation media. One participant made one error with the tactile medium and two with the tactile/audio.

Table 5 suggests that participants were faster with the tactile presentation medium than with either the audio or tactile/audio. This may be a question of familiarity. We performed a within-subject ANOVA analysis on time data and found a weak significance in total time $F_{total}(2, 10) = 3.43, p = .07$.

## 5 Conclusion

The results of our first experiment suggest that it may be useful to provide both grid lines and values on tactile only presentation of bar charts since this will generally be preferred by participants and will significantly reduce the number of reading errors although it may increase reading time. It also indicates that many blind people find it difficult to read tactile only presentations of stacked bar charts, suggesting that either training in their use is required or that the data is better presented as compound bar charts.

The results of our second experiment provided some support that tactile and tactile/audio presentation of multi-dimensional data is preferred to audio presentation. However, we note that we only used JAWS–other kinds of audio presentation may be preferred. Our experiments did not provide support for the use of tactile/audio presentation of diagrams rather than the more traditional tactile presentation. We believe that this may have been due to unfamiliarity and lack of robustness in the implementation. We also note that tactile/audio has clear benefits for users who are not proficient in reading Braille. Our results emphasize the importance of appropriate training in tactile and tactile/audio diagram understanding.

We also found that tactile table presentation was preferred and gave performance benefits over a tactile bar chart presentation. This implies that participants did not significantly gain from the use of a bar to indicate the magnitude of a data value or to sense trends.

On the other hand, participants did not like an audio presentation. This suggests that they liked the tactile and tactile/audio presentation because it provided a 2-D index into the data in the bar chart. This is supported by the way in which participants performed the initial exploration of the chart. The first task performed on a tactile chart is to read all the Braille text as to memorize all the information on the layout. The tactile elements on the layout are then used as a reference point to recall the values, or at least give the position of the relevant information.

**Acknowledgement.** We thank Vision Australia transcription department, especially Kathryn Randall, for their support in the user studies, and the participants for their effort and great feedback. We also thank Peter Moulder for performing an initial statistical analysis. Finally, we acknowledge the support of the ARC through Discovery Project Grant DP0987168.

## References

1. Challis, B., Edwards, A.: Design Principles for Tactile Interaction. In: Brewster, S., Murray-Smith, R. (eds.) Haptic HCI 2000. LNCS, vol. 2058, pp. 17–24. Springer, Heidelberg (2001)
2. Round Table on Information Access for People with Print Disabilities Inc: Guidelines on conveying visual information (2005)
3. The N.S.W. Tactual and Bold Print Mapping Committee: A guide for the production of tactual and bold print maps (2006)
4. Meyer, J., Shinar, D., Leiser, D.: Multiple factors that determine performance with tables and graphs. Human Factors 39(2) (1997)
5. Gardner, J., Bulatov, V.: Scientific Diagrams Made Easy with IVEO. In: Miesenberger, K., Klaus, J., Zagler, W.L., Karshmer, A.I. (eds.) ICCHP 2006. LNCS, vol. 4061, pp. 1243–1250. Springer, Heidelberg (2006)

6. Landau, S., Gourgey, K.: Development of a talking tactile tablet. Information Technology and Disabilities 7(2), 4 (2001)
7. Bentzen, B., Peck, A.: Factors Affecting Traceability of Lines for Tactile Graphics. Journal of Visual Impairment and Blindness 73(7), 264–269 (1979)
8. Lambert, L., Lederman, S.: An Evaluation of the Legibility and Meaningfuiness of Potential Map Symbols. Journal of Visual Impairment & Blindness 397 (1989)
9. Millar, S.: The perception of complex patterns by touch. Perception 14(3), 293 (1985)
10. Lederman, S., Kinch, D.: Texture in Tactual Maps and Graphics for the Visually Handicapped. Journal of Visual Impairment and Blindness 73(6), 217–227 (1979)
11. Lederman, S., Campbell, J.: Tangible Line Graphs: An Evaluation and Some Systematic Strategies for Exploration. Journal of Visual Impairment and Blindness 77(3), 108–112 (1983)
12. Aldrich, F.K., Parkin, A.J.: Tangible line graphs: an experimental investigation of three formats using capsule paper. Human Factors 29(3), 301–309 (1987)
13. Yu, W., Brewster, S.: Comparing two haptic interfaces for multimodal graph rendering. In: Proceedings of Haptic Interfaces for Virtual Environment and Teleoperator Systems, HAPTICS 2002, pp. 3–9 (2002)
14. Yu, W., Brewster, S.: Evaluation of multimodal graphs for blind people. Universal Access in the Information Society 2(2), 105–124 (2003)
15. Wall, S., Brewster, S.: Feeling what you hear: tactile feedback for navigation of audio graphs. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1123–1132. ACM, New York (2006)
16. Landau, S., Russell, M., Gourgey, K., Erin, J., Cowan, J.: Use of the Talking Tactile Tablet in mathematics testing. Journal of Visual Impairment and Blindness 97(2), 85–96 (2003)
17. Hatwell, Y.: Images and non-visual spatial representations in the blind. John Libbey Eurotext, 13–35 (1993)
18. Goncu, C., Marriott, K.: Tactile chart generation tool. In: Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS 2008, pp. 255–256 (2008)
19. Card, S., Mackinlay, J., Shneiderman, B.: Readings in information visualization: using vision to think. Morgan Kaufmann Publishers Inc., San Francisco (1999)
20. Cleveland, W., McGill, R.: Graphical perception: Theory, experimentation, and application to the development of graphical methods. Journal of the American Statistical Association 79(387), 531–554 (1984)
21. Kosslyn, S.: Graph design for the eye and mind, pp. 30–36. Oxford University Press, USA (2006)
22. Walker, W., Kwok, P., Lamere, P.: FreeTTS Open Source Speech Synthesis. Sun Microsystems Laboratories, Speech Integration Group (2002)

# Diagram Editing on Interactive Displays Using Multi-touch and Pen Gestures

Mathias Frisch, Jens Heydekorn, and Raimund Dachselt

User Interface & Software Engineering Group
Otto-von-Guericke-University Magdeburg,
39106 Magdeburg, Germany
{mfrisch,jheyde,dachselt}@isg.cs.uni-magdeburg.de

**Abstract.** Creating and editing graphs and node-link diagrams by means of digital tools are crucial activities in domains such as software or business process modeling. However, these tools have several drawbacks with regard to interaction techniques and usability. In order to address these issues, we investigate the promising combination of pen and multi-touch input on interactive displays. In this work, we contribute a gesture set to make the interaction with diagrams more efficient and effective by means of pen and hand gestures. Thereby, two prevalent mental models are supported: *structural editing* and *sketching*. The gesture set is based on the results of a previous pilot study asking users for suggestions to accomplish diagram editing tasks on tabletops. In this paper, we provide a careful analysis of the resulting user-elicited gestures. We propose solutions to resolve ambiguities within this gesture collection and discuss design decisions for a comprehensible diagram editor. We also present the multi-touch and pen gesture set as a fully implemented prototype for diagram editing on interactive surfaces.

**Keywords:** interaction techniques for diagrams, user-centered approach, mental models, multi-touch and pen interaction, digital pen, sketching, tabletops.

## 1 Introduction

Current research in the field of creating diagrams is often focused on methods and algorithms to produce and to process diagrams in an automatic way. This comprises, for example, the generation and transformation of diagrams or automatic layout algorithms. However, creating graph and node-link diagrams from scratch and editing them manually in digital editors are very common activities in domains such as software modeling, business process modeling, project management and simulation.

With regard to interaction techniques, there basically exist two approaches to create and edit graphical node-link diagrams. There are digital editors to build diagrams, commonly based on formal notations, by means of *structural editing*. They are usually limited to traditional point and click interaction techniques, e.g. dragging and dropping diagram elements from a toolbar or switching modes by means of buttons or menus. In domains such as software development these editing tools are often perceived as constrictive and inflexible [8], [15].

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 182–196, 2010.

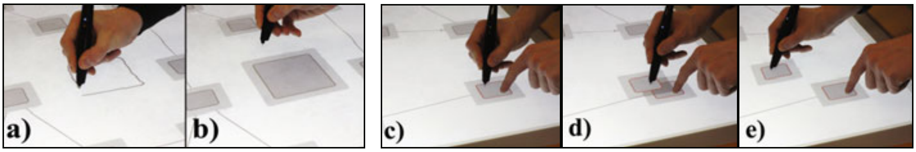



**Fig. 1.** Creating a node by sketching (a), copying an already exiting node by means of a bimanual hold and drag gesture: hold and drag have to start from the interior of the node (b). When the pen is dragged, the copy appears (c) and can be rearranged (d).

This is one reason why freehand *sketching* is often preferred. Beyond that, drawing with pens is a more natural human capability. Therefore, in many situations diagrams are drafted on whiteboards or flip charts [6]. With these traditional setups, diagrams can be produced in an informal and ad hoc way. As a consequence, they often have to be remodeled in digital tools, which is a time consuming process. Digital sketching tools try to solve this problem and additionally offer techniques such as rearranging, grouping or scaling elements on electronic whiteboards or Tablet PCs [8], [15]. Usually, these kinds of applications solely support pen or single touch interaction. Hence, for tasks such as zooming or changing types of elements users still have to navigate through menus or click buttons, which disrupts the editing process.

In contrast to that, we are investigating the combination of multi-touch and pen interaction for the domain of diagram editing on interactive displays. We expect that this approach is able to make the handling of diagrams more efficient and effective. Certain tasks such as rearranging or deleting elements can be accomplished simultaneously by using both hands. Beyond that, it is conceivable to switch modes by means of hand gestures without interrupting the current workflow (see Figure 1 c-e). Moreover, gestures with a physical or metaphorical nature can make the interaction more natural, which is important especially for novices.

Devices with pen and multi-touch support occur in different form factors, reaching from handheld tablets to huge wall size displays. Some of them are commercially available currently [22], and it can be expected that they will be increasingly applied.

In order to start investigating how node-link diagrams can be edited with touch and pen interaction on interactive displays, we conducted a pilot study [14]. The study applied a user-centered design approach. As a result, a collection of user-defined gestures was identified. However, the collection included several ambiguities, because in some cases the participants assigned the same gestures to different tasks.

At the heart of this work, we solve these ambiguities and contribute a successfully implemented gesture set which is based on this collection. The set allows the combination of pen and hand input and supports both approaches mentioned above – *structural editing* and *sketching*. The paper is structured as follows: After a brief summary of the design and the results of the pilot study, we present our methodology how to resolve the conflicts within the collection of gestures. We determine design goals, analyze the elicited gestures and propose options which can be applied to resolve the ambiguities. After that, we present the implementation of the gesture set in detail and give an insight into the architecture of the prototype. Finally, we discuss how the prototype can be extended to a diagram editor for more complex diagram notations and give an outline of future work.

# 2 Related Work

## 2.1 Digital Diagram Sketching

Various tools have been realized for sketching on electronic whiteboards or Tablet PCs. In [23], several domain-independent pen interaction techniques are presented. They cover copying, pasting or scaling elements, and the respective prototype is also capable of recognizing node-link diagrams. Beyond that, there are digital sketching tools such as presented in [8], [15], [17] or [5]. They are tailored to the domain of software engineering and convert sketches to diagram notations such as UML. In contrast to the prototype presented in this paper, they solely support pen interaction and do not consider additional modalities. Especially in the domain of software development, several studies have been conducted [8], [6], [9]. They investigate how and for what purposes software designers use whiteboards for diagram sketching. As a result, design principles for digital sketch applications were concluded. However, to our knowledge, the combination of multi-touch and pen interaction has not been studied yet in the domain of diagram sketching.

## 2.2 Multi-touch Gestures for Tabletops

Over the past years, various multi-touch enabled interactive displays have been developed. Some of these devices are already commercially available. They are applying different approaches to detect touch events from users, such as computer vision [18], [22] or capacitive technology [10], [25]. These devices have been used to investigate and to propose a number of multi-touch gestures for particular purposes.

Gestures proposed in [26] cover panning, scaling, rotating, and picking up objects. In [30] a set of gestures for multi-user tabletops is presented. It includes gestures for rotating, collecting objects, and for private viewing. Wu et al. [30] describe design principles for gesture design and built a prototype of a publishing application to illustrate the usage of their principles. Other research on gestures can be found in [20], [26] or [4]. Amongst others, they are including whole-hand gestures on interactive surfaces. Concerning the interaction with node-link diagrams, Dwyer et al. [11] conducted a study to investigate how users would layout graphs on tabletops in a manual way. They observed several gestures participants applied to reposition and group nodes. However, pen interaction and the combination of pen and touch were not considered. Besides that, the given tasks were limited to layout purposes and they do not propose a particular gesture set for a respective diagram editor application. Our contribution instead covers diagram editing tasks, such as creating, deleting or copying of nodes and edges.

## 2.3 Combination of Pen and Touch Interaction

In addition to the aforementioned devices, there are technical systems which explicitly support multi-touch and pen input simultaneously. Flux [19] is a tabletop of this type and can be tilted to horizontal, vertical and slanted positions. For the prototype presented here we are using a similar technical approach which will be discussed in Section 6. Another vertical-only solution is INTOI [3], including the capability of pen and hand gesture recognition.

Concerning interaction techniques, the combination of pen input and single touch is investigated by Yee [31]. He proposes panning the canvas with the finger while drawing with the pen. Beyond that, the usage of digital pens and multi-touch on tabletops has been studied in Brandl et al. [2]. They suggest general design principles and present interaction techniques for a graphics application. Both works consider Guiard's Kinematic Chain Model [16], which proposes principles for the division of roles between hands: the dominant hand moves within the frame of reference set by the non-dominant hand, the non-dominant hand precedes the dominant hand, and the dominant hand performs more precise actions.

### 2.4 Research on User-Defined Gestures

All of the gesture sets mentioned above are designed by experts. In contrast to that, there are approaches to elicit gestures from users. For that, Nielsen et al. [24] propose a procedure of four steps. Initially, the system functions should be found, which the gestures will have to communicate. Then participants are asked to perform spontaneous gestures for each of the functions. This is done under video surveillance. As a next step, the video material is evaluated to extract the gesture vocabulary. Finally, the elicited gestures are benchmarked. In their work, Epps et al. [12], Micire et al. [21] and Wobbrock et al. [28] used this approach for their studies. The latter conducted a study to develop a user-defined set of general one-hand and two-hand gestures and presented a respective gesture taxonomy.

The gesture set introduced in this paper is the result of our study with similar design [14]. In contrast to the aforementioned approached, we studied both, multi-touch and pen interaction for the domain of diagram editing. A short summary of our study is given in the following section.

## 3 Eliciting Gestures for Diagram Editing

We conducted a pilot study applying a user participatory approach based on the work of Nielsen et al. [24] (see Section 2.4) to investigate how users would edit node-link diagrams on an interactive tabletop display [14]. To our knowledge, this was the first work which applied this approach to the domain of diagram editing. Particular editing tasks were given to the participants, and they were asked to perform spontaneous hand and pen gestures to solve these tasks. From this, we were capable to get an insight into preferred modalities and prevalent mental models. However, when applying such an approach, one should bear in mind that users are not interaction designers. Therefore, our goal was not to come up with a final user-defined gesture set which can be implemented in a straightforward way. Our main purpose was rather to involve the users in the design process right from the beginning and to get a feeling for preferred interaction procedures in this particular domain.

The result of the study was a collection of pen and hand gestures elicited from users. The gesture set presented in this paper (see Table 1) is based on this collection. Furthermore, we made several additional observations concerning participants' remarks and behavior. These results served as a starting point for designing and implementing the gesture set described later in this paper.

### 3.1 Design of the Pilot Study

The pilot study applied a within-subjects design. Every participant was asked to complete 14 basic editing tasks in a fixed order (see left column of Table 1 for a condensed overview). Besides elementary tasks like creating and deleting elements, we also asked for more specific procedures such as copying a sub-graph (see task 8 in Table 1) and changing a solid edge to a dashed one (see task 9 in Table 1). In order to produce results which are applicable to a variety of visual diagram languages, we used an elementary variant of node-link diagrams. Nodes were represented by simple rectangles and connected by directed or undirected edges.

### 3.2 Participants and Apparatus

Seventeen participants took part in the study. All of them had a solid background in software engineering. Therefore, they were familiar with visual node-link diagrams such as UML. They were neither expert users of visual diagram editors nor UI experts. The study was conducted on a tabletop display which combines multi-touch with pen interaction (see Section 6, Figure 4). Users' inputs were recorded by video camera, by the vision system of the tabletop and by taking notes during the procedure.

### 3.3 Tasks and Procedure

The display was horizontally divided into two areas. The lower area displayed a diagram in the original state, and the upper area showed its final state. For each task, the participants were asked to transfer the diagram from the original state to the final state by performing a spontaneous gesture inside the lower area. Thereby, they got no feedback from the system. Participants performed gestures with three different interaction techniques per task: with one hand (whereby they could use all fingers of the hand), with two hands, and with pen (held in their dominant hand). For the latter, it was free to them to combine the pen gesture with all fingers of the non-dominant hand. Furthermore, each subject was asked to start with the variant that he or she considered as most suitable for the respective task. After each task, the participants were asked to fill out a questionnaire concerning the suitability of each interaction technique.

### 3.4 Results

**Gestures.** We analyzed a total number of 658 gestures. A result of this first analysis was that no task was solved by a single or unique gesture among all participants. The absolute number of variations was highest for the *copy* task (33 variations), which can be certainly attributed to its rather abstract nature. The lowest amount of variations was elicited for the *select node* task (13 variations). Overall, we observed that in general one-hand and pen modality was preferred for solving most of the tasks. However, there were also situations where bimanual interaction was preferred, such as zooming and scaling, copying elements or achieving a mode switch by resting the non-dominant hand on the background.

**Mental Models.** We identified two classes of basic mental models the participants relied on while they solved the given tasks: *sketching* and *structural editing*. The *sketching* class is characterized by physical imitation of real-world ad hoc sketching such as drawing a node or edge. Beyond that, gestures with a metaphorical nature fall into this class as well, e.g. deleting an element by wiping (task 5 in Table 1) or creating a dashed edge by performing a "rake" gesture (see task 9 in Table 1). However, the *sketching* class is not stringently associated with the usage of a pen. In contrast, the *structural editing* class is oriented more towards digital diagram editors and applies a higher grade of abstraction to phrase the intention. The associated gestures, e.g. for copying elements, therefore have a more abstract nature.

**Gesture Collection.** We used these results and observations from the study to identify the top candidates for each task and created a respective collection of pen and multi-touch gestures. Where appropriate, we assigned more than just one gesture to a task. For example, we considered both prevalent mental models mentioned above. In order to support both approaches, gestures based on *sketching* as well as gestures based on *structural editing* were assigned to respective tasks.

## 4 Expert Analysis of the Elicited Gestures

In Table 1 our proposed pen and hand gesture set for node-link diagram editing is depicted. It is based on the elicited collection from the pilot study described above. Where possible, it still supports both approaches – *sketching* and *structural editing*. In Table 1 gestures which correspond to the *sketching* approach have a gray background. However, the proposed set comprises several ambiguities and conflicts, as in some cases the same gesture was assigned to different tasks by the participants. This is no surprise, because we did not explicitly ask them to be consistent and to apply a particular gesture just once. Nevertheless, this prevents a straightforward implementation of the gestures. Therefore, preliminary considerations and a prior analysis by experts are necessary. In the following sub-sections we describe our methodology to resolve these ambiguities and to design a system which realizes the gesture set.

### 4.1 Design Goals

Before conducting a deeper analysis of the existing conflicts, we developed general design goals for gestural diagram editing. They are based on the results and the observations from the study to preserve the user-centered approach:

- G1 Preserving support for both mental models – *sketching* and *structural editing* – as they are essential for a satisfying system for most users in this domain.
- G2 Keeping the introduction of new special gestures to a minimum by reusing proposed identical gestures for different tasks [30].
- G3 Providing ad hoc and direct creation of content without time-consuming navigation through options offered by menus (as suggested in [8]).

### 4.2 Resolving Conflicts

In particular, the following conflicts occur within the set of elicited gestures, because users assigned identical gestures to the same task:

C1 Creating an edge vs. moving a node: When a node is touched and the finger or pen starts moving, it is not clear if the node should be dragged or an edge should be inserted.

C2 Selecting nodes vs. creating an edge by tapping: When several nodes are sequentially tapped, it is ambiguous if they should be selected or if edges should be created between them.

C3 Drawing an edge vs. dragging an edge: Within the *create edge* task it is not obvious if an edge should be dragged to the target node (like a rubber band) or if the edge is meant to be sketched like on a whiteboard.

In some cases during the pilot study, participants realized they were suggesting conflicting gestures and indicated how they would resolve them. We took these suggestions and identified four general options to resolve the aforementioned ambiguities with regard to the design goals G1 - G3. These are general solutions and can also be applied in the design process of other systems.

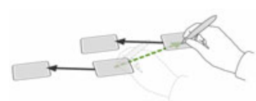
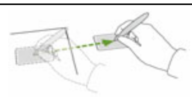
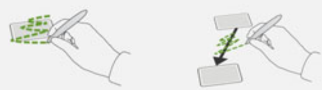
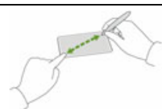
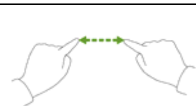
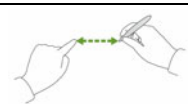
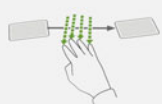
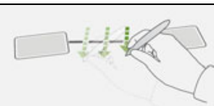
**Additional gestures.** A trivial solution for all conflicts is the definition of additional gestures. In that way, for each task a specific gesture can be assigned. However, this approach would contradict design goal G2 and would require users' effort for memorizing multiple unique gestures.

**Mode switch.** Similarly to WIMP (Windows, Icons, Menus, Pointer), static menus and variations of buttons could be provided for mode switches. However, they can be difficult to reach on large surfaces, especially if they are placed on traditional places such as the top border of the display. The usage of context menus is more appropriate. However, this is contrary to design goal G3 and an additional gesture for invocation would be necessary. Furthermore, the non-dominant hand can be placed on the background to cause a mode switch while the actual gesture is performed with the dominant hand. However, this is only applicable for one-hand gestures and there must be free and reachable background.

**Distinction of input modalities.** Another way to resolve the conflicts is to distinguish between different input modalities - in our case between touch and pen. This would mean, for example, that drawing elements is solely assigned to the pen, whereas functions such as dragging or scaling elements can only be done with fingers. However, we did not identify a general preference for one modality and would therefore not force users to switch between hand and pen interaction.

**Graphical contexts.** The user interface can offer additional graphical regions which are sensitive to gestures. These regions can represent different contexts. For example, the border of a node is explicitly touchable to create edges, whereby the interior of the node is used for dragging or selecting. This approach can solve the conflicts C1 and C2.

**Table 1.** The gesture set for diagram editing based on the user-elicited gesture collection [14]. It considers structural editing (white background) and sketching (gray background). All gestures depicted with pen can also be performed with a single finger.

| Task | Gesture |
|---|---|
| 1. Create Node |  single tap;  copy node by holding and dragging;  drawing node |
| 2. Create Edge |  „dragging" edge;  sequential tapping;  drawing (un)directed edge |
| 3. Select Node(s) |  single tap;  encircle node(s) |
| 4. Move Node(s) |   |
| 5. Delete Node or Edge |  dragging to off-screen;  wiping |
| 6. Scale Node |    |
| 7. Zoom Diagram |   |
| 8. Copy Sub-graph |  hold and drag (after select nodes) |
| 9. Change Edge from Solid to Dashed |  „rake"-gesture;  sequential crossing |

# 5 The Gesture Set

We successfully implemented the proposed gesture set depicted in Table 1. In order to resolve the conflicts identified in Section 4, the set of interactive elements has been extended. We decided to add an interactive border region around every node, which serves as an additional graphical context (e.g. see Figure 2). It can be used to insert edges by tapping or drawing. Every time it is touched with the pen or fingers, the border region changes its color to give a visual feedback. In that way, all gestures can be performed by means of finger or pen and in an ad hoc way – a prior mode switch is not necessary. Details are explained in the following subsections, where we present the gesture set in the way it is implemented.

## 5.1 Creating Diagram Elements

**Creating Nodes.** Nodes can be created in two ways: by sketching their outlines (see Figure 1 a, b) or by tapping with the finger or pen on the background. The latter corresponds to *structural editing*, and a standard-sized node is created at the place where the tap occurs. In order to avoid that nodes are created by unintended touches, we propose that the touches have to be held for a short time delay until the node appears. Beyond that, nodes can be created by copying already existing nodes. This is done through a bimanual gesture. The non-dominant hand holds the node and the copy is dragged from it with a finger of the dominant hand or the pen. Both parts of this gesture - hold and drag - have to start from the interior of the node (see Figure 1 c-e).

**Creating Edges.** We implemented three solutions to create an edge between two nodes: sketching, dragging and tapping.

*Sketching.* Edges can be sketched by starting a drag gesture from the interactive border of a node (see Figure 2 a). Thereby, drawing a simple line results in an undirected edge and drawing a line with arrow head results in a directed edge.

*Dragging.* The second method is to drag edges in a rubber band style, like in structural editors (see Figure 2 d-f). For that, the same bimanual gesture as for the copy task is applied: holding the node with one finger and dragging the edge with another one. In contrast to the copy task, the dragging gesture has to start from the interactive border region of the node. This modifies the proposed gestures of Table 1 just slightly, but solves C1 by means of the additional graphical context.

*Tapping.* Edges can also be created by tapping. In particular, there are two techniques: *sequential tapping* and *holding & tapping*. With *sequential tapping* an edge is created between nodes when their border regions are tapped sequentially. By means of *holding & tapping* the border region of a node can be held with a finger of the non-dominant hand. Tapping borders of other nodes with a finger of the dominant hand or the pen results in an edge (see Figure 2 b-c). If many edges are going from the same node, this bimanual gesture can serve to create them in a fast way.

In general, these tapping techniques are beneficial for connecting nodes on rather large interactive surfaces, such as tabletops or wall-sized displays, as dragging edges between nodes located far away from each other can be cumbersome. Of course, the nodes have to be in reach of both arms.

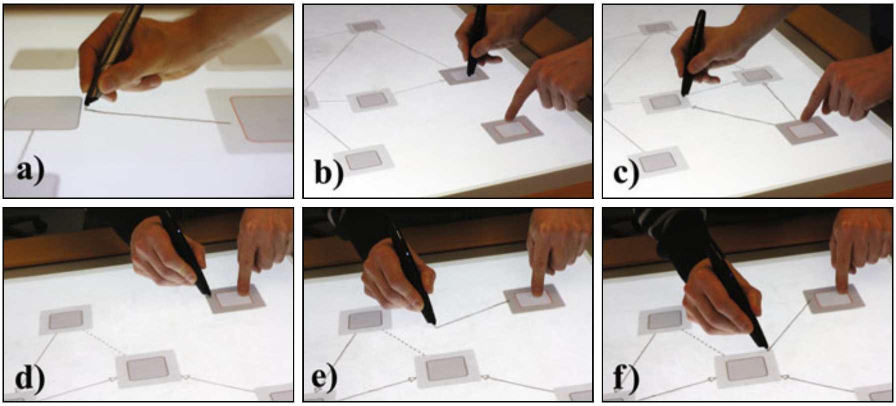



**Fig. 2.** Creation of an edge by sketching, sketching has to start on the interactive border region (a), creation of an edge by *holding & tapping*: the border of a node is held (b), tapping the border of other nodes creates edges (c), dragging edge by *holding & dragging*: dragging has to start from the border region (d), edge can be dragged like a rubber band (e, f).

### 5.2 Selecting and Moving Nodes

Nodes can be selected by tapping their interior or by encircling them. The latter rather corresponds to the sketching approach. If more than two nodes are selected they are aggregated to a sub-graph (see Figure 3 a, b). The aggregation can be undone by shortly tapping the background. Moving single nodes and sub-graphs is possible by touching their interior and dragging them to the appropriate position by means of finger or pen. It is also possible to copy sub-graphs by applying the aforementioned copy-gesture for nodes (*holding & dragging*). As a result, all nodes within the sub-graph and edges between nodes of the sub-graph are copied. Edges going from and to the sub-graph are not present in the copied sub-graph.

### 5.3 Deleting Diagram Elements

All kinds of diagram elements can be deleted by performing a wipe gesture, like on whiteboards (see Figure 3 e). Thereby, the gesture has to start on the background and all elements intersecting its bounding rectangle are deleted when the finger or pen is lifted. For deleting single nodes or sub-graphs it is also possible to drag them to off screen; an approach which we observed several times during the pilot study. In our current implementation nodes can be dragged “from the screen” in all four directions. When nodes are dragged out of the visible part of the workspace, the nodes (and all their associated edges) are deleted when the finger or pen is lifted (see Figure 3 c, d).

### 5.4 Changing the Type of an Edge

We implemented two different gestures of metaphorical nature to change solid edges to dashed ones. The task can be performed with a multi-touch “rake” gesture. Thereby, three or four fingers are crossing the edge in parallel. After lifting the fingers, the appearance of the edge is changed (see Figure 3 f-h). Besides that, it is also possible to change the edge by crossing it three times sequentially. In this way users are also able to perform the task with one finger or pen. In order to change a dashed edge back to a solid one, the "rake" gesture can be performed again or an edge can be created on top of the existing one. This gesture is an example for a shortcut gesture which is applied to perform a domain-specific diagram editing task. For more suggestions concerning that type of gesture, please see Section 7.

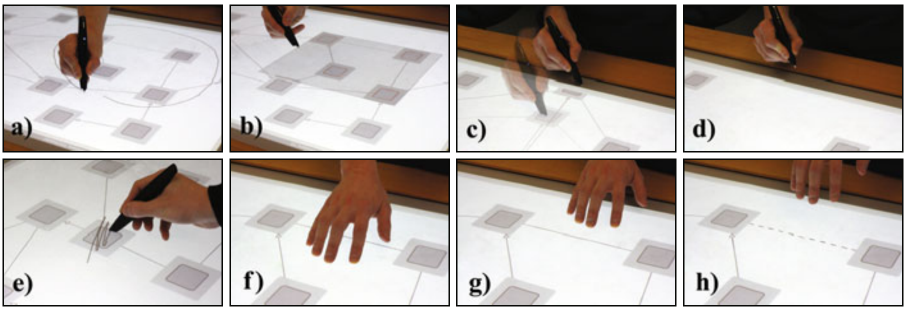



**Fig. 3.** Creating a sub-graph by encircling (a, b), deleting a node by dragging to off screen (c, d) and by wiping (e), changing a solid edge to a dashed one by "rake" gesture (f-h)

### 5.5 Scaling, Zooming and Panning

A pinch gesture with two fingers [25], or finger and pen respectively, is applied for scaling and zooming operations. For zooming the whole diagram, the gesture has to be performed on the background. For scaling it has to be performed on the respective node or sub-graph. Panning a diagram was not part of the tasks of the pilot study. For that, we added an additional gesture. Performing a multi-touch drag gesture with five fingers activates panning, similar to [2]. This gesture does not have to be performed on the background; panning is possible even if some elements are touched. This approach can be beneficial, especially in large diagrams with a huge amount of elements located very close to each other.

## 6 Technical Implementation

For our prototype we are using a multi-touch enabled tabletop system based on Frustrated Total Internal Reflection (FTIR) [18]. The display has a size of 102 x 77 cm and a resolution of 1280 x 800 pixels. For recognizing touch input we are using the Community Core Vision Toolkit [7]. It sends events by means of the TUIO protocol [27] which delivers basic information for each touch, such as position and ID. Beyond that, our hardware is able to distinguish between touch and pen input. To achieve that, the Anoto technology [1] is applied for pen interaction which is similar to [19]. Digital pens read the Anoto pattern that is printed onto the surface of the display and stream the pen coordinates to our software via Bluetooth (see Figure 4 left). The FTIR approach registers touch events within the IR spectrum. As the digital pens emit an IR signal, pens also produce touch events in addition to the Anoto messages. Therefore, a pen is recognized by the system as finger *and* pen. In order to avoid this, a preceding software component is necessary (*InputMerger*, see Figure 4 right) before the events are sent to the editor application. The *InputMerger* ensures that during pen interaction the touch events arising from the pen are omitted and only the respective Anoto events are transmitted to the application. Beyond that, it converts incoming events (TUIO and Anoto) to a uniform data format.

We implemented our own gesture recognizer. Incoming events are clustered by timestamp, distance and graphical context. After recognizing basic gestures, such as dragging, holding or tapping, these gestures are combined, e.g. to *hold and drag* gestures. When a drag gesture is performed with one finger or pen, the respective stroke data is sent to a sketch recognizer. However, this happens just if the gesture is performed on the background or if it is started on the interactive border region of a node as these activities can result in a sketched shape (see Figure 4 right).

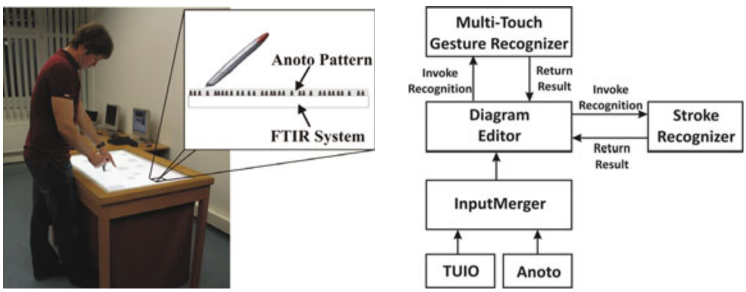



**Fig.4.** Tabletop setup with Anoto pattern on surface (left), Software Architecture (right): TUIO and Anoto events are converted to a consistent data format by *InputMerger*. The application invokes gesture and stroke recognition.

## 7 Discussion

The presented gesture set and prototype covers the most basic functionalities for diagram editing by means of user-defined gestures. It can serve as a basis for further research in this area. In this section, we discuss ways of extending the prototype to a full-value editor for more complex diagram notations.

**Shortcut gestures:** We propose to add additional shortcut gestures. On the one hand, this can be done for general tasks such as undo and redo or cut and paste (e.g. as proposed by [29]). On the other hand, gestures for more domain-specific diagram editing tasks can be added. As a first example we implemented the “rake” gesture to change a dashed edge to a solid one (see 5.4). Further functionalities that can be supported by single pen and hand gestures are, for example the creation of directed edges going back and forth between two nodes or the creation of diagram layouts. For the latter, gestures for generating automatic layouts of whole diagrams or parts of the graph can be applied. We also suggest determining layout constraints manually by means of hand gestures. Early explorations on this topic can be found in [11].

**Contextual assistance:** We propose contextual assistance (e.g. adapted menus) for complex notations consisting of a huge amount of different types of nodes and edges. Of course, contextual assistance can be applied to automatically suggest valid diagram elements. For example, this can be beneficial in situations where elements shall be created, but some of them would break syntactic rules. Beyond that, we propose to apply contextual assistance to gestures. When a gesture is started, the system could give suggestions in which way the gesture can be continued and possibly preview the respective outcomes (similar to [13]).

**Menus:** For editing complex diagram languages with many different types of nodes and edges, menus are certainly necessary. However, as stated above, invoking menus and selecting items usually disrupts the current workflow. Traditional context menus cope with this issue. They are appearing in situ and their items are adapted to the current context. To support a smooth workflow, and with our design goal G3 in mind (see 4.1), we additionally propose that context menus should explicitly support interaction by means of pen & multi-touch input. Therefore, they have to be carefully designed and adapted to these modalities.

## 8 Conclusion and Future Work

We contributed a user-defined and expert-refined gesture set for diagram editing on interactive surfaces which serves as a basis for the implementation of a diagram editor. It is the first one in this domain and consists of uni- and bimanual gestures combining touch and pen input. A collection of user-elicited gestures provided the starting point for the research presented here. Involving users right from the beginning is beneficial, not only to elicit preferred gestures for given tasks, but also to observe prevalent mental models and behaviors. Although users are not designers, these observations can give valuable hints how design challenges can be solved.

In order to realize the set, we identified basic design goals and analyzed existing conflicts within the collection of gestures. We discussed options how these ambiguities can be solved, whereby we seized on users' suggestions. Thereafter, we described the implemented gesture set in detail and briefly presented the architecture of the prototypical editor application.

For future work we will evaluate the currently implemented gesture set and extend it for example by expert gestures. Furthermore, features such as contextual assistance and menu techniques for more complex diagram types shall be added (see Section 7). Beyond that, additional scenarios need to be investigated and carefully studied.

**Acknowledgements.** We thank Sebastian Kleinau for his great support. This work was funded by the "Stifterverband für die Deutsche Wissenschaft" from funds of the Claussen-Simon-Endowment and the German Ministry of Education and Science (BMBF) within the ViERforES project (no. 01 IM 08003).

## References


1. Anoto Group AB, http://www.anoto.com/
2. Brandl, P., Forlines, C., Wigdor, D., Haller, M., Shen, C.: Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. In: Proc. of AVI 2008, pp. 154–161. ACM, New York (2008)
3. Brandl, P., Haller, M., Hurnaus, M., Lugmayr, V., Oberngruber, J., Oster, C., Schafleitner, C., Billinghurst, M.: An Adaptable Rear-Projection Screen Using Digital Pens And Hand Gestures. In: Proc. of ICAT 2007, pp. 49–54. IEEE Computer Society, Los Alamitos (2007)
4. Cao, X., Wilson, A.D., Balakrishnan, R., Hinckley, K., Hudson, S.E.: ShapeTouch: Leveraging Contact Shape on Interactive Surfaces. In: Proc. of TABLETOP 2008, pp. 139–146 (2008)
5. Chen, Q., Grundy, J., Hosking, J.: An e-whiteboard application to support early design-stage sketching of UML diagrams. In: Proc. of HCC 2003, pp. 219–226. IEEE, Los Alamitos (2003)
6. Cherubini, M., Venolia, G., DeLine, R., Ko, A.J.: Let's go to the whiteboard: how and why software developers use drawings. In: Proc. of CHI 2007, pp. 557–566. ACM, New York (2007)
7. Community Core Vision, NUI Group, http://ccv.nuigroup.com/
8. Damm, C.H., Hansen, K.M., Thomsen, M.: Tool support for cooperative object-oriented design: gesture based modeling on an electronic whiteboard. In: Proc. of CHI 2000, pp. 518–525. ACM, New York (2000)
9. Dekel, U.: Supporting distributed software design meetings: what can we learn from co-located meetings? In: Proc. of HSSE 2005, pp. 1–7. ACM, New York (2005)
10. Dietz, P., Leigh, D.: DiamondTouch: a multi-user touch technology. In: Proc. of UIST 2001, pp. 219–226. ACM, New York (2001)
11. Dwyer, T., Lee, B., Fisher, D., Quinn, K.I., Isenberg, P., Robertson, G., North, C.: A Comparison of User-Generated and Automatic Graph Layouts. IEEE Trans. on Visualization and Computer Graphics 15(6), 961–968 (2009)
12. Epps, J., Lichman, S., Wu, M.: A study of hand shape use in tabletop gesture interaction. In: CHI 2006 Ext. Abstracts, pp. 748–753. ACM, New York (2006)
13. Freeman, D., Benko, H., Morris, M.R., Wigdor, D.: ShadowGuides: Visualizations for In-Situ Learning of Multi-Touch and Whole-Hand Gestures. In: Proc. ACM ITS 2009, pp. 183–190. ACM, New York (2009)
14. Frisch, M., Heydekorn, J., Dachselt, R.: Investigating Multi-Touch and Pen Gestures for Diagram Editing on Interactive Surfaces. In: Proc. of ACM IST 2009, pp. 167–174. ACM, New York (2009)
15. Grundy, J., Hosking, J.: Supporting Generic Sketching-Based Input of Diagrams in a Domain-Specific Visual Language Meta-Tool. In: Pro. of ICSE 2007, pp. 282–291 (2007)
16. Guiard, Y.: Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. The Journal of Motor Behavior 19(4), 486–517 (1987)
17. Hammond, T., Davis, R.: Tahuti: a geometrical sketch recognition system for UML class diagrams. In: ACM SIGGRAPH 2006 Courses. ACM, New York (2006)
18. Han, J.Y.: Low-cost multi-touch sensing through frustrated total internal reflection. In: Proc. of UIST 2005, pp. 115–118. ACM, New York (2005)
19. Leitner, J., Powell, J., Brandl, P., Seifried, T., Haller, M., Dorray, B., To, P.: Flux: a tilting multi-touch and pen based surface. In: CHI 2009 Ext. Abstracts, pp. 3211–3216. ACM, New York (2009)

20. Malik, S., Ranjan, A., Balakrishnan, R.: Interacting with large displays from a distance with vision-tracked multi-finger gestural input. In: Proc. UIST 2005, pp. 43–52. ACM, New York (2005)
21. Micire, M., Desai, M., Courtemanche, A., Tsui, K., Yanco, H.: Analysis of Natural Gestures for Controlling Robot Teams on Multi-touch Tabletop Surfaces. In: Proc. ACM IST 2009. ACM, New York (2009)
22. Microsoft Surface, http://www.microsoft.com/surface/
23. Moran, T.P., Chiu, P., van Melle, W.: Pen-based interaction techniques for organizing material on an electronic whiteboard. In: Proc. UIST 1997, pp. 45–54. ACM, New York (1997)
24. Nielsen, M., Störring, M., Moeslund, T.B., Granum, E.: A procedure for developing intuitive and ergonomic gesture interfaces for HCI. In: Camurri, A., Volpe, G. (eds.) GW 2003. LNCS (LNAI), vol. 2915, pp. 409–420. Springer, Heidelberg (2004)
25. Rekimoto, J.: SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. In: Proc. CHI 2002, pp. 113–120. ACM, New York (2002)
26. Ringel, M., Ryall, K., Shen, C., Forlines, C., Vernier, F.: Release, relocate, reorient, resize: fluid techniques for document sharing on multi-user interactive tables. In: Ext. Abstracts CHI 2004, pp. 1441–1444. ACM, New York (2004)
27. TUIO Protocol, http://www.tuio.org/
28. Wobbrock, J.O., Morris, M.R., Wilson, A.D.: User-defined gestures for surface computing. In: Proc. of CHI 2009, pp. 1083–1092. ACM, New York (2009)
29. Wu, M., Balakrishnan, R.: Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In: Proc. of UIST 2003, pp. 193–202. ACM, New York (2003)
30. Wu, M., Shen, C., Ryall, K., Forlines, C., Balakrishnan, R.: Gesture Registration, Relaxation, and Reuse for Multi-Point Direct-Touch Surfaces. In: Proc. TABLETOP 2006, pp. 185–192. IEEE, Los Alamitos (2006)
31. Yee, K.: Two-handed interaction on a tablet display. In: Ext. Abs. CHI 2004, pp. 1493–1496. ACM, New York (2004)

# The Effects of Perception of Efficacy and Diagram Construction Skills on Students' Spontaneous Use of Diagrams When Solving Math Word Problems

Yuri Uesaka[1], Emmanuel Manalo[2], and Shin'ichi Ichikawa[3]

[1] Japan Society for the Promotion of Science, Tokyo Institute of Technology
[2] Waseda University
[3] The University of Tokyo
y.uesaka@nm.hum.titech.ac.jp, emmanuel.manalo@gmail.com, ichikawa@p.u-tokyo.ac.jp

**Abstract.** Although diagram use is considered to be one of the most effective strategies for solving problems, reports from applied educational research have noted that students lack spontaneity in using diagrams even when teachers extensively employ diagrams in instructions. To address this problem, the present study investigated the effectiveness of teacher-provided verbal encouragement (VE) and practice in drawing diagrams (PD), as additions to typical math classes, for promoting students' spontaneous use of diagrams when attempting to solve problems. The participants were 86 8th graders who were assigned to one of four instruction conditions: VE+PD, VE only, PD only, and with no addition to typical instruction (Control). The highest improvement in spontaneous diagram use was observed in the VE+PD condition. This finding suggests that, to promote spontaneity in students' diagram use, helping students appreciate the value of diagram use is important, as well as developing procedural knowledge in using diagrams.

**Keywords:** spontaneous diagram use, math word problem solving, perception of efficacy of diagrams use, construction skills in drawing diagrams.

## 1 Introduction

The use of diagrams is considered by many researchers and teachers to be a strategy that promotes efficacy when solving problems. The usefulness of diagrams in problem solving situations can probably be best understood in terms of Larkin and Simon's [1] explanation that diagrammatic representations are more computationally effective than sentential representations because they reduce mental loads associated with memory and searching. The benefits of employing diagrams and other forms of visual representation when problem solving have been empirically demonstrated in numerous studies [e.g., 2, 3, 4, 5; see also a review by Cox, 6].

However, novices and many school students appear not to be able to use diagrams as effectively as teachers and researchers. Numerous problems relating to students' use of diagrams have been pointed out in reports from applied educational research

⋆ An erratum for this chapter can be found at: http://dx.doi.org/10.1007/978-3-642-14600-8_51

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 197–211, 2010.

and in observations of students' actual problem solving: for example, students demonstrate poor choice of diagrams to use [7, 8], they fail to draw appropriate inferences when using diagrams [9, 10], and they lack spontaneity in using diagrams [11, 12, 13]. All of these problems need to be resolved if the development of skills in effective problem solving among school students is to be facilitated via the use of diagrams.

### 1.1 Students' Lack of Spontaneity in Diagram Use

Students' lack of spontaneity in using diagrams is one of the most critical of the problems, and has been noted in reports concerning applied educational activities. Ichikawa [11], for example, described the case of an 8th grade student who tried in vain during a test situation to solve a problem without the use of any diagram and gave up – despite the fact that, just prior to the test, she was provided instruction and encouragement by the researcher to use diagrams. Other reports have also been made about this lack of spontaneity in students' attempts to solve problems for which the use of diagrams would have been considered effective [e.g., 12, 13].

There is also a contrasting difference between the amount of diagrams that teachers and students use in the process of problem solving. For example, Dufour-Janvier et al. [12] pointed out that students did not use diagrammatic representations in mathematics even though they had plenty of opportunities to observe their teachers using diagrams in class. In addition, although a video-based study conducted by TIMSS (Trends in International Mathematics and Science Study) [14] revealed that Japanese teachers use a lot of diagrams in class for instruction, a number of research studies [e.g., 13, 15] have found that Japanese students are weak in the spontaneous use of diagrams when solving math word problems. These findings suggest the necessity of developing teaching methods that would promote the spontaneous use of diagrams, and which could be used to supplement the teaching methods that are normally used in class.

To date, however, the main focus of psychological studies that have been published about diagrams have been to demonstrate their effects [e.g., 3, 4] and the mechanisms by which they promote problem solving [e.g., 1, 2]; only a limited number of studies have examined problems concerning diagrams use. Additionally, the studies that have looked into problems of diagrams use have been concerned with people's inappropriate choice [e.g., 7, 8] and failure to make correct inferences when using diagrams [e.g., 9, 10]. The problem of lack of spontaneity with which people use diagrams has not been sufficiently investigated in prior research in this area.

The importance of addressing the issue of students' lack of spontaneity in diagrams use becomes even more apparent when one considers findings from previous studies that self-constructed diagrams are particularly powerful heuristics in problem solving [e.g., 4, 5, 6]. The lack of spontaneity means that many students fail to gain the potential benefits of diagrams use in learning – especially about problem solving. Previous studies that the present authors have conducted and published on this topic have revealed some of the factors that may contribute to the lack of spontaneity [e.g., 16, 17]; however, the possible methods that could address this problem have not been adequately examined. It was therefore the purpose of the present study to develop and evaluate a teaching method that would promote students' spontaneous use of diagrams when attempting to solve math word problems.

### 1.2 Factors Influencing the Spontaneous Use of Diagrams

Although previous research has not put forward teaching methods to promote students' spontaneous use of diagrams, findings from the learning strategies research area provide some helpful suggestions about factors that need to be considered when developing such methods. One such factor is the user's perception of the strategy under consideration: it is important that the strategy is perceived as adding efficiency to the task at hand. For example, Garner [18] emphasized the importance of people understanding the value of a particular strategy if it is to be spontaneously used. Other empirical research [e.g., 19, 20], including where diagrams use is concerned [16], has also demonstrated that students' perception of the efficacy of a strategy influences the use of that strategy. Thus, in the present study, encouragement aimed at improving students' perception of the efficacy of diagrams use was included as part of the intervention.

However, on its own, perception of the efficacy of a strategy is not enough to promote spontaneous use. In the case of diagrams use, students who have not acquired sufficient skills in using diagrams are unlikely to use them spontaneously even if they perceive likely benefits in their use. As Ames and Archer [21] pointed out, research focusing on user perceptions about a strategy actually stemmed from, and was a reaction to, an initial focus by researchers only on the crucial role of skills training. Murayama [22] also pointed out the importance of acquiring the necessary skills when using external resources like diagrams. Thus the intervention used in the present study included not only encouragement to enhance the perception of efficacy of diagrams use, but also steps to promote the development of construction skills for drawing diagrams.

The framework used in this study of combining promotion of the relevant skills development and instruction to improve perception of the value of a strategy is supported by some of the research literature in this area. For example, the "informed training" approach proposed by Brown, Bransford, Ferrara, and Campione [23] emphasized the importance of adding into skills training explicit instruction about the values of the strategy concerned and the situations in which its use is beneficial.

Research in the area of learning strategies use also suggests that the effect of an intervention would be influenced by the students' beliefs about learning. Shinogaya [24] found that an intervention to encourage students to read their study text before a history class benefited only those students who had a "meaning orientation" (i.e., those who valued memorizing with understanding rather than just rote memorizing). This suggests that the efficacy of the interventions used in the present study is also likely to be affected by students' beliefs about learning. In other words, if students believe that getting the correct answer is all that matters and neglect the process of how to get that answer, the effect of the intervention for promoting diagrams use would likely be minimal, if any. Thus, in the present study, students' process/outcome orientation (i.e., whether they valued the process of problem solving or simply getting the correct answer [25]) was also taken into consideration and included as a covariant in the analysis of the results.

### 1.3 Means for Improving the Perception of Efficacy and Construction Skills

Although the approach of enhancing both perception of efficacy and skills in use of a strategy could be considered potentially effective in promoting students' spontaneity in diagrams use, the exact means for implementing such an approach in a classroom teaching situation still needed to be developed. For enhancing the perception of efficacy, as suggested in Brown et al.'s [23] idea of "informed training", the teacher may well need to explicitly explain the beneficial effects of diagrams use in problem solving. The findings from a survey conducted by Uesaka et al. [13] suggested that students tend to think of diagrams as "tools for teachers' instruction" but not as "tools for their own problem solving", and this tendency linked to lower levels of spontaneity in diagrams use. Therefore, there are clear indications here that the teacher needs to explicitly tell students about the efficacy of diagrams use in their – the students' – own problem solving.

Moreover, the teachers' explicit encouragement might be more effective if effort or lack of effort in using the strategy is appropriately attributed to success or failure in problem solving. Chan [26], for example, attempted to improve 7th graders' strategy use during reading and found that combining skills training and attributional training, during which teachers explicitly urged student to attribute successes and failures in reading to use of reading strategies, improved reading strategy use and overall performance. Other studies have also suggested the importance of attribution in promoting strategy use [e.g., 27]. Thus it would appear beneficial to include attribution training in an intervention for enhancing the perception of efficacy of diagrams use.

To develop students' skills in constructing diagrams, the provision of opportunities for students to actually construct and use diagrams as tools for solving problems would appear crucial. Previous educational programs developed to promote students' strategies use have also included many opportunities for participants to practice how to use the target strategies in realistic tasks [e.g., 26, 28]. In addition, Uesaka and Manalo [17] observed that students encountered greater difficulties in using diagrams when the problems they were attempting to solve required the transformation of the concrete situation depicted in the problem to a more abstract diagrammatic representation. This would therefore suggest that in developing students' skills in diagrams construction, it would be important for the teacher to provide opportunities for practice in producing appropriate diagrams for a wide range of problems – including those that are more demanding in terms of the abstractness of the diagrammatic representations they require.

In addition to that, including error-related feedback about students' use of diagrams would appear beneficial to include in promoting the development of students' diagram construction skills. Research about metacognitive learning strategies suggests that, in order to learn, students need to know the reason behind experiences of failure [e.g., 25, 29]. Where practice in drawing to promote diagram construction skills is concerned, pointing out to students where errors occur in their diagram use would contribute to the development of procedural knowledge in constructing appropriate diagrams for problem solving. Thus, in the intervention used in the present study, error-related feedback, as well as feedback about the formula and answer to the problem given, was included.

### 1.4 Hypothesis and Overview of the Present Study

The main hypothesis investigated in the present study was that an instructional intervention that sought to improve both students' perceptions about the efficacy resulting from diagram use and their procedural knowledge in drawing diagrams would produce the greatest improvements in the students' spontaneous use of diagrams when attempting to solve math word problems. It was also hypothesized that students' process/outcome orientation would influence any changes in their spontaneous use of diagrams that might result from the interventions provided.

To enhance participants' perceptions about the efficacy of diagrams use, teacher verbal encouragement to use diagrams was added to math class instruction based on what could be considered 'traditional' (i.e., in which teachers use diagrams to demonstrate how to solve problems but do not explicitly encourage their use [16]). And to improve participants' diagram construction skills, practice in drawing diagrams was provided – again in addition to traditional math class instruction. This study also examined whether providing teacher encouragement in diagrams use would improve students' perceptions about the efficacy of diagrams use. Likewise, it examined whether providing practice in drawing diagrams would produce evidence of improvements in procedural knowledge in constructing diagrams.

## 2 Method

### 2.1 Participants and Experimental Design

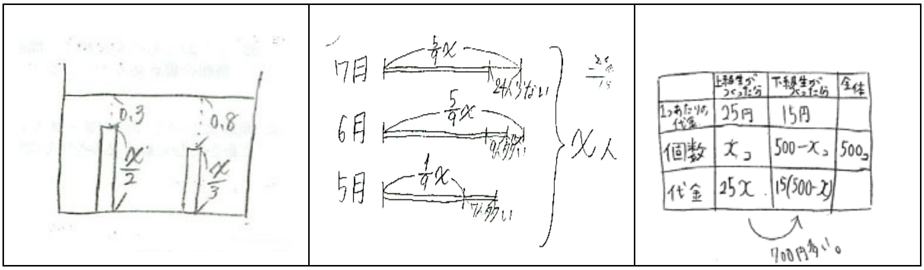
The participants were assigned to one of four conditions: a class in which both verbal encouragement and skills training for drawing diagrams were provided in addition to instruction in math word problem solving (VE+PD); a class in which only verbal encouragement was additionally provided (VE); a class in which only skills training in drawing diagrams was additionally provided (PD); and a class in which no additional manipulations were added to the regular instruction in math word problem solving provided (Control).

A randomized block design was used to reduce potential confounding effects relating to the students' school achievement. Information about the participants' school achievement was gathered from their parents via a survey administered before the experiment, and responses to this survey were used to achieve approximate equivalence in student assignment across the four groups. The parents' responses to the survey were not included in subsequent analyses.

In total there were 86 participants: 24 in the condition with VE+PD, 19 in the condition with VE only, 21 in the condition with PD only, and 22 in the Control condition.

### 2.2 Materials

Three categories of math word problems were used in the experiment (see Appendix 1 for examples of the actual problems, and Fig. 1 for examples of diagrams produced by students during the instruction sessions). The first category comprised of "picture problems", for which constructing a mechanical or geometric illustration is effective for solving. The second category comprised of "bar chart problems", for which solutions are facilitated by constructing a diagram in the form of a bar chart that represents the important quantities (bar charts shown in Fig. 1 are popular visual representations in Japanese educational settings). The third category comprised of "table problems", for which drawing a table depicting relationships between quantities described is helpful for solving. These three categories of math word problem were used during the first, second, and third days of instruction respectively, and on each day three problems from the assigned category were given to the participants to solve. (The effectiveness of the kinds of diagrams corresponding to each category noted above was confirmed with three certified math teachers in Japan.)




Picture Problem Bar Chart Problem Table Problem

**Fig. 1.** Typical Diagrams Produced During the Instruction Sessions

**Math Word Problem Solving Assessment.** This was administrated at pre-instruction and post-instruction to find out how often students spontaneously used diagrams when solving math word problems. The assessment at pre-instruction comprised of three math word problems, one from each of the three categories (i.e., picture, bar chart, table), with their level of difficulty being the same as in the instruction sessions. The assessment conducted at post-instruction also consisted of three math word problems, one from each category. However, these problems were different from, and more difficult than, the problems used at the pre-instruction assessment and instruction sessions (see Appendix 2). The increase in difficulty was decided as general improvements in the participants' problem solving competence was expected following the instructions provided. The participants were asked to show their working throughout. Four minutes were allowed for each problem.

Immediately after the post-instruction assessment, the participants were given a new booklet containing the same three problems, but this time participants were explicitly asked to use diagrams in their attempts to solve them. This additional procedure was undertaken to assess participants' skills in diagram construction via the quality of diagrams they produced (when required to construct them). This time, considering that they were the exact same problems that the participants had just attempted solving (and thus they did not need to read the problems again in detail), they were allowed only 3 minutes for each problem.

**Survey Conducted Before and After the Experimental Class.** A survey was sent to participants (using conventional mail) about one month prior to the start of the experimental classes. Two items for assessing participant's process/outcome orientation, drawn from a questionnaire developed by Ichikawa and his colleagues to assess students' motivations and beliefs about learning [see, 25], were included: "The process of problem solving is as important to me as obtaining the correct answer", and "The process of problem solving does not really matter as long as I obtain the correct answer" – the second being a reverse item. The participants had to respond on a 5-point Likert-type scale (with the end points being anchored as: 5 = "completely think so" and 1 = "do not think so at all").

The post-instruction survey was distributed on the final day of the class, together with a return mail envelope. To determine whether participants who were in the conditions with VE would show greater evidence of appreciating the efficacy that diagram use brings to problem solving, the participants were given the opportunity to write freely to the teacher of the class through a "free comment" space in the survey. The comments provided were used to assess the participants' levels of perception about the efficacy of diagram use.

### 2.3 Procedure

The experimental classes were organized and provided over the course of five days at the University of Tokyo. The pre- and post-instruction assessments were administered on the first and fifth days. In the three days in between, the instruction classes were provided. The daily sessions each lasted about 50 minutes.

In each group, the instruction sessions followed the same basic procedure. First, the teacher introduced the first of the three problems for the day and asked the participants to attempt to solve it. Then, the teacher explained on the board how to solve the problem correctly (with the use of a diagram). Following this, the teacher introduced the second problem and asked the participants to try to solve it. After waiting for a while, the teacher again explained on the board (with the use of a diagram) how to correctly solve it. Finally, the teacher introduced the final problem for the day and asked the participants to try to solve it. The teacher then collected the participants' efforts at solving this third problem, and returned these together with feedback at the start of the next class.

In addition to the basic procedure described above, in the conditions with VE, the following manipulations were included. After the students' attempt at solving the first problem, the teacher asked the participants if they were successful in solving it. The teacher then told the participants who indicated they were not successful that they would have fared better had they used an appropriate diagram. At this time, the teacher also stressed the effectiveness of using diagrams when problem solving, and encouraged the participants to construct diagrams particularly when attempting difficult problems. To participants who were able to solve the first problem without the use of diagrams, the teacher still pointed out the efficacy that diagram use brings, and encouraged them to use diagrams when they encountered more difficult problems. Prior to the participants' attempts at solving the second and third problems, the teacher also provided encouragement for them to use diagrams.

In the conditions with PD, the following three manipulations were included additional to the basic procedure described above. Firstly, in this condition, the space in the students' worksheet for solving the given problem was segmented into two parts: space for writing the appropriate numerical formula, and space for constructing diagrams. Thus, they were more or less compelled to construct diagrams when considering the problem. Secondly, these participants were also asked to reconstruct diagrams that the teacher had used on the board in demonstrating how to solve each of the problems. In each instance, only 2 minutes were allowed for this, and students were encouraged not to look at the diagram on the board or their notes. Thirdly, the way that feedback was provided differed depending on the condition. The teacher collected the last problem at the end of the session for return the following day. Although participants in all conditions received feedback on the answer and numerical formula they produced, participants in the conditions with PD were provided additional feedback when they made errors in the diagrams they used.

# 3 Results

## 3.1 Spontaneous Use of Diagrams in Math Word Problem Assessment

To evaluate whether participants tried to use diagrams in attempting to solve the problems administered at pre- and post instruction assessment, participants' responses were independently scored by the first author and another scorer. For the purposes of the present study, a diagram was defined as any representation of the problem other than words, sentences, or numerical formulas. Tables were counted as diagrams and, a table was defined as a depiction of at least a pair of values arrayed to represent two related variables. The inter-rater agreements (measured by calculating Cohen's kappa coefficients) were found to be .71 for the pre-instruction assessment data and .84 for the post-instruction assessment data, which can both be considered as substantially concordant. The results are shown in Table 1 and Fig. 2.

**Table 1.** Average Number of Problems Where Participants Spontaneously Used Diagrams in Each Assessment (Out of 3 Problems) and the Process/Outcome Orientation in Each Condition

| Condition | Pre-instruction (SD) | Post-instruction (SD) | Process Orientation (SD) |
|---|---|---|---|
| VE+PD | 0.17 (0.38) | 1.58 (1.28) | 3.70 (0.74) |
| VE | 0.26 (0.65) | 0.58 (0.90) | 3.66 (1.13) |
| PD | 0.52 (0.75) | 1.09 (1.30) | 3.82 (0. 94) |
| Control | 0.32 (0.65) | 0.82 (1.14) | 3.86 (0.86) |

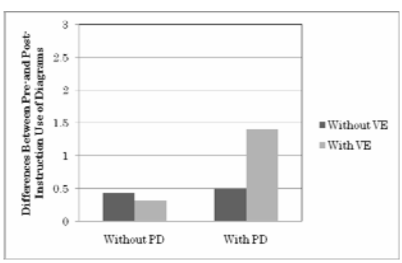



**Fig. 2.** Calculated Differences Between Pre- and Post-Instruction Use of Diagrams

A 2 x 2 analysis of covariance (ANCOVA) ([with vs without VE] x [with vs without PD], with the participants' process orientation scores as a covariant) was carried out on the calculated differences between the participants' spontaneous diagram use at pre-instruction and at post-instruction. VE and PD were between-subject factors, and process orientation was a within-subject variable.

A significant effect stemming from PD ($F_{(1, 79)} = 5.11$, $p < .05$) and a significant interaction effect between PD and VE ($F_{(1, 79)} = 4.07$, $p < .05$) were found. The effect of VE was not statistically significant ($F_{(1, 79)} = 1.91$, n.s.). The process orientation was marginally significant ($F_{(1, 79)} = 3.18$, $p < .10$). The analysis revealed no interaction effects between process/outcome orientation and the other factors examined.

The significant interaction effect between PD and VE was analyzed in more detail. An analysis of simple effects revealed the effects of PD was significant when it was provided with VE ($F_{(1, 79)} = 9.31$, $p < .01$). In contrast, the effects of PD was not significant when it was provided without VE ($F(1, 79) = .03$, n.s.). This interaction effect suggests that diagram use was most effectively promoted in the condition where *both* VE and PD were provided as shown clearly in Fig. 2. This result supports the main hypothesis.

## 3.2 Analysis of the Perception of Efficacy of Diagrams

To examine whether participants provided with VE would manifest improvements in their perceptions of the efficacy of diagrams use, participant's responses in the "free comments" section of the post-instruction survey were analyzed. Two scorers independently examined each participant's response and determined in each case whether reference was made to the importance of, or efficacy relating to, diagram use. The inter-rater agreement (Cohen's kappa coefficient) was found to be .91, which indicates almost perfect concordance. The following statements are examples of responses that participants made (translated from Japanese) that were deemed as referring to the value of diagram use: "I was glad to learn how to make tables and diagrams; I am not good at making equations, so I want to use these other ways", "I now understand the importance of diagrams", and "I used to have a problem of getting confused when reading word problems, but this hasn't happened as much recently when I have been using diagrams".

**Table 2.** Ratios of Participants Referring to the Efficacy of Diagram Use in the Free Comments Section of the Post-instruction Survey

| | With PD | Without PD | Totals |
|---|---|---|---|
| With PD | 0.46 (11/24) | 0.42 (8/19) | 0.44 (19/43) |
| Without PD | 0.10 (2/21) | 0.18 (4/22) | 0.14 (6/43) |
| Totals | 0.29 (13/45) | 0.29 (12/41) | 0.29 (25/86) |

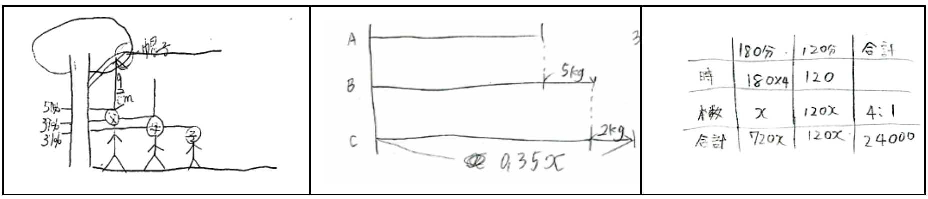
*Note.* In parentheses are the number of participants who referred to the efficacy of diagram use, and the total number of participants, in each condition.

By using arcsine transformations, the proportions of participants referring to diagram use efficacy in each condition were compared. These proportions are shown in Table 2. The main effect of VE was significant ($\chi^2_{(1)} = 8.54$, $p < .01$). However, neither the main effect of PD ($\chi^2_{(1)} = .05$, n.s.), nor the interaction between the two factors ( $\chi^2_{(1)}$ = .35, n.s.) was significant. These results suggest that participants in the conditions with VE evidenced better appreciation of the efficiencies that diagram use brings to problem solving.

### 3.3 Analysis of the Diagram Construction Skills Participants Acquired

To examine whether participants with PD would improve their diagram construction skills, the quality of diagrams drawn in the latter half of the post-instruction assessment was analyzed. During this session, all participants were required to draw diagrams in their attempts to solve the problems given. The quality of the diagrams that the participants produced was used as an indication of the diagram construction skills that they had acquired through the experimental class.

The quality of the diagrams was rated using specified criteria. In essence, the diagrams were classified as being "high quality" when they saliently represented the relationships between the most important quantities from the math word problems given: this function of diagrams has been identified as being important in earlier research by Uesaka and Manalo [17]. Two scorers independently carried out the diagram ratings; inter-rater agreement (Cohen's kappa coefficient) was found to be .82, which was considered as substantially concordant. Examples of diagrams produced by participants are shown in Fig. 3.



Length Problem (One of the Picture Problems) | Weight Problem (One of the Bar Chart Problems) | Video Problem (One of the Table Problems)

**Fig. 3.** Examples of Diagrams Constructed in Problems Given in Post-Instruction Assessment

A 2 [with vs without VE] x 2 [with vs without PD] analysis of variance (ANOVA) was carried out on the total numbers of problems where diagrams deemed as being of "high quality" were constructed by participants. The main effect of PD was significant ($F_{(1, 82)}$ = 6.51, $p$ < .05). However, both the effect of VE ($F_{(1, 82)}$ = 1.48, n.s.), and the interaction between the two variables ($F_{(1, 82)}$ = .07, n.s.), were not significant. This finding suggests that providing PD improved participants' diagram construction skills. Results are shown in Table 3.

To examine the validity of the criteria used for classifying the quality of the participants' diagrams, the relationship between diagrams classification and the correctness of the answers produced was analyzed. In every math word problem given in the post-instruction assessment, greater proportions of correct solutions were found in cases where the diagrams used had been classified as "high quality" compared to

**Table 3.** The Total Number of Problems for which Participants Produced High Quality Diagrams When Asked to Use Diagrams in the Post-instruction Assessment

| | With PD | Without PD | Totals |
|---|---|---|---|
| With PD | 2.46 (0.59) | 1.95 (0.91) | 2.23 (0.78) |
| Without PD | 2.19 (0.75) | 1.77 (1.06) | 1.98 (0.94) |
| Totals | 2.46 (0.59) | 1.95 (0.91) | 2.23 (0.78) |

cases where the diagrams had been classified as “low quality”. The differences were statistically significant or marginally significant in all three problem categories ($\chi^2_{(1)} = 14.90$, $p < .01$; $\chi^2_{(1)} = 7.40$, $p < .01$; $\chi^2_{(1)} = 3.41$, $p < .10$ respectively).

## 4 Discussion

The results of the present study revealed that an intervention incorporating teacher-provided verbal encouragement and skills training in drawing diagrams effectively promoted students’ spontaneous use of diagrams when attempting to solve math word problems. The analysis of the free comment in the post-instruction survey showed that teacher-provided verbal encouragement enhanced the perception of efficacy of diagrams, and the analysis of the quality of diagrams produced in the post-instruction session demonstrated that practice in drawing diagrams improved diagram construction skills.

This finding suggests that for students to spontaneously use diagrams when given math word problems to solve, they need to appreciate the efficacy of diagram use, as well as know how to use diagrams correctly for such purposes. In other words, students need to have a clear sense that using diagrams in problem solving is “worth the effort” *and* that they “can do this” (without too much difficulty). The provision of verbal encouragement to use diagrams appeared to effectively facilitate the former, and the skills training for drawing diagrams appeared to do the same for the latter.

### 4.1 Contributions to Diagrams Research

The useful contribution of the present study to diagrams research is that it identified a possible teaching method for improving students’ spontaneity in using diagrams when solving math word problems. Although reports and observations from educational practice have previously highlighted the problem of students’ lack of spontaneity in diagrams use, and some of the contributing factors to this problem have been identified, no concrete instructional methods for addressing the problem had been put forward prior to the present study. The findings of the present study provide some indication of how an autonomous learner with a sufficiently positive attitude to learning (i.e., process oriented) can be developed through instruction to more spontaneously utilize external resources (i.e., diagrams) in appropriate situations (i.e., when problem solving).

The problem – about students not using diagrams spontaneously even when they have lots of opportunities to observe their teachers using diagrams during classroom instructions – is one that had not been dealt with in previous research. It is important, however, to seek viable solutions to this problem as spontaneity is an indispensable first step in the autonomous use of self-constructed diagrams – and, as noted earlier, prior research has shown self-constructed diagrams as being powerful heuristics in problem solving [e.g., 4, 5, 6]. As positive outcomes were obtained from the strategies employed in the present study, it provides an initial step towards the development of solutions to the identified problem.

The finding of this study about students' process/outcome orientation working as a covariant suggests that the effects of the intervention used for enhancing students' spontaneous use of diagrams were affected by the students' beliefs about learning. The effects of this kind of general belief about learning had not previously been examined in research involving diagrams. It suggests that students who regard the production of the correct answer as being the sole objective of learning would not benefit as much from the provision of the intervention used. Future investigations will need to examine in more detail the mechanisms involved in process/outcome orientation affecting the use of diagrams.

### 4.2 Contributions to Educational Practices

The findings of this study also draw attention to the incorrectness of the assumption that, in teaching, demonstration alone is adequate. In the present study, even though the teacher in the control group demonstrated with the use of diagrams in providing instructions, the resulting spontaneity of diagrams use for this group was lower than that for students in the conditions with teacher-provided verbal encouragement and skills training. However, the assumption that demonstration is enough tends to permeate most areas of educational practice, from teaching the very basics of reading and arithmetic to young children, to the delivery of lectures to college students. Talking and demonstrating to students are assumed to be the basic – and sometimes the *only* – mechanisms for instigating developments in the knowledge and skills students possess. This is despite previous research findings about the limited efficacy of the demonstration method on its own. For example, in arithmetic instruction for students with learning disabilities, the addition of imitation as a step to promote the actual acquisition of the procedures in question [see, e.g., 30] and process mnemonics as a method for enhancing retention of those procedures [31] have been reported as being effective. As the findings of this study suggest, students benefit from additionally knowing "why" (they should use a particular strategy) and knowing exactly "how" (to use that strategy), and simply demonstrating to them is not enough to facilitate these.

It would be useful in future investigations to examine more closely the effects resulting from the additional manipulations provided in this study to enhance spontaneity in diagrams use. For example, participants in the condition with skills training were 1) compelled to construct diagrams when attempting to solve the problems given in the final worksheet, 2) asked to reconstruct diagrams that the teacher had used on the board in demonstrating how to solve each of the problems, and 3) provided additional feedback when they made errors in the diagrams they used. Which of these additional manipulations contributed most to the spontaneous use of diagrams was not examined in this study. In addition, although this study investigated the factors promoting spontaneous use of diagrams and proposed a concrete method for teaching, there is no guarantee that the method would work in natural settings like the social context in schools. Uesaka et al. [16], for example, suggested that providing opportunities for students to communicate with the use of diagrams would promote subsequent spontaneity in the students' use of diagrams. More ecologically valid teaching method including considerations of some of the factors noted is another possible direction for future research.

## References

1. Larkin, J.H., Simon, H.A.: Why a diagram is (sometimes) worth ten thousand words. Cognitive Science 11, 65–99 (1987)
2. Ainsworth, S., Th Loizou, A.: The effects of self-explaining when learning with text or diagrams. Cognitive Science 27, 669–681 (2003)
3. Pedone, R., Hummel, J.E., Holyoak, K.J.: The use of diagrams in analogical problem solving. Memory & Cognition 29, 214–221 (2001)
4. Cheng, P.C.H.: Electrifying diagrams for learning: Principles for complex representational systems. Cognitive Science 26, 685–736 (2002)
5. Stern, E., Aprea, C., Ebner, H.G.: Improving cross-content transfer in text processing by means of active graphical representation. Leaning and Instruction 13, 191–203 (2003)
6. Cox, R.: Representation construction, externalised cognition and individual differences. Learning and Instruction 9, 343–363 (1999)
7. Grawemeyer, B., Cox, R.: The effect of knowledge-of-external-representations upon performance and representational choice in a database query task. In: Blackwell, A.F., Marriott, K., Shimojima, A. (eds.) Diagrams 2004. LNCS (LNAI), vol. 2980, pp. 351–354. Springer, Heidelberg (2004)
8. Uesaka, Y., Manalo, E.: Active comparison as a means of promoting the development of abstract conditional knowledge and appropriate choice of diagrams in math word problem solving. In: Barker-Plummer, D., Cox, R., Swoboda, N. (eds.) Diagrams 2006. LNCS (LNAI), vol. 4045, pp. 181–195. Springer, Heidelberg (2006)
9. Mokros, J.R., Tinker, R.F.: The impact of microcomputer-based labs on children's ability to interpret graphs. Journal of Research in Science Teaching 24, 369–383 (1987)
10. Cox, R.: Analytical reasoning with multiple external representations. Unpublished Ph.D Thesis: The University of Edinburgh, United Kingdom (1996)
11. Ichikawa, S. (ed.): Gakusyuu wo sasaeru ninnchikaunsering: Shinrigaku to kyouiku no aratana setten (Cognitive counseling that supports learning: A new approach bridging psychology and education). Brain Press, Tokyo (1993)
12. Dufour-Janvier, B., Bednarz, N., Belanger, M.: Pedagogical considerations concerning the problem of representation. In: Janvier, C. (ed.) Problems of representation in the teaching and learning of mathematics, pp. 110–120. Erlbaum, Hillsdale (1987)
13. Uesaka, Y.: How learning skills support through cognitive counseling can provide new perspective in both cognitive research and school curriculum development - Focusing on the strategy of diagram use in problem solving. Cognitive Studies 16, 313–332 (2009)

14. Mullis, I.V.S., Martin, M.O., Gonzalez, E.J., Chrostowski, S.J.: TIMSS 2003 international mathematics report- Findings from IEA's trends in international mathematics and science study at the fourth and eighth grades. In: TIMSS & PIRLS International Study Center, Lynch School of Education, Boston College, Chestnut Hill, MA (2004)
15. Ichikawa, E., Seo, M., Kiyokawa, S., Uesaka, Y.: Development and application of COMPASS - Componential assessment for basic competence and learning in mathematics. In: Proceedings of 2007 international conference of competences based education reform, pp. 379–430. Center for Research of Core Academic Competences, Tokyo (2007)
16. Uesaka, Y., Manalo, E., Ichikawa, S.: What kinds of perceptions and daily learning behaviors promote students' use of diagrams in mathematics problem solving? Learning and Instruction 17, 322–335 (2007)
17. Uesaka, Y., Manalo, E.: Task-related factors that influence the spontaneous use of diagrams in math word problem solving. Paper Presented at the 7th Biennial Meeting of the Society for Applied Research in Memory and Cognition, Lewiston, Main (2007)
18. Garner, R.: When children and adults do not use learning strategies: Toward a theory of settings. Review of Educational Research 60, 517–529 (1990)
19. Murayama, K.: Learning strategy use and short- and long-term perceived utility. Japanese Journal of Educational Psychology 51, 130–140 (2003)
20. Sato, J.: Effects of learners' perceptions of utility and costs, and learning strategy preferences. Japanese Journal of Educational Psychology 46, 367–376 (1998)
21. Ames, C., Archer, J.: Achievement goals in the classroom - students' learning strategies and motivation processes. Journal of Educational Psychology 80, 260–267 (1988)
22. Murayama, I.: Transformation of tasks and cognitive agents by external resources. Cognitive Studies 2(4), 28–38 (1995)
23. Brown, A.L., Bransford, J.D., Ferrara, R., Campione, J.: A tetrahedral framework for exploring problems of learning. In: Flavell, J.H., Markman, E.M. (eds.) Handbook of child psychology, Cognitive development, 4th edn., vol. 3, pp. 85–106. Wiley, New York (1983)
24. Shinogaya, K.: Effects of preparation on learning: Interaction with beliefs about learning. Japanese Journal of Educational Psychology 56, 256–267 (2008)
25. Ichikawa, S.: Cognitive counseling to improve students' metacognition and cognitive skills. In: Shwalb, D.W., Nakazawa, J., Shwalb, B.J. (eds.) Applied developmental psychology - Theory, practice, and research from Japan, pp. 57–87. Information Age, Greenwich (2005)
26. Chan, L.K.S.: Combined strategy and attributional training for seventh grade average and poor readers. Journal of Research in Reading 19, 111–127 (1996)
27. Borkowski, J.G., Carr, M., Rellinger, L., Pressley, M.: Self-regulated cognition-Interdependence of metacognitions attributions and self-esteem. In: Jones, B.J., Idol, L. (eds.) Dimensions of thinking and cognitive instruction, pp. 53–92. Erlbaum, Hillsdale (1990)
28. Guthrie, J.T., Wigfield, A., VonSecker, C.: Effects of integrated instruction on motivation and strategy use in reading. Journal of Educational Psychology 92, 331–341 (2000)
29. Ellis, S., Davidi, I.: After-event reviews - Drawing lessons from successful and failed experience. Journal of Applied Psychology 90, 857–871 (2005)
30. Rivera, D.M., Smith, D.D.: Using a demonstration strategy to teach midschool students with learning disabilities how to compute long division. Journal of Learning Disabilities 21, 77–81 (1988)
31. Manalo, E., Bunnell, J.K., Stillman, J.A.: The use of process mnemonics in teaching students with mathematics learning disabilities. Learning Disability Quarterly 23, 137–156 (2000)

# Appendix 1. Examples of Problems Used in Instruction Session

**Example of Picture Problem Used**

Masashi decided to measure the height of his bed. First he used one-fifth of a string, and found it was 22 cm shorter than the height of the bed. So he measured again by using one-third of the same length of string, and found this time that it was 2 cm longer than the height of the bed. How high was his bed?

**Example of Bar Chart Problem Used**

Hiroko examined the numbers of the students in her high school. The number in the $1^{st}$ grade was 15 students fewer than 36% of all in the school, and the number of students in the $2^{nd}$ grade was 7 students more than in the $1^{st}$ grade. Finally, the total in the $3^{rd}$ grade was 4 students fewer than 34% of all students. At this time, how many students were there in the $2^{nd}$ grade?

**Example of Table Problem Used**

200 handkerchiefs were put out for sale at 300 yen. But not many handkerchiefs were sold at this price. The price was therefore changed to 220 yen. Then the rest of the handkerchiefs were sold. The total money from selling all of the handkerchiefs was 46,400 yen. How many handkerchiefs were sold at 300 yen?

# Appendix 2. Problems Provided at Post-Instruction Assessment

**Tree Problem (One of the Picture Problems)**

There is a big tree in Takashi's house. The heights of his father, mother and him are 50%, 39% and 31% of the tree. One day, a hat was blown away and was caught on the tree. Someone examined the height of the hat, and found that the father's height was 92cm shorter than the place where the hat was caught, and even the total of Takashi's and his mother's height was 20 cm shorter than where the hat was. How tall was the point where the hat was caught?

**Weight Problem (One of the Bar Chart Problems)**

There are three people, called A, B and C, and they examined their weights. When D asked about the result, one of them replied as follows: "The weight of C is 35% of the total of the three people. B is 2 kg lighter than C, and A is 5 kg lighter than B". How heavy was A?

**Video Problem (One of the Table Problems)**

A community center sometimes shows videotapes for children. There are 2 kinds of videotapes and one type is 120 minutes and another is 180 minutes. Someone investigated how many tapes were used in one year. The result showed the ratio of the longer tape to the shorter tape was 4:1. In addition to this, someone examined the total showing time. It revealed that the difference between the total showing times of the longer tape and the shorter tape was 24,000 minutes. How many longer tapes were shown in that year?

# Hi-tree Layout Using Quadratic Programming

Tim Dwyer[1,⋆], Kim Marriott[2,⋆⋆], and Peter Sbarski[3,⋆⋆⋆]

Clayton School of Information Technology,
Monash University, Clayton, Victoria 3800, Australia
Kim.Marriott@infotech.monash.edu.au

**Abstract.** Horizontal placement of nodes in tree layout or layered drawings of directed graphs can be modelled as a convex quadratic program. Thus, quadratic programming provides a declarative framework for specifying such layouts which can then be solved optimally with a standard quadratic programming solver. While slower than specialized algorithms, the quadratic programming approach is fast enough for practical applications and has the great benefit of being flexible yet easy to implement with standard mathematical software. We demonstrate the utility of this approach by using it to layout hi-trees. These are a tree-like structure with compound nodes recently introduced for visualizing the logical structure of arguments and of decisions.

## 1 Introduction

Various problems in graph and tree layout have been modelled as a quadratic program (QP): that is, minimization of a convex quadratic objective function subject to linear constraints. In some problems linear constraints are used to ensure non-overlap in the horizontal or vertical direction and the objective function measures the distance between connected nodes. Examples include general graph layout [1], fast layout of large directed graphs using partitioning constraints [2], and horizontal placement of nodes in tree layout [3,4]. QP has also been used to remove node overlap as a post-processing step in general graph layout [5,6]. Again linear constraints ensure non-overlap but now the objective function measures the distance of each node from its original position. However, in virtually all cases specialized algorithms rather then generic off-the shelf QP solvers have been used to solve the problem.

Algorithms for solving QP problems have improved considerably over the last twenty years and there are now polynomial time algorithms (based on interior point methods) that can solve very large problems with hundreds of variables and constraints in a few seconds. Furthermore, high-quality QP solvers are available both as commercial software and as open source projects. Thus, while slower than specialized layout algorithms, as we shall see, a generic QP approach is

⋆ Now at Microsoft Research Lab., Redmond.

⋆⋆ Partially supported by the ARC through Discovery Project Grant DP0987168.

⋆⋆⋆ Now at Vic. Dept. of Innovation, Industry & Regional Development (DIIRD)

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 212–219, 2010.

now more than fast enough for practical applications and has the great benefit of being very simple to implement due to the use of an off-the-shelf solver.

However, the true advantage of using a generic QP approach is its flexibility. It allows the programmer to define layout declaratively by simply specifying goal terms and constraints which, so long as the goal function is quadratic and convex and the constraints linear, can be solved using a QP solver. This means that the designer need not worry about algorithmic details and can rapidly prototype different layout styles.

In this paper we illustrate the advantages of the generic QP approach over the more conventional specialized layout algorithms by means of a case study based on *hi-trees*. These are a tree-like structure that have been developed for visualizing the logical structure of arguments and of decisions (see Figure 1). Two applications we have helped to develop that use hi-trees provide a number of layout styles implemented using specialized layout algorithms. We have re-implemented these different layout styles using an open-source implementation[1] of the Goldfarb and Idnani [7] QP solver. This is an an active-set dual method which has been used to solved a wide variety of convex quadratic problems such as portfolio optimization. We compare efficiency, implementation effort and quality of layout of the QP approach with the original specialised layout algorithms. Our results suggest that the use of a generic QP solver is the method of choice for non-standard layout problems involving layered drawings of tree-like structures.

## 2 Case Study Background

Our case study is layout of *hi-trees* [8]. These are a recently introduced visual representation for hierarchical data in which, depending on the kind of parent

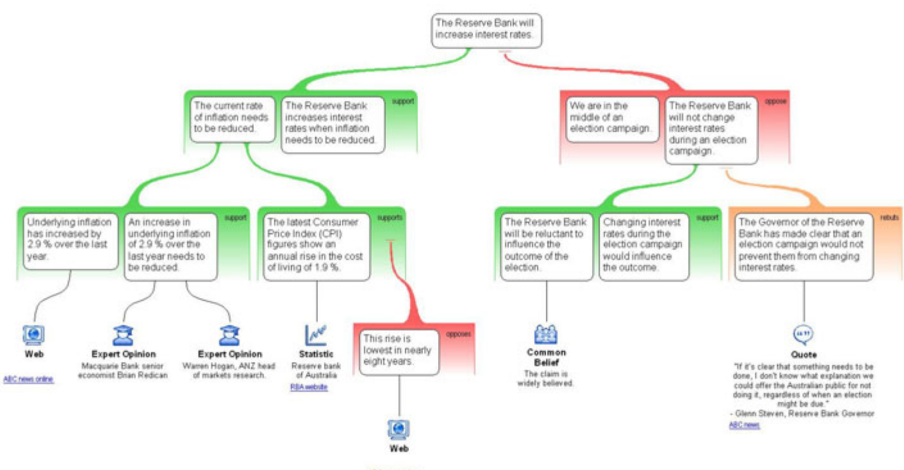



**Fig. 1.** Example hi-tree laid out in the standard layout style. It shows the logical structure of an argument debating a possible interest rate increase.

[1] Many thanks to Luca Di Gaspero `http://www.diegm.uniud.it/digaspero/`

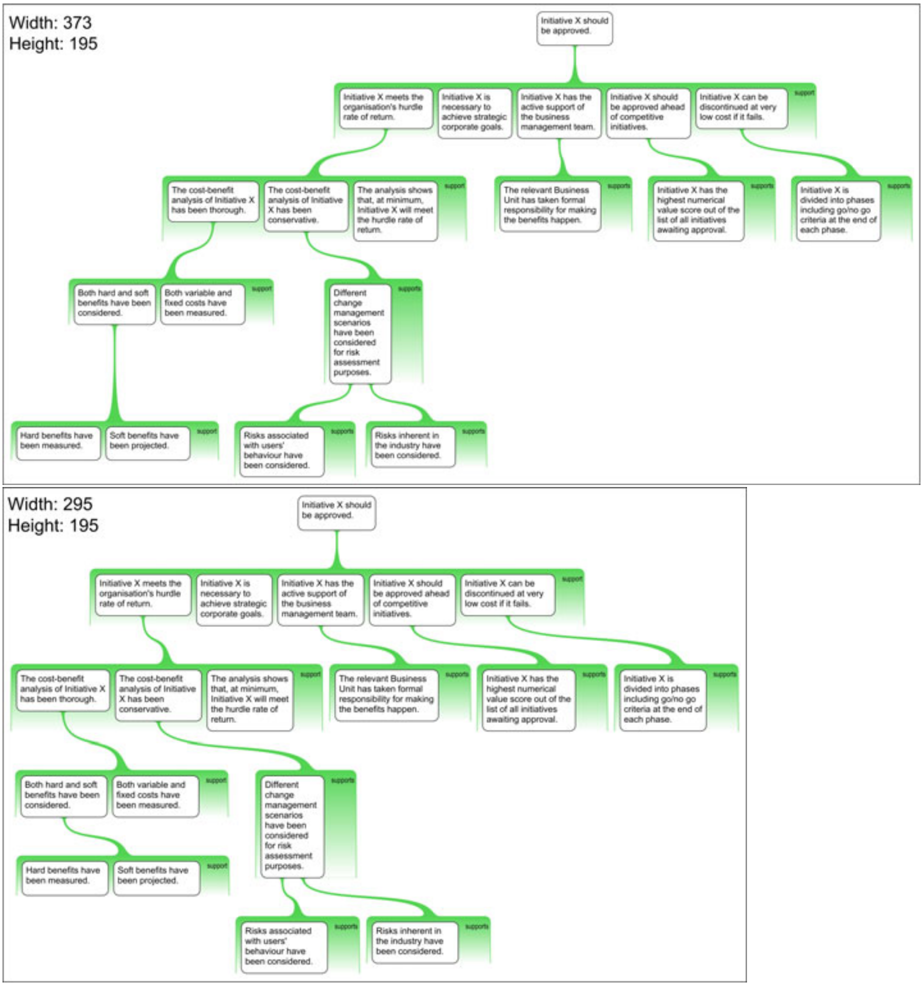



**Fig. 2.** Example of compaction. At the top is the argument map before compaction, on the bottom the layout after horizontal and vertical compression.

node, the child relationship is represented using either containment or links. As an example, consider Figure 1 which shows an argument represented using a hi-tree. The conclusion is at the top and supporting and refuting sub-arguments arranged hierarchically below. Propositions are represented by simple nodes while compound nodes represent a reason.

The advantage of using a hi-tree to represent the logical structure of an argument is that it supports multi-premise reasoning: it is the combination of propositions in the compound node that provide the reason, not each proposition in isolation. The compound node represents the inference rule and can itself have supporting or refuting arguments. For instance, there is an argument opposing the inference that the high current consumer price index (CPI) rate

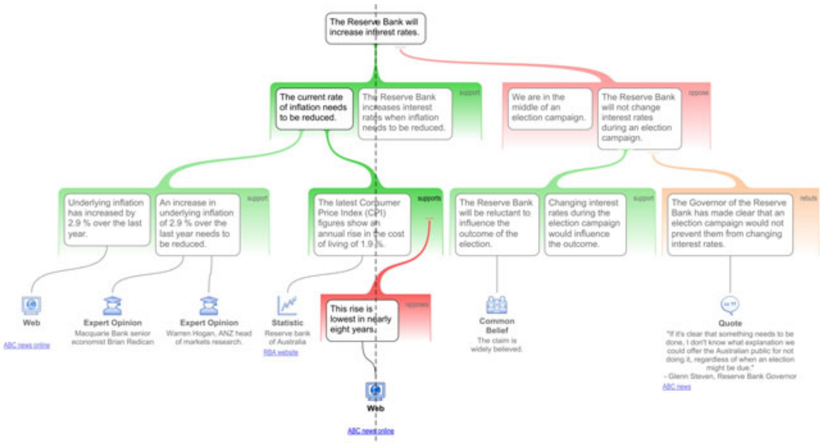



**Fig. 3.** Example of the contextual layout tool *show ancestors* applied to the interest rate argument map from Figure 1

implies that the current inflation rate needs to be reduced. Labels such as "oppose" and "support" indicate the evidential relationship of the reason to its parent proposition. This evidential relationship is also indicated using color.

While hi-trees were originally developed to show the logical structure of arguments, they have also been used to show the structure of decisions. In this case the hi-tree is used to show the initial question, the various options considered when making the decision, and the reasons for and against each option. Austhink Pty Ltd. has developed two tools that utilize hi-trees in these two application domains. The first (*Rationale*) is for construction and visualization of arguments while the second (*bCisive*) is for business decision support. The applications and their layout algorithms are described more fully in [9,8].

Both applications provide a number of different hi-tree layout styles. Figure 1 shows the *standard layout*. As shown in Figure 2, the applications also provide a *compact* layout style in which the layout is compressed both horizontally and vertically. In addition, *Rationale* allows the user to select a node and choose a number of different "contextual layout" tools which modify the layout so as to better show the relationship between the selected node and the contextual nodes. For instance, when a node is selected with the "show ancestors" contextual layout tool, its parent and other ancestors form the context nodes. They are repositioned directly above it thus showing all ancestors in a direct line and all other nodes are pushed away from this particular "path". An example is shown in Figure 3.

Both *Rationale* and *bCisive* use layout algorithms based on extensions to Walker's Algorithm [10] for $n$-ary tree layout. This is based on the Reingold-Tilford [11] algorithm for binary tree layout which recursively lays out the subtrees bottom-up, placing children as close together as possible. Development of these modifications to Walker's Algorithm was non-trivial and required several months of algorithm design and programming. This was because Walker's algorithm is quite complex and also because of the need to encode and experiment with different layout styles.

In the following case study we evaluate the use of quadratic programming for finding hi-tree layouts for the aforementioned layout styles: standard, compact and show ancestors contextual layout.

## 3 Modelling Hi-tree Layout Using QP

In this section we describe how the hi-tree layout styles provided in *Rationale* can be modelled as a quadratic program. The same approach can be used to model the styles provided in *bCisive*. We first formalize a *hi-tree*. Like any ordered tree it consists of *nodes*, $N = \{1, ..., n\}$, and a function $ch : N \rightarrow N^*$ which maps each node to the ordered sequence of nodes which are its children. No node $i$ can be the child of more than one node: the node of which it is a child is called its *parent*. There is one distinguished node, the root node $r$, which has no parent.

The nodes $N$ are partitioned into *proposition nodes* $N_P$ representing a single proposition, and *compound nodes* $N_C$ representing a conjunction of propositions that form a supporting or refuting reason for the parent proposition. It is a bipartite tree: The children of a proposition must be compound nodes and children of a compound node must be propositions. The root is required to be a compound node. We call the children of a compound node its *components*.

The child relationship of a compound node is represented using containment and the child relationship of a simple node is represented using a link. The compound nodes connected by links form the *visual tree*. The *visual level* of a node is its level in the visual tree.

The layout problem is to find a horizontal position $x_i$ and vertical position $y_i$ for the center of each node $i \in N$. However, since the vertical position of the nodes can be computed using a simple algorithm we focus on using QP to determine their horizontal position. Since each compound node is drawn as compactly as possible with its component nodes separated only by a minimum gap $g_c$, the position of each component node $j \in N_P$ is a simple function of its parent node $par(j) \in N_C$: $y_j = y_{par(j)}$ and $x_j = x_{par(j)} + a_j$ where $a_j$ is some fixed offset. Each node $i \in N$ has a width $w_i$ and height $h_i$. The width and height of a proposition node is given by the text and/or image in the node while the width and height of a compound node is computed from the width and height of its components.

**Standard layout:** In standard layout the vertical position of each node $i$ is directly proportional to its *visual level*. Each compound node, therefore, has at most one compound node on the same level to its immediate left and one to its immediate right. We let $left(j)$ and $right(j)$ be the set of compound nodes to the immediate left and to the immediate right, respectively, of compound node $j$.

The horizontal position of the nodes in the standard layout can be found by solving the following QP which minimizes the distance between each compound node and the children of its component propositions:

Minimize $\sum_{c \in N_C} \sum_{p \in ch(c)} \sum_{c' \in ch(p)} wt_{c'}((x_c + a_p) - x_{c'})^2$ where $wt_{c'}$ is a weighting constant based on the size of the sub-tree with root $c'$. This is subject to the constraints

- $x_r = 0$ and
- for each node $c \in N_C$ and $c' \in left(c)$, we impose $x_{c'} + \frac{w_{c'}}{2} + g \leq x_c - \frac{w_c}{2}$

The first constraint fixes the position of the root node and the remaining constraints ensure that nodes on the same level are separated by a minimum gap $g$.

**Compact layout:** For the compact layout the (simple) algorithm from [8] is used to find the vertical position for the compound nodes. This splits layers to allow the children of relatively short nodes to move closer to their parents. As a result, a node can now span multiple layers and so may have more than one immediate neighbour to the left and to the right. We extend the definition of $left(c)$ and $right(c)$ to include all of these immediate neighbours.

Horizontal compaction of the layout works by choosing a certain width $W$ for the layout and then finding a layout in which the nodes in each layer are placed between 0 and $W$ (we assume $W$ is larger than the width of the nodes in the widest layer). $W$ of course could be the page width. Thus the horizontal position of the nodes in the compact layout can be found by solving the following QP: Minimize $\sum_{c \in N_C} \sum_{p \in ch(c)} \sum_{c' \in ch(p)} wt_{c'}((x_c + a_p) - x_{c'})^2$ subject to the constraints

- for each node $c \in N_C$ and each $c' \in left(c)$, we impose a minimum gap between the nodes $x_{c'} + \frac{w_{c'}}{2} + g \leq x_c - \frac{w_c}{2}$
- for each node $c \in N_C$, if it is the leftmost node on the level, i.e., $left(c) = \emptyset$ then we impose $0 \leq x_c - \frac{w_c}{2}$
- for each node $c \in N_C$, if it is the rightmost node on the level, i.e.,$right(c) = \emptyset$, then we impose $x_c + \frac{w_c}{2} \leq W$

**Contextual layout:** It is simple to add contextual layout such as showing a node's ancestors to both standard and compact layout. We simply need to add an equality constraint between the horizontal position of the selected node and its ancestors. In addition we need to appropriately modify the minimum allowed gap between contextual nodes and non-contextual nodes.

## 4 Evaluation

The QP approach gives layouts that are very similar to those obtained with the original specialized hi-tree layout algorithms. All argument maps shown in this paper were laid out using QP. In the case of compact layout the QP approach is more robust since the specialized algorithm is, in rare cases, not guaranteed to obtain maximal horizontal compaction [8]. The QP approach was also considerably easier to implement. The specialized algorithms required a non-trivial extension to the Walker algorithm and resulted in over 2900 lines of code while the QP approach required only 240 lines of straightforward code. (The algorithms were implemented using Microsoft Visual C# Compiler version 8.00.50727.42 for Microsoft .NET Framework version 2.0.50727 under the Windows XP Service Pack 2 operating system.)

We also compared the speed of the QP approach with that of the original specialized hi-tree layout algorithms. We laid out a number of random hi-trees ranging from 100 to 400 compound nodes. We measured layout time in seconds for standard layout and also for a compact layout with maximum vertical and horizontal compaction:

| # Compound | Standard Layout | | Compact Layout | |
|---|---|---|---|---|
| Nodes | Spec. Algm | QP | Spec. Algm | QP |
| 100 | 0.010 | 0.038 | 0.010 | 0.041 |
| 150 | 0.022 | 0.049 | 0.022 | 0.053 |
| 200 | 0.026 | 0.071 | 0.031 | 0.080 |
| 250 | 0.042 | 0.120 | 0.048 | 0.126 |
| 300 | 0.059 | 0.235 | 0.064 | 0.254 |
| 350 | 0.081 | 0.401 | 0.091 | 0.402 |
| 400 | 0.106 | 0.517 | 0.120 | 0.531 |

In both cases the QP approach while slower (as expected) is still fast enough for real-time application for large hi-trees. All experiments were run on a 1.83 GHz Intel Centrino with 1GB of RAM.

Thus, our case study provides support for the original claim that generic off-the-shelf QP solvers provide a flexible, practical method for non-standard layout problems involving layered drawings of graphs and trees.

## References


1. Dwyer, T., Koren, Y., Marriott, K.: IPSep-CoLa: An incremental procedure for separation constraint layout of graphs. IEEE Transactions on Visualization and Computer Graphics 12(5), 821–828 (2006)
2. Dwyer, T., Koren, Y., Marriott, K.: Drawing directed graphs using quadratic programming. IEEE Transactions on Visualization and Computer Graphics 12(4), 536–548 (2006)
3. He, W., Marriott, K.: Constrained graph layout. Constraints 3, 289–314 (1998)
4. Bohanec, M.: DEXiTree: A program for pretty drawing of trees. In: Proc. Information Society IS 2007, pp. 8–11 (2007)
5. Marriott, K., Stuckey, P., Tam, V., He, W.: Removing node overlapping in graph layout using constrained optimization. Constraints 8, 143–171 (2003)
6. Dwyer, T., Marriott, K., Stuckey, P.J.: Fast node overlap removal. In: Healy, P., Nikolov, N.S. (eds.) GD 2005. LNCS, vol. 3843, pp. 153–164. Springer, Heidelberg (2006)
7. Goldfarb, D., Idnani, A.: A numerically stable dual method for solving strictly convex quadratic programs. Mathematical Programming 26, 1–33 (1983)

8. Marriott, K., Sbarski, P., Gelder, T., Prager, D., Bulka, A.: Hi-Trees and Their Layout. IEEE Transactions on Visualization and Computer Graphics (to appear)
9. Sbarski, P., Gelder, T., Marriott, K., Prager, D., Bulka, A.: Visualizing Argument Structure. In: Proceedings of the 4th International Symposium on Advances in Visual Computing, pp. 129–138 (2008)
10. Walker I, J.Q.: A node-positioning algorithm for general trees. Softw. Pract. Exper. 20(7), 685–705 (1990)
11. Reingold, E.M., Tilford, J.S.: Tidier drawings of trees. IEEE Transactions on Software Engineering 7(2), 223–228 (1981)

# Recognizing the Intended Message of Line Graphs*

Peng Wu[1], Sandra Carberry[1], Stephanie Elzer[2], and Daniel Chester[1]

[1] Computer and Information Science department, University of Delaware
[2] Computer Science department, Millersville University

**Abstract.** Information graphics (line graphs, bar charts, etc.) that appear in popular media, such as newspapers and magazines, generally have a message that they are intended to convey. We contend that this message captures the high-level knowledge conveyed by the graphic and can serve as a brief summary of the graphic's content. This paper presents a system for recognizing the intended message of a line graph. Our methodology relies on 1)segmenting the line graph into visually distinguishable trends which are used to suggest possible messages, and 2)extracting communicative signals from the graphic and using them as evidence in a Bayesian Network to identify the best hypothesis about the graphic's intended message. Our system has been implemented and its performance has been evaluated on a corpus of line graphs.

## 1 Introduction

Information graphics are non-pictorial graphics such as bar charts and line graphs. Although some information graphics are only intended to convey data, the overwhelming majority of information graphics in popular media, such as newspapers and magazines, have a message that they are intended to convey. For example, the line graph in Figure 1 appeared in *USA Today* and ostensibly is intended to convey the message that there has been a recent decrease in box office gross revenue in contrast with the preceding rising trend. We contend that a graphic's intended message constitutes a brief summary of the graphic's high-level content and captures how the graphic should be "understood".

This paper presents our methodology for inferring the intended message of a line graph. In previous research[7], we developed a system for identifying the intended message of a simple bar chart. However, line graphs differ from bar charts in several ways that significantly impact the required processing. First, line graphs are the preferred medium for conveying trends in quantitative data over an ordinal independent axis[12]. Second, as our extensive corpus studies demonstrated, the kinds of messages conveyed by line graphs differ from those conveyed by simple bar charts. For example, the line graph in Figure 2 ostensibly is intended to

* This material is based upon work supported by the National Science Foundation under Grant No. IIS-0534948 and by the National Institute on Disability and Rehabilitation Research under Grant No. H133G080047.

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 220–234, 2010.

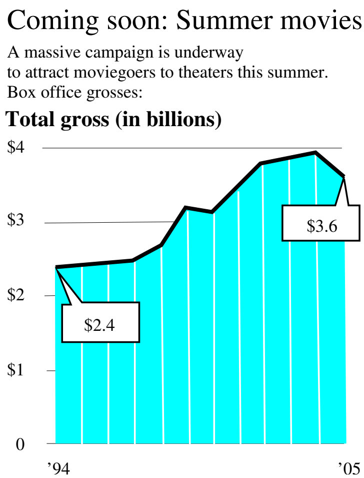



**Fig. 1.** Line Graph from USA Today

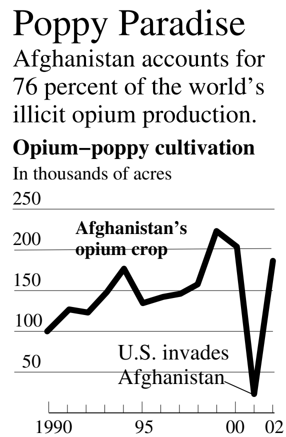



**Fig. 2.** Line Graph from Newsweek

convey a sudden big drop in Afghanistan's opium crop that is not sustained; in our research, we have not encountered a bar chart that conveys a message of this type. Third, although line graphs and bar charts share some of the same kinds of communicative signals, line graphs use other communicative signals that are not found in bar charts. Fourth, recognition of the message conveyed by a line graph relies on the viewer's ability to perceive it as a sequence of visually distinguishable trends rather than a set of discrete data points. Thus we need a method for identifying these trend segments. Moreover, these latter two factors necessitate a different structure and different processing for the message recognition system than was used for bar charts which relied heavily on perceptual task effort.

We are pursuing several projects that utilize a graphic's intended message. Our digital library project will use the graphic's intended message for indexing and retrieving graphics and for taking graphics into account when summarizing multimodal documents. Our assistive technology research is concerned with providing indiviuals with sight impairments with access to graphics in multimodal documents. Other work has tried to render graphics in an alternative form (such as musical tones or tactile images)[1,18] or as verbal descriptions of the appearance and data points in the graph[8]. We are taking a radically different approach. Rather than describing what the graphic looks like, we provide the user with a brief summary based on the graphic's intended message, along with a facility for responding to followup questions about the graphic.

Section 2 describes our overall architecture. Section 3 presents our approach to recognizing the intended message of a line graph. Section 4 presents the results of an evaluation of our implemented system, and Section 5 presents examples of graphics that have been processed by our system. Section 6 discusses related work, and Section 7 presents our conclusions and discusses future work. To our knowledge, our research group is the only effort aimed at automatically recognizing the communicative goal of an information graphic.

## 2 System Architecture

Figure 3 shows our overall architecture. A Visual Extraction Module[3] is responsible for analyzing the graphic and providing an XML representation that captures a sampling of the data points (thereby discretizing a continuous line graph into a set of sampled data points), the axis labels, any annotations on the graphic, the caption, etc. The Caption Tagging Module[6] is responsible for extracting evidence from the caption (see Section 3.3) and producing an augmented XML representation that includes it. The Intention Recognition Module takes as input the augmented XML representation of a graphic and uses a Bayesian Network to identify its intended message. The remainder of this paper focuses on the Intention Recognition Module, which is enclosed by a dashed box in Figure 3.

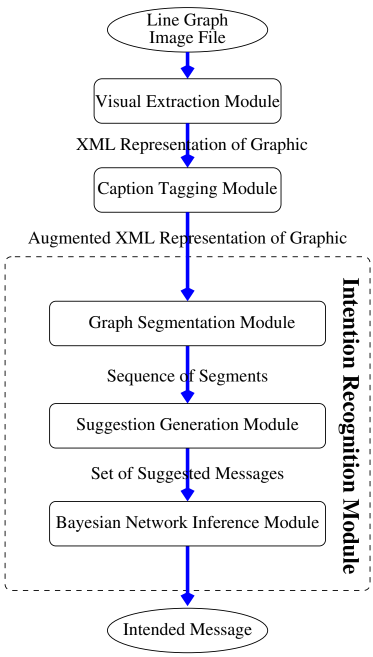



**Fig. 3.** Overall Architecture

## 3 Intended Message Recognition

Clark[4] has noted that language is more than just words; it is any deliberate action (or lack of action when one is expected) that is intended to convey a message. Under this definition of language, information graphics in popular media are a form of language with a communicative goal or intended message. In language understanding research, listeners use the communicative signals present in an utterance (such as cue words, intonation, mood, etc.) to deduce the speaker's intended meaning. We are extending this to the understanding of information graphics. Our methodology relies on extracting communicative signals from the graphic and using them as evidence in a Bayesian Network that hypothesizes the graph designer's communicative goal — i.e., the graphic's intended message. Of course, a graphic might be poorly designed, in which case the message that the graphic conveys might not be what the graph designer intended. But this is true of language in general; for example, if a speaker chooses the wrong words or uses the wrong intonation, his utterance will be misunderstood.

Our methodology for recognizing the intended message of a line graph consists of three steps: 1) segment the line graph into visually distinguishable trends, 2) use this segmentation to suggest possible messages for consideration by a Bayesian Network, and 3) extract communicative signals from the line graph and use them as evidence in the Bayesian Network to identify the graphic's intended message. The following sections discuss each of these steps.

### 3.1 Segmenting a Line Graph into Trends

In recognizing a line graph's intended message, human viewers do not reason about the set of individual data points connected by small line segments. Instead, they appear to treat the line graph as capturing a sequence of visually distinguishable trends. For example, the line graph in Figure 4 consists of many short rises and falls, but a viewer summarizing it would be likely to regard it as consisting of a short overall stable trend from 1900 to 1930, followed by a long rising trend (both with high variance). As observed by Zacks and Tversky[24], this tendency to associate lines with trends exists in part because of cognitive naturalness and in part because of ease of perceptual processing. In comparing bar charts and line graphs, they claim that people "should more readily associate

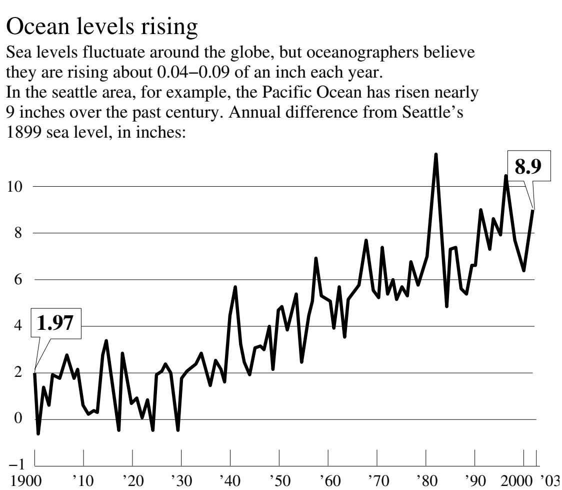



**Fig. 4.** Line graph from USA Today which consists of many short rises and falls

lines with trends because lines connect discrete entities and directly represent slope" and their experiments uphold this prediction. In fact, the cognitive "fit" of the line graph for representing trends is upheld by multiple findings from the basic Gestalt principles to the Wickens and Carswells' Proximity Compatibility Principle (grouping objects that are meant to be processed together) [22] to Pinker's model of graph comprehension [16]. As Zacks and Tversky noted[24], once this cognitive bias is utilized consistently by graph designers, viewers may come to rely on it. This is consistent with our view of graphics as a form of language with communicative signals. The graph designer is attempting to convey a trend, and is trying to make this message as easy as possible for the viewer to extract from the graph. Over time, the choice of graph type itself may become a type of communicative signal.

Our Graph Segmentation Module takes a top-down approach[11] to identifying sequences of rising, falling, and stable segments in a line graph. It starts with the original graphic as a single segment and decides whether it should be split into two subsegments; if the decision is to split the segment, then the split is made at the point which is the greatest distance from a straight line between the two end points of the segment. This process is repeated on each subsegment until no further splits are identified. The Graph Segmentation Module returns a sequence of straight lines representing a linear regression of the points in each subsegment, where each straight line is presumed to capture a visually distinguishable trend in the original graphic.

We used SMO (Sequential Minimal Optimization)[17] for training a support vector machine that makes a decision about whether a segment should be split. 18 attributes, falling into two categories, are used in building the model. The first category captures statistical tests computed from the sampled data points in the XML representation of the graphic. Two examples of such attributes are:

- Correlation Coefficient: The Pearson product-moment correlation coefficient [19] measures the tendency of the dependent variable to have a linearly rising or falling relationship with the independent variable. We hypothesized that the correlation coefficient might be helpful in determining whether a long set of short jagged segments, such as those between 1930 and the end of the graph in Figure 4, should be captured as a single rising trend and thus not be split further.
- Runs Test: The Runs Test estimates whether a regression is a good fit for the data points[2]. A run is a sequence of consecutive sampled points that all fall above the regression line or all fall below the regression line. The number of runs is then compared with an estimate of the expected number of runs $R_{mean}$ and its standard deviation $R_{SD}$; if the actual number of runs exceeds $(R_{mean} - R_{SD})$, then the Runs Test suggests that the regression is a good fit and the segment should not be split. We hypothesize that the Runs Test might be helpful when a segment consists of more than two trends.

The second category of attributes captures explicit features of the segment and the graphic. The following is an example of such an attribute:

- Segment Fraction: This attribute measures the proportion of the total graph that comprises this segment. We hypothesize that segments that comprise more of the total graph may be stronger candidates for splitting than segments that comprise only a small portion of the graph.

We trained our graph segmentation model on a set of 649 instances that required a split/no-split decision. These instances were recursively constructed from a corpus of line graphs: each line graph constituted a training instance and, if that instance should be split, then each of the segments produced by splitting represented other training instances. Using leave-one-out cross validation, in which one instance is used for testing and the other 648 instances are used for training, our model achieved an average success rate of 88.29%.

The output of this Graph Segmentation Module is a sequence of segments that are hypothesized to represent visually distinguishable trends. For example, after the Visual Extraction Module converted Figure 4 from GIF format into an XML representation and the data points were sampled, the Graph Segmentation Module then segmented the data series into two visually distinguishable trends: a relatively stable trend from 1900 to 1930 and a rising trend from 1930 to 2003. As another example, the Graph Segmentation Module segmented the data series produced by the VEM for Figure 7 into two visually distinguishable trends: a rising trend from 1997 to 1999 and a falling trend from 1999 to 2006.

### 3.2 Suggesting Possible Messages

We analyzed a set of simple line graphs collected from various popular media, including magazines such as *Newsweek*, *Time*, and *BusinessWeek* as well as local and national newspapers. We identified a set of 10 high-level message categories that we believe capture the kinds of messages that are conveyed by a simple line graph. Table 1 presents these message categories.

To utilize a Bayesian Network for identifying the intended message of an information graphic, we need a means for suggesting the set of possible messages that should be considered in the network. The Suggestion Generation Module uses the 10 high-level message categories to construct all possible messages from the sequence of segments produced by the Graph Segmentation Module. In addition, we hypothesize that small changes at the end of a line graph, as in Figure 1, may be particularly salient to a viewer, especially if they represent the value of an entity near the current time. However, the Graph Segmentation Module will most likely smooth such small changes into an overall longer smoothed trend. Thus, a short routine using a statistical test is run that examines the end of the line graph and if it represents a change in slope from the preceding points, that short portion is treated as a separate segment. This short segment (if any) is merged with the result produced by the Graph Segmentation Module, and a Contrast-Trend-Last-Segment (CSCT) and a Contrast-Segment-Change-Trend

**Table 1.** Categories of High Level Messages for Line Graphs

| Intention Category | Description |
|---|---|
| **RT: Rising-Trend** | There is a rising trend from <param$_1$ > to <param$_2$ > |
| **FT: Falling-Trend** | There is a falling trend from <param$_1$ > to <param$_2$ > |
| **ST: Stable-Trend** | There is a stable trend from <param$_1$ > to <param$_2$ > |
| **CT: Change-Trend** | There is a <slope$_2$ > trend from <param$_2$ > to <param$_3$ > that is significantly different from the <slope$_1$ > trend from <param$_1$ > to <param$_2$ > |
| **CTLS: Contrast-Trend-Last-Segment** | There is a <slope$_2$ > segment from <param$_2$ > to <param$_3$ > that is not long enough to be viewed as a trend but which is different from the <slope$_1$ > trend from <param$_1$ > to <param$_2$ > |
| **CTR: Change-Trend-Return** | There is a <slope$_3$ > trend from <param$_3$ > to <param$_4$ > that is different from the <slope$_2$ > trend between <param$_2$ > and <param$_3$ > and reflects a return to the kind of <slope$_1$ > trend from <param$_1$ > to <param$_2$ > |
| **CSCT: Contrast-Segment-Change-Trend** | There is a <slope$_3$ > segment from <param$_3$ > to <param$_4$ > that is not long enough to be viewed as a trend but which suggests a possible return to the kind of <slope$_1$ > trend from <param$_1$ > to <param$_2$ > which was different from the <slope$_2$ > trend from <param$_2$ > to <param$_3$ > |
| **BJ: Big-Jump** | There was a very significant sudden jump in value between <param$_1$ > and <param$_2$ > which may or may not be sustained |
| **BF: Big-Fall** | There was a very significant sudden fall in value between <param$_1$ > and <param$_2$ > which may or may not be sustained |
| **PC: Point-Correlation** | There is a correlation between changes at {<param$_1$ >,...,<param$_n$ >} and the text annotations {<annot$_1$ >,...,<annot$_n$ >} that are associated with these points. |

(CSCT) message are proposed for the last two or three segments of the graphic respectively.

Consider, for example, the graphic in Figure 5. The Graph Segmentation Module produces a sequence of three visually distinguishable segments. The Suggestion Generation Module proposes the following 11 possible messages[1]:

RT (5-21-05, 9-1-05)
RT (12-1-05, 4-25-06)
FT (9-1-05, 12-1-05)
BJ (5-21-05, 9-1-05)
BF (9-1-05, 12-1-05)
CT (5-21-05, 9-1-05, 12-1-05)
CT (9-1-05, 12-1-05, 4-25-06)
CTR (5-21-05, 9-1-05, 12-1-05, 4-25-06)
CTLS (9-1-05, 12-1-05, 4-25-06)
CSCT (5-21-05, 9-1-05, 12-1-05, 4-25-06)
PC (5-21-05, 9-1-05, 12-1-05, 4-25-06)

[1] Our system works with the actual points in the graph; for clarity of presentation, we only show the x-values for the points corresponding to <param$_i$ > in Table 1.

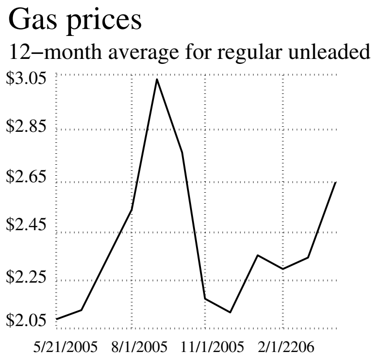



**Fig. 5.** Line graph from a local newspaper

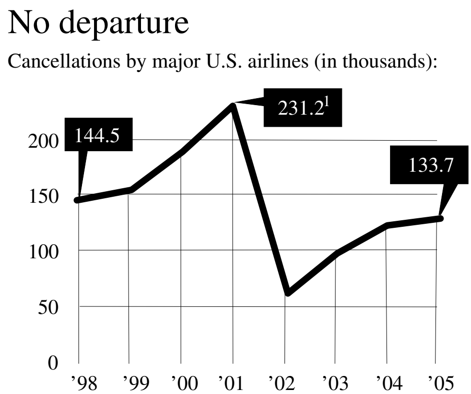



**Fig. 6.** Line graph from USA Today with multiple annotations. (This graphic appeared on a slant in its original form.)

### 3.3 Communicative Signals as Evidence

Just as listeners use evidence to identify the intended meaning of a speaker's utterance, so also must a viewer use evidence to recognize a graphic's intended message. We hypothesize that if the graphic designer goes to the effort of entering attention-getting devices into a graphic to make one or more of the entities in the graphic particularly salient, then the designer probably intends for these entities to be part of the graphic's intended message. There are several ways in which a graphic designer explicitly makes an entity in a line graph salient.

The graphic designer may annotate a point on a line graph with a value or a piece of text. This draws attention to that point in the line graph and serves as evidence that the point plays a role in the graphic's intended message. Consider the graphic in Figure 6. Three points in the graphic are annotated with their value. This suggests that these points are particularly important to the graphic's intended message — in terms of our representation, the points might serve as parameters of the graphic's intended message. This provides strong evidence for a Change-Trend-Return('98,'01,'02,'05) message since three of the four parameters of the message are salient in the graphic. Similarly, consider Figure 2. The low point in the graphic is annotated with text, suggesting that it is important to the graphic's message. This annotation might provide evidence for a Big-Fall(00,01) or for a Change-Trend-Return (where the annotation is on the point where the return begins), among others. The Visual Extraction Module is responsible for producing an XML representation of a graphic that indicates any annotated points and their annotations.

A point in the line graph can also become salient by virtue of its being referenced by a noun in the caption. This can occur by the caption referring to its x-axis value or even to its y-value, although the latter occurs less often. For example, if the caption on the graphic in Figure 2 were *"Poppies Missing in 01"*, the reference to the year "01" would lend salience to the low point in the graphic

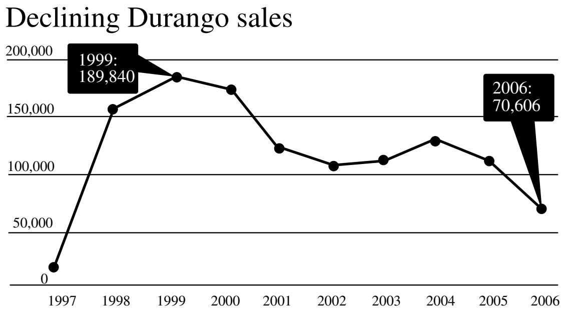



**Fig. 7.** Line graph from a local newspaper

even if it were not annotated. The Caption Tagging Module is responsible for augmenting the XML representation of a graphic so that it indicates any points that are referenced by nouns in the caption.

Certain parts of a graphic become salient without any effort on the part of the graphic designer. For example, a viewer's attention will be drawn to a sudden large rise or fall in a line graph. Similarly, a viewer will be interested in the segment at the end of a line graph since it captures the end of the quantitative changes being depicted. Although no specific effort is required by the graph designer, we posit that it is mutually believed by both graph designer and viewer that such pieces of the graphic will be salient. Our system extracts such evidence by analyzing the segments produced by the Graph Segmentation Module and using their slopes, their relative change with respect to the range of y-values in the graph, and their positions in the graphic as evidence.

Captions are often very general and do not capture a graphic's intended message[6]. For example, the caption on the graphic in Figure 2 fails to capture its message that there was a sudden big fall (that was not sustained) in Afghanistan opium production. Moreover, even when a caption conveys some of the graphic's message, it is often ill-formed or requires extensive world knowledge to understand. However, as in our work on simple bar charts, we have found that verbs in a caption often suggest the general category of the graphic's message. Adjectives and adverbs function similarly. For example, the adjective *"declining"* in the caption of Figure 7 suggests a Falling-Trend message or perhaps a Change-Trend message where the trends change from rising to falling.

Using WordNet, we identified potentially helpful verbs and organized them into classes of similar verbs. For example, the verbs *"jump"* and *"boom"* reside in one verb class, whereas the verbs *"resume"* and *"recover"* reside in a different verb class. The Caption Tagging Module is responsible for extracting such helpful words from the caption and augmenting the XML representation of the graphic to indicate the presence of any of our six identified verb classes.

Several features of the segments comprising a suggested message also provide evidence for or against that proposed message being the intended message of the graphic. The graphic designer presumably had a reason for including all of the points in a line graph. Thus the fraction of a line graph covered by the segments comprising a suggested message serves as evidence about whether that was the graphic designer's intended message — presumably, messages that cover much of the line graph are more likely to be the designer's intended message. (However, the intended message need not cover the entire graphic. For example, it appears that when conveying a Rising-Trend or a Falling-Trend, the graphic designer sometimes includes a small segment of points prior to the start of the trend in order to keep the viewer from inferring that the rise or fall might have started at earlier points not depicted in the graphic.)

### 3.4 The Bayesian Network

A Bayesian Network is a probabilistic reasoning system that can take into account the multiple pieces of evidence in a line graph in order to evaluate the various message candidates proposed by the Suggestion Generation Module. Rather than identifying an intended message with certainty, a Bayesian Network gives us the posterior probability of each candidate message, thereby reflecting any ambiguity in the graphic. In our project, a new Bayesian Network is built dynamically (using Netica[15]) each time a line graph is processed. The top-level node of our Bayesian Network represents each of the possible high-level message categories, such as Change-Trend or Big-Jump. Each of these high-level message categories appears as a child of the top-level node; this is purely for ease of representation. The children of each of these high-level message category nodes are the suggested messages (with instantiated parameters) produced by the Suggestion Generation Module.

Once nodes for each of the messages suggested by the Suggestion Generation Module have been added to the Bayesian Network, evidence nodes are entered into the network to reflect the evidence for or against the different suggested messages. Verb and adjective/adverb evidence suggest a general category of message, such as Rising-Trend or Change-Trend; thus they are attached as children of the top-level node in the Bayesian Network. Other evidence, such as whether there are annotations and whether they correspond with parameters of a message, serve as evidence for or against each suggested message; thus these evidence nodes are entered as children of each suggested message node.

Associated with each node in a Bayesian Network is a conditional probability table that reflects the probability of each of the values of that node given the value of the parent node. (The conditional probability table for the top-level node captures the prior probabilities of each of the message categories.) To construct the conditional probability tables, each line graph in our corpus of 215 line graphs was first annotated with its intended message as identified by consensus among three coders; we then analyzed each line graph to identify the evidence that was present, and computed the conditional probability tables from this analysis. One such conditional probability table is shown in Table 2. It gives the conditional probability that the endpoints $<param_1>$ and $<param_2>$ of a Rising-Trend($<param_1>$, $<param_2>$) message are annotated in the graphic, given that the intended message is (or is not) a Rising-Trend. For example, the

**Table 2.** A sample conditional probability table

Endpoints Annotated Table

| Rising-Trend(<param$_1$ >, <param$_2$ >) | InPlan | NotInPlan |
|---|---|---|
| Only one endpoint is annotated | 12.3% | 26.2% |
| Both endpoints are annotated | 55.4% | 3.6% |
| No endpoint is annotated | 32.3% | 70.2% |

InPlan column of the conditional probability table shows that the probability that both endpoints are annotated is 55.4% if a Rising-Trend is the intended message, and the NotInPlan column shows that the probability is 3.6% if it is not the intended message.

## 4 Evaluation of the System

We evaluated the performance of our system for recognizing a line graph's intended message on our corpus of 215 line graphs that were collected from various magazines such as *Newsweek*, *BusinessWeek*, and from local and national newspapers. Input to the Intention Recognition Module is the augmented XML representation of a graphic. We used leave-one-out cross validation in which each of the graphics is used once as the test graphic, with the conditional probability tables computed from the other 214 graphics. Our system recognized the correct intended message with the correct parameters for 157 line graphs, which gave us a 73.0% overall success rate.

The system's errors are primarily due to sparseness of data. For example, if we have only one graphic where a particular verb class is used to indicate an intention category, then leave-one-out cross validation has no means to connect the verb class with that intention category and we are likely to get an incorrect result when hypothesizing the intended message of that graphic. In addition, if the Graph Segmentation Module does not produce the correct segmentation of a graphic, the Suggestion Generation Module is unlikely to produce a set of suggested messages that includes the graphic's intended message, and thus the Bayesian Network will not correctly hypothesize it. Therefore to improve the performance of our intention recognition system, we are working on identifying additional attributes that can produce a better learned model for graph segmentation, and we are collecting additional line graphs for training our Bayesian Network. However, even when our system does not produce the ideal result, the message hypothesized by our system still reflects the information in the graphic.

## 5 Examples

Consider the graphic in Figure 5. As described in Sections 3.2, our Graph Segmentation Module hypothesizes that the graphic consists of three visually distinguishable trends and our Suggestion Generation Module suggests a set of

11 possible messages for consideration by the Bayesian network. The Bayesian network hypothesizes that the graphic is conveying a Change-Trend-Return message — in particular, that the trend in gas prices changed in 9/1/05 (from rising to falling) but returned in 12/1/05 to its previous trend (rising). The system assigns this message a probability of 98.7% indicating that it is very confident of its hypothesis. Next consider the line graph in Figure 4 which illustrates the processing of a line graph consisting of a large number of short line segments. Our Graph Segmentation Module segments this line graph into two visually distinguishable trends, and the Bayesian network hypothesizes that the graphic conveys a changing trend from relatively stable between 1900 and 1930 to rising between 1930 and 2003 and assigns this hypothesis a probability of 99.9%

Now let us consider two graphics where the system is less certain about its hypothesized messages. In the case of the graphic in Figure 6, the system hypothesizes that the graphic is conveying a Change-Trend-Return in cancellations by major U.S. airlines (rising, then falling, then returning to a rising trend) and assigns the hypothesis a probability of 63.1%. However, the system also assigns a probability of 36.1% to the hypothsis that the graphic's intended message is that there was a big fall in cancellations between 2001 and 2002. The system prefers the change-trend-return hypothesis due to the stronger evidence — for example, there is no annotation on the low point at 2002 (thereby suggesting that the fall is not the primary message of the graphic), and there are annotations on other points in the graphic (thereby suggesting that those points should be parameters of the message).

As a second example where the system is less certain about its hypothesized message, consider the graphic in Figure 7 and two of the suggestions proposed by the Suggestion Generation Module: a Change-Trend(1997,1999,2006) and a Falling-Trend(1999,2006). There are a number of communicative signals in the graphic that were deliberately entered by the graph designer: 1) the annotation giving the value for the year 1999, 2) the annotation giving the value for the year 2006, and 3) the adjective *"declining"* in the caption *"Declining Durango sales"*. Other evidence entered into the Bayesian Network includes (among others) the portion of the graphic covered by each suggested message, and the relative width of the last segment of each message. For the Change-Trend message, the message covers the whole graphic and the last segment covers more than half of the graphic; for the Falling-Trend message, the last (and only) segment covers much, but not all, of the graphic.

The system considers all of the suggested messages and the evidence entered into the Bayesian Network; it hypothesizes that the graphic's intended message is that there is a falling trend in Durango sales from 1999 to 2006 and assigns this hypothesis a probability of 54.06%. The hypothesis that the graphic is intended to convey a Change-Trend (rising from 1997 to 1999 and then falling from 1999 to 2006) is assigned a probability of 45.90%. All the other suggested messages share the remaining 0.04% probability. The probabilities assigned to the Falling-Trend and Change-Trend messages reflect the ambiguity about the intended message that is inherent in the graphic. The presence of the adjective *"declining"* and the annotations on both points that are parameters of the Falling-Trend message, but only annotations on two of the three points that are parameters of the Change-Trend message, caused the system to prefer the Falling-Trend message over the Change-Trend message. Notice that while the graphic in Figure 7 does show a short rising segment prior to the long falling trend from 1999 to 2006, the focus of the graphic is on the falling trend rather than on a change in trend. (Production of Durango cars only started in 1997, so the first part of the graph primarily reflects the "ramp up" in initial sales, not a changing trend.)

Now let us examine how the system's hypothesis changes as we vary the communicative signals in the graphic. Suppose that we add an extra annotation giving the value of Durango sales in 1997. Now the system's hypothesis changes dramatically — it identifies the Change-Trend as the intended message of the graphic and assigns it a probability of 99.5%, with the Falling-Trend message assigned a probability of 0.5%. Note that although the adjective *"declining"* is most associated with a Falling-Trend message, it can also be used with a Change-Trend message to draw attention to the falling portion of the changing trend.

Now let's return to the original graphic in Figure 7 with only two points annotated, but let's change the caption to *"Durango sales changed"*. Whereas *"declining"* might be used in the caption of a Change-Trend message, it is less likely that the verb *"changed"* would be used with a Falling-Trend message. Once sgain, the system hypothesizes that the graph is intended to convey the changing trend in Durango sales, rising from 1997 to 1999 and then falling from 1999 to 2006, but only assigns it a probability of 95.2% due to the ambiguity resulting from only two points being annotated.

## 6 Related Work

Shah et. al.[20] had people describe line graphs to examine how the graph design affects what people get as the message of the graphic. Our work used Bayesian network to reason about the messages of the graphic from the evidences, which implemented the automated recognition of line graph's messages.

Yu et. al.[23] developed a pattern recognition algorithm for summarizing interesting features of automatically generated time-series data such as from a gas turbine engine. However, they were analyzing automatically generated machine data, not graphs designed by a graphic designer whose intention was to convey a message to the viewer. Futrelle and Nikolakis[10] developed a constraint grammar for parsing vector-based visual displays and producing structured representations of the elements comprising the display. The goal of Futrelle's work is to produce a graphic that is a summary of one or several more complex graphics[9]. Note that the end result will again be a graphic, whereas our goal is to recognize a graphic's intended message.

A number of researchers have studied the problems of classifying time series data into a pattern category[14] or judging the similarity between time-series data[13,21]. Their main goal is to identify the pattern of a query time series by calculating its similarity with a predefined pattern. Dasgupta et.al.[5] identify anomalies or events in a data series. Our research differs from these efforts in that we are segmenting line graphs into visually distinguishable trends that can be used to suggest possible messages for consideration by a system that recognizes the graphic's intended message.

## 7 Conclusion and Future Work

Information graphics in popular media generally have a message that they are intended to convey, and this message is often not captured by the graphic's caption or given in the accompanying article's text. This paper has presented an implemented and evaluated methodology for identifying the intended message of a line graph. Our methodology involves segmenting the graphic into visually distinguishable trends, extracting communicative signals from the graphic, and using these in a Bayesian Network that hypothesizes the graphic's intended message. The evaluation of our system's performance demonstrates the effectiveness of our approach.

Our current work is using a graphic's recognized message as the basis for summarizing the high-level content of graphics from popular media, in order to provide alternative access for individuals with sight-impairments. We are also investigating the use of the intended message to index and retrieve information graphics, to produce summaries that take into account a multimodal document's information graphics as well as its text, and to extract information from multimodal documents. To our knowledge, our project is the only current research effort to identify an information graphic's intended message and utilize it in processing multimodal documents.

## References


1. Alty, J., Rigas, D.: Communicating graphical information to blind users using music: The role of context. In: Proceedings of CHI 1998, pp. 574–581 (1998)
2. Bradley, D.C., Steil, G.M., Bergman, R.N.: OOPSEG: a data smoothing program for quantitation and isolation of random measurement error. Computer Methods and Programs in Biomedicine 46, 67–77 (1995)
3. Chester, D., Elzer, S.: Getting computers to see information graphics so users do not have to. In: Proc. of the International Symposium on Methodologies for Intelligent Systems, pp. 660–668 (2005)
4. Clark, H.: Using Language. Cambridge University Press, Cambridge (1996)
5. Dasgupta, D., Forrest, S.: Novelty detection in time series data using ideas from immunology. In: Neural Information Processing Systems Conference (1996)
6. Elzer, S., Carberry, S., Chester, D., Demir, S., Green, N., Zukerman, I., Trnka, K.: Exploring and exploiting the limited utility of captions in recognizing intention in information graphics. In: Proc. of the Association for Computational Linguistics, pp. 223–230 (2005)
7. Elzer, S., Carberry, S., Demir, S.: Communicative signals as the key to automated understanding of bar charts. In: Barker-Plummer, D., Cox, R., Swoboda, N. (eds.) Diagrams 2006. LNCS (LNAI), vol. 4045, pp. 25–39. Springer, Heidelberg (2006)

8. Ferres, L., Verkhogliad, P., Lindgaard, G., Boucher, L., Chretien, A., Lachance, M.: Improving accessibility to statistical graphs: the inspectgraph system. In: Proc. of ACM SIGACCESS Conference on Computers and Accessibility (2007)
9. Futrelle, R.: Summarization of diagrams in documents. In: Mani, I., Maybury, M. (eds.) Advances in Automated Text Summarization, pp. 403–421. MIT Press, Cambridge (1999)
10. Futrelle, R., Nikolakis, N.: Efficient analysis of complex diagrams using constraint-based parsing. In: Proc. of the International Conference on Document Analysis and Recognition, pp. 782–790 (1995)
11. Keogh, E., Kasetty, S.: On the need for time series data mining benchmarks: A survey and empirical demonstration. Data Mining and Knowledge Discovery 7, 349–371 (2003)
12. Levy, E., Zacks, J., Tversky, B., Schiano, D.: Gratuitous graphics? putting preferences in perspective. In: Proceedings of the ACM Conference on Human Factors in Computing Systems, pp. 42–49 (1996)
13. Lin, J., Keogh, E., Lonardi, S., Patel, P.: Finding motifs in time series. In: Proceedings of the Workshop on Temporal Data Mining, pp. 53–68 (2002)
14. Megalooikonomou, V., Wang, Q., Li, G., Faloutsos, C.: A multiresolution symbolic representation of time series. In: Proc. of the International Conference on Data Engineering, pp. 668–679 (2005)
15. N.S.C. Netica (2005)
16. Pinker, S.: A theory of graph comprehension. In: Freedle, R. (ed.) Artificial Intelligence and the Future of Testing, pp. 73–126. Lawrence Erlbaum, Mahwah (1990)
17. Platt, J.C.: Fast training of support vector machines using sequential minimal optimization. In: Advances in Kernel Methods: Support Vector Learning, pp. 185–208 (1999)
18. Ramloll, R., Yu, W., Brewster, S., Riedel, B., Murton, M., Dimigen, G.: Constructing sonified haptic line graphs for the blind student: First steps. In: Proceedings of Assets 2000, pp. 17–25 (2000)
19. Rodgers, J.L., Nicewander, W.A.: Thirteen ways to look at the correlation coefficient. The American Statistician 42, 59–66 (1988)
20. Shah, P., Mayer, R.E., Hegarty, M.: Graphs as aids to knowledge construction: Signaling techniques for guiding the process of graph comprehension. Journal of Educational Psychology 91(4), 690–702 (1999)
21. Toshniwal, D., Joshi, R.: Finding similarity in time series data by method of time weighted moments. In: Proc. of Australasian Database Conference, pp. 155–164 (2005)
22. Wickens, C.D., Carswell, C.M.: The proximity compatibility principle: Its psychological foundation and relevance to display design. Human Factors 37(3), 473–494 (1995)
23. Yu, J., Reiter, E., Hunter, J., Mellish, C.: Choosing the content of textual summaries of large time-series data sets. Natural Language Engineering 13, 25–49 (2007)
24. Zacks, J., Tversky, B.: Bars and lines: A study of graphic communication. Memory & Cognition 27(6), 1073–1079 (1999)

# Mapping Descriptive Models of Graph Comprehension into Requirements for a Computational Architecture: Need for Supporting Imagery Operations

B. Chandrasekaran and Omkar Lele

Department of Computer Science and Engineering
The Ohio State University
Columbus, OH 43210 USA
chandra@cse.ohio-state.edu, omkar.lele@osumc.edu

**Abstract.** Psychologists have developed many models of graph comprehension, most of them descriptive, some computational. We map the descriptive models into requirements for a cognitive architecture that can be used to build predictive computational models. General symbolic architectures such as Act-R and Soar satisfy the requirements except for those to support mental imagery operations required for many graph comprehension tasks. We show how Soar augmented with DRS, our earlier proposal for diagrammatic representation, satisfies many of the requirements, and can be used for modeling the comprehension and use of a graph requiring imagery operations. We identify the need for better computational models of the perception operations and empirical data on their timing and error rates before predictive computational models can become a reality.

**Keywords:** graph comprehension, cognitive architectures, diagrammatic reasoning, mental imagery.

## 1 Graph Comprehension Models

In this paper, we investigate the requirements for a cognitive architecture that can support building computational models of graph comprehension. We start by reviewing research on graph comprehension models that psychologists and cognitive scientists have built over the last three decades.

***High-Level Information Processing Accounts.*** Bertin [1] proposed a semiotics-based task decomposition that anticipated the later information-processing accounts of Kosslyn [2] and Pinker [3]. These accounts provide a framework to place the rest of the research. Because of their information-processing emphasis and consequent greater relevance to computational modeling, we focus on [2] and [3]. Their proposals have much in common, though they differ in their emphases and details. They envision a process that produces a representation ("perceptual image" for Kosslyn, and "visual description" for Pinker) in visual working memory (WM), respecting Gestalt Laws and constraints such as distortions and discriminability. In [3], the visual

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 235–242, 2010.

description is an organization of the image into visual objects and groups of objects – lines, points, regions, and abstract groupings corresponding to clusters. It is not clear if this visual description is purely symbolic or if it retains shape information, as Kosslyn's images do, but keeping the shape information is essential, as we shall see.

In both models, the construction of this internal visual representation is initially a bottom-up process, but soon after, it is a sequential process in which top-down and bottom-up processes are opportunistically combined: the state of the partial visual representation at any stage and the agent's goals trigger the retrieval of relevant knowledge from Long-Term Memory (LTM), which in turn directs further processes.

Pinker's account of retrieval of goal- and state-relevant information from LTM uses the idea of "schemas" (or "frames" as they are often called in AI), organized collections of knowledge about the structure of graphs, in general as well as for various graph types. Comprehension of the specific graph proceeds by filling in the "slots" – the graph type, the axes, the scales, quantities represented, etc. – in the schemas. The schema knowledge guides the agent's perception in seeking the information needed to perform this slot-filling.

Shah et al. [4][1] propose that graph comprehension can be divided, at a high level, into two primary processes – a pattern recognition process followed by a bottom-up and top-down integrative cognitive process, an account consistent with the Pinker/Kosslyn accounts. Their research on attention during this process highlights the role of domain-specific background knowledge in guiding the comprehension process. Their model includes an account of learning: novice graph users, in contrast to experts, tend to spend more time understanding the graph structure (how the graph represents information). In the schema language, learning results in the details of the general schema being filled in so that only the problem-specific instantiation needs to take place, considerably speeding up the process.

***Perception.*** Graphs contain information that is encoded using graphical properties such as position, size, and angle. The accuracy of the information obtained from the graph depends on the how well humans can decode the encoded information. Cleveland and McGill (see, e.g., [5]) identify a set of "elementary graphical perception tasks," that is, tasks which they propose are performed instantaneously with no apparent mental effort. They order them in terms of accuracy: Position along common scale, Position on identical but non-aligned scales, Length, Angle or Slope (with angle not close to 0, $90^o$ or $180^o$), Area, Volume, Density, and Color Saturation. Simkin and Hastie [6] claim that the ordering is not absolute, but changes as the graphs and conditions change. They further argued that these judgment tasks were not always instantaneous, but often required a sequence of mental imagery operations such as anchoring, scanning, projection and superimposition.

Human perception is good at instantly perceiving certain properties, e.g., $90^o$ angle, the midpoint of a line segment. *Anchoring* is the use of such standard perceptions to estimate properties that are not standard. For example, estimating the distance of a point *p* on a line segment from one end of the line can be done by a series of midpoint perceptions of segments containing *p*. Relative lengths of segments from *p* to the two ends of the line can be estimated by the relative *scanning* durations. *Projection* is

[1] Due to space limitations, we only cite a subset of relevant papers by many authors. The bibliographies of the cited references contain pointers to other relevant work by the same authors.

when a ray is sent out from one point in the image to another point for the purpose of securing an anchor. *Superimposition* is when one object is mentally moved onto another object, such as when a smaller bar might be mentally moved on to a longer one so that their bottoms align, as part of estimating the ratio of their lengths.

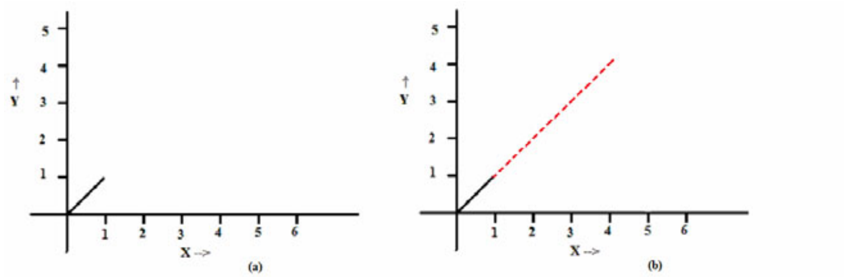



**Fig 1.** Example from [10]: Need to imagine line extension to answer question extended by the user

Gillan et al. (see, e.g., [7]) describe a model for graph comprehension, in which in addition to external perception, imagery operations are used, such as rotate and move image objects, and mental comparison of lengths of lines. Trafton et al. (see e.g., [8]) draw attention to how the individual steps of information extraction from a representation are integrated to answer a question. Their cognitive integration is functionally akin to the schema-based comprehension account of Pinker, while their visual integration includes spatial transformations and imagination. Fig. 1 is an example, where the user has to extend a line in their imagination to answer a question about a value in the future.

***Computational Models.*** The models discussed so far have been qualitative and descriptive. The two models we discuss in this section, Lohse's [9] and Peebles and Cheng's [10], are both computational and share many features. Both models use a goal-seeking symbolic architecture with working and long-term memories. Both the models assume an agent who knows how to use the graph, that is, the early general schema instantiation steps did not need to be modeled. That still leaves many visual search, perceptions and information integration tasks that the agent needs to perform. The models share other features as well: neither model has an internal representation of the spatiality of the graphical objects, nor use visual imagination, their examples being simple enough not to need it; and neither of them has a computational version of external perception, whose results are instead made available to the models as needed for simulation. Lohse's model could predict the times taken to answer specific questions based on empirical data on timings for the various operations, and he reported agreement between the predictions and the results, but Foster [11] did not find such an agreement in his experiments using Lohse's model.

The research in [10] used ACT-R/PM, a version of ACT-R that has a visual buffer that can contain the location, but not the full spatiality, of the diagrammatic objects in the external representation. Procedural and declarative knowledge about using the graphs is represented in ACT-R's production rules and propositions. The research also modeled learning the symbol-variable associations and the location information, reducing visual search. Predictions from the model of changes in the reaction times as questions changed matched the human performance qualitatively, but not quantitatively.

***Putting the Models Together.*** The process that takes place when a user is presented with a graph along with some questions to answer can be usefully divided into two stages: one of general comprehension in which the graph type is understood and some details instantiated, and a second stage in which the specific information acquisition goals are achieved. Initial perception organizes the image array into visual elements (Kosslyn, Shah, Trafton), at the end of which working memory contains both symbolic and diagrammatic information, the latter supporting the user's perceptual experience of seeing the graph as a collection of spatial objects in a spatial configuration (Kosslyn). After this, the two stages share certain things in common: a knowledge-guided process involving both bottom-up and top-down operations performing visual and cognitive integrations (Pinker, Shah, Trafton). Perceptions invoke (bottom-up) schema knowledge (Pinker) about graphs and graph types, which in turn guide (top-down) perceptions in seeking information to instantiate the schemas, and the processes are deployed opportunistically. Comprehension initially focuses on identifying the type of graph and then on instantiating the type: the domain, the variables and what they represent, the scales, etc. For information extraction, in simple cases, the user may apply a pre-specified sequence of operations, as in the methods of Lohse and Peebles. In more complex cases, increasingly open-ended problem solving may be necessary for visual and cognitive integration, driven by knowledge in LTM.

While some of the perceptions simply require access to the external image and result in symbolic information to cognition, others require mental imagery operations (Simkin, Gillan, Trafton), such as imagining one or more of the image objects as moved, rotated, or extended, and elements combined or abstracted. Relational perception may be applied to objects some of which are the results of imagery operations. Perception may be instantaneous, or involve visual problem solving.

## 2 Requirements on a Cognitive Architecture to Support Modeling Graph Comprehension and Use

The family of symbolic general architectures, with Soar and Act-R as two best known members, has the architectural features required for building computational models of graph comprehension, with the important exception of supporting mental imagery operations.

*Supporting goal-directed problem solving*. The requirements for cognitive integration – combining bottom-up and top-down activities, representation of schema knowledge, instantiating the schema to build comprehension models, and using schema-encoded information acquisition strategies by deploying appropriate perceptions – can be handled by the architectural features of Act-R and Soar. Schemas are just a higher level of knowledge organization than rules and declarative sentences, and knowledge representation formalisms in Act-R and Soar can be used to implement the schemas. Both architectures have control structures that can produce a combination of bottom-up (information from perception triggering retrieval of new knowledge) and top-down (newly retrieved knowledge creating new perception goals) behaviors. Appropriate knowledge can produce needed cognitive integration. Act-R and Soar also support certain types of learning, with Act-R providing more learning mechanisms than Soar. The available mechanisms can be used to capture some of the observed learning phenomena [4], as demonstrated in [10].

*Supporting Imagery Operations.* For imagery operations, the architectures need a working memory component with a representation functionally organizing the external or internal representation as a spatial configuration of spatial objects, tagging which objects are from the external representations and which belong to imagery. Operations to create imagery objects and configurations should be available: imagery elements may be added afresh, or may be a result of operations such as moving, rotating, or modifying existing objects so they satisfy certain spatial constraints. Diagrammatic perception operations, by which we mean relational and property perceptions after figure-ground, are to be applied to configurations of diagrammatic objects, whether the objects correspond to external objects, imagined objects or a combination. There may also be benefits to having some of the perception operators be always active without the need for cognition to specifically deploy them, so that a certain amount of bottom-up perception is always available. Such bottom-up perceptions can be especially useful in early stages.

*Treating Perceptions as Primitives vs modeling the computational details.* The cognitive mechanisms of Act-R and Soar, especially the former, derive validation to a more or less degree – both from human problem solving experiments and from neuroimaging studies. However, there is really not much in the way of detailed computational models for perception and mental imagery operations, especially ones that would replicate timing and error data. Such computational models would have a role for pre-attentive perception as well, e.g., to explain when certain perceptions are instantaneous and when they require extended visual problem solving. Because of the lack of computational models, one approach is to treat the internal perception and imagery operators as primitives and simply program them without concern about the fidelity of their implementation with respect to human abilities. Models built in this way will be good for certain purposes, e.g., the effect on agent's behavior of the availability or the absence of specific pieces of knowledge and strategies, and not for others, e.g., predict timing data, or perceptual learning. It should be a goal of the modeling community to develop computational models of perception and imagery operations that account for human performance, including pre-attentive phenomena.

*Augmenting Architectures with DRS – A Diagrammatic Representation System.* We propose that the DRS representation and the associated perception and action routines reported in [12] provide the basis for augmenting the architectures in the symbolic family with an imagery capability. The DRS system as it exists only supports diagrams composed of points, curves and regions as elements, which happens to cover most of the graphs. DRS is a framework for representing the spatiality of the diagrammatic objects that result after the early stage of perception has performed a figure-ground separation. DRS is a list of internal symbols for these objects, along with a specification of their spatiality, the intended representation as a point, curve or region, and any explicit labeling of the objects in the external diagram. DRS can also be used to represent diagrammatic objects in mental images, or a combination from external representation and internal images, while also keeping track of each object's provenance. A DRS representation may have a hierarchical structure, to represent any component objects of an object.

The DRS system comes with an open-ended set of imagery and perception operations. The imagery operations can move, rotate or delete elements of a DRS representation, and add DRS elements satisfying certain constraints, to produce new DRS structures. Relational perception operations can be defined between elements of a DRS representation: e.g., Longer(C1, C2), Inside(R1,R2). Operators are also available to detect emergent objects when diagrammatic objects overlap or intersect. Kurup [13] has built biSoar, a bi-modal architecture, in which DRS is integrated with Soar. Matessa [14] has built Act-R models that perform diagrammatic reasoning by integrating DRS with Act-R.

## 4 Building a Computational Model Using DRS

In this section, we will show the functional adequacy of a general symbolic cognitive architecture augmented with DRS to build a computational model for graph comprehension. We use biSoar, but we could have used Act-R plus DRS as well. We implemented the scanning, projection, anchoring and superimposition operators [6], the last three being imagery operations (that is, they create objects in WM corresponding to imaged objects). We treat scanning as instantaneous only for obvious judgments such as 50%. If the proportion is say 70%, the agent we model would perform it by a recursive mid-point algorithm until an estimate within a specified tolerance is reached.

While we modeled several graph usage tasks, we will use the model for the graph in Fig. 1, where a line needed to be mentally extended so as to answer a question about future, "What might the Y value be for x = 4?". The model starts with a DRS representation corresponding to figure-ground separated version of the external representation. This is what perception would deliver to cognition. In this example, the DRS consisted of the curves for the axes and the graph for the x-y function, the scale points, and the point for the origin. For convenience in simulation, the entire DRS representation is not in WM, rather it is kept separately as a proxy for the intensity array version of the external representation. Depending on attention, parts of this DRS will be in WM, along with diagrammatic objects resulting from imagery operations. Certain initial perceptions are automatically deployed at the beginning, that is, without any specific problem solving goal in mind. These initial perceptions are intended to model what a person familiar with the graph domain might notice when first looking at a graph, such as intersecting horizontal and vertical lines. These will serve as cues to retrieve the appropriate schemas from LTM, in this case the schema for a graph of a function in Cartesian coordinates. This schema sets up perceptions to identify the axes, the scale markers, the origin, the functional curve, and the variables. The schema also has procedures to answer classes of questions, such as the Y value for a given X, and performing trend estimates, and using them for inferring Y values for ranges of X not covered by the existing graph. The extension of the graph is now imagined and added to the DRS. The general procedure is instantiated to call for the perception of the point on the extension that corresponds to x =4, which in turn called for a vertical line from x = 4 to be imagined, and an anchor point mentally created, and then a projection to be drawn to the Y-axis and another anchor to be created on Y-axis, and finally the value to be determined. The representational capabilities of biSoar and the associated perceptions and imagery operations were adequate for all steps of the process.

The above model, and others we have built, display all the phenomena identified in our review: visual image in WM, visual descriptions built guided by graph schema knowledge in LTM, bottom-up and top-down processing for cognitive integration (Shah, Trafton), goal-driven problem solving, and the use of imagery operations in cognition (Simkin, Gillan, Trafton). While the level of modeling we described can be useful to investigate the role of different pieces of knowledge and certain types of learning, the true usefulness of such models is the potential to predict timing and error rates in the use of graphs, so that proposed graphs designs can be evaluated. For this we need human performance data, and, even better, computational models that reproduce human performance, on a variety of perceptions and imagery operations required for graph use. Empirical research of this sort, and deeper understanding of the underlying perceptual mechanisms are needed.

**Acknowledgments.** This research was supported by the Advanced Decision Architectures Collaborative Technology Alliance sponsored by the U.S. Army Research Laboratory under Cooperative Agreement DAAD19-01-2-0009.

## References

[1] Bertin, J.: Semiology of Graphs. University of Wisconsin Press, Madison (1983)

[2] Kosslyn, S.: Understanding charts and graphs. Applied Cognitive Psychology 3, 185–226 (1989)

[3] Pinker, S.: A theory of graph comprehension. In: Feedle, R. (ed.) Artificial Intelligence and the Future of Testing, pp. 73–126. Erlbaum Hillsdale, New Jersey (1990)

[4] Shah, P., Freedman, E.: Toward a model of knowledge-based graph comprehension. In: Hegarty, M., Meyer, B., Narayanan, N.H. (eds.) Diagrams 2002. LNCS (LNAI), vol. 2317, pp. 18–31. Springer, Heidelberg (2002)

[5] Cleveland, W.S., McGill, R.: Graphical perception and graphical methods for analyzing scientific data. Science 229(4716), 828–833 (1985)

[6] Simkin, D., Hastie, R.: An information-processing analysis of graph perception. Journal of the American Statistical Association 82(398), 454–465 (1987)

[7] Gillan, D.J.: A Componential model of human interaction with graphs: VII. A Review of the Mixed Arithmetic-Perceptual Model. In: Proceedings of the Human Factors and Ergonomics Society 52th Annual Meeting, pp. 829–833 (2009)

[8] Trafton, J.G., Ratwani, R.M., Boehm-Davis, D.A.: Thinking graphically: extracting local and global information. Journal for Experimental Psychology 14(1), 36–49 (2008)

[9] Lohse, J.: A cognitive model for the perception and understanding of graphs. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Reaching Through Technology, New Orleans, Louisiana, pp. 137–144 (1991)

[10] Peebles, D., Cheng, P.C.: Modeling the effect of task and graphical representation on response latency in a graph reading task. Human Factors 45(1), 28–46 (2003)

[11] Foster, M.E.: Evaluating models of visual comprehension. In: Proceedings of Eurocogsci 2003: The European Cognitive Science Conference 2003. Erlbaum, Mahwah (2003)

[12] Chandrasekaran, B., Kurup, U., Banerjee, B., Josephson, J.R., Winkler, R.: An architecture for problem solving with diagrams. In: Blackwell, A.F., Marriott, K., Shimojima, A. (eds.) Diagrams 2004. LNCS (LNAI), vol. 2980, pp. 151–165. Springer, Heidelberg (2004)

[13] Kurup, U.: Design and Use of a Bimodal Cognitive Architecture for Diagrammatic Reasoning and Cognitive Modeling, Ph D. Thesis, Columbus. The Ohio State University, Ohio (2007)

[14] Matessa, M., Archer, R., Mui, R.: Dynamic Spatial Reasoning Capability in a Graphical Interface Evaluation Tool. In: Proc. 8th International Conference on Cognitive Modeling, Ann Arbor, MI, USA, pp. 55–59 (2007)

# Getting a Clue: Gist Extraction from Scenes and Causal Systems

Alexander Eitel, Katharina Scheiter, and Anne Schüler

Knowledge Media Research Center,
Konrad-Adenauer-Str. 40, 72072 Tübingen, Germany
{a.eitel,k.scheiter,a.schueler}@iwm-kmrc.de

**Abstract.** Numerous studies in basic cognitive research have shown that the gist in realistic scenes is extracted after very short presentation times. So far, the investigation of gist extraction has been limited to pictures of scenes. The present study investigates whether the gist in diagrams of causal systems, which are typically used as instructional material, is extracted as fast as the gist in pictures of scenes, and whether more than just the gist is extracted from a causal system (i.e., information concerning its details and functioning) at slightly longer presentation times. Schematic and realistic pictures of scenes and causal systems were presented to subjects (*N* = 24) at different presentation times. Results showed that the gist in causal systems is extracted as fast as in scenes, but not much more than the gist is extracted after slightly longer presenting a diagram of a causal system.

**Keywords:** Gist, Causal Systems, Short Presentation Times.

## 1 Introduction

### 1.1 Processing of Text and Diagrams

When confronted with information from text and picture, subjects often initially look at the picture for a short time before they start to read the text. This pattern of processing has been shown in advertisements [1], comics [2], real-world scenes [3], and biology schoolbooks [4]. In a study from Stone and Glock [5], in which subjects had to learn how to build a cardboard loading cart, subjects first looked at the diagram for 1000 to 2000 ms, before they started to read the text. According to the authors, subjects initially looked at the diagram in order to get a first impression of what the material was about. Whereas this phenomenon has not been directly addressed in research on learning from text and diagrams, there has been ample research in basic cognitive psychology about the extraction of information from briefly looking at pictures. In this research, it is assumed that briefly looking at pictures gives us a holistic impression of the overall meaning of the picture, that is, its gist.

### 1.2 Extraction of Gist from Scenes

In an early study of Biederman and colleagues [6], subjects had to select one out of two labels, which they judged to better describe a picture of a scene. When the two

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 243–249, 2010.

labels were similar (e.g. "shopping plaza" vs. "busy road and stores"), accuracy of selecting the right label was at 100% for 22 out 32 subjects after 300 ms of presentation. When the two labels were dissimilar, a ceiling effect in accuracy of selecting the right label occurred already after 100 ms. The authors concluded that information about the gist of a scene is already extracted after a single fixation, which enabled subjects to perform the task correctly. Similarly, Loftus, Nelson and Kallman [7] conducted a study in which subjects had the task of deciding whether a picture had already been presented or not. Subjects were told to base their decision either on general properties of the picture (camera angle, mirrored) or on detail information. When the decision was based on general properties of the picture, performance increased much less between 250, 500, and 1000 ms presentation time than when the decision was based on detail information. The authors concluded that most holistic information in scenes is extracted from the first fixation (~230ms; [8]) and subsequent fixations have the primary purpose of seeking out relevant details. This rapid extraction of holistic information in pictures of realistic scenes has later been referred to as scene-gist [9] [10]. The concept of gist is however not clearly defined and results from respective studies largely depend on the resolution level of the ontological categories subjects have to choose from (e.g., shepherd dog or poodle vs. dog or tree). Despite this vagueness, studies in basic cognitive psychology consistently demonstrate that holistic information on a realistic scene's overall meaning can be extracted within the first fixation.

### 1.3 Research Questions

Looking at a picture for a short time is sufficient to extract the gist [7]. However, this rapid gist extraction has yet only been demonstrated with pictures of realistic scenes, whereas subjects also first looked at the picture for a short time when confronted with stimulus material other than realistic scenes [2]. When confronted with schematic learning material [5], subjects first looked at the diagram for 1000 to 2000 ms, which was interpreted as the time to extract the gist and is thus much longer than one fixation (~230ms), within which the gist was extracted in scenes. So, either the gist takes much longer to be extracted in schematic learning material than in realistic scenes or more than just the gist is extracted at longer presentation times. In realistic scenes, longer presentation times (1000ms) led to better extraction of details [7]. It is yet unclear whether this is also true for causal systems, which have often been used in studies on learning from text and diagrams. Since in causal systems, at least two components and their relationship have to be correctly identified in order to understand the functioning of the system, it is highly unlikely that this is extracted along with the gist. Thus, in causal systems, longer presentation times may lead to a better understanding of the functioning of the system.

The present study deals with the questions whether the gist in causal systems is extracted as fast as the gist in scenes, whether details are rapidly extracted in scenes and in causal systems, and whether the functioning of causal systems is understood at longer presentation times. Hence, in the present study it is investigated whether findings from basic cognitive research can be applied to instructional material. To overcome the confounding that in previous research, mostly realistic scenes and schematic causal systems were used, in the current study we experimentally varied the degree of realism in the pictures.

# 2 Method

## 2.1 Participants and Design

Twenty-four students (15 female, 9 male, average age: $M$ = 23.83 years, $SD$ = 3.50) from the University of Tuebingen, Germany, took part in the experiment for either payment or course credit. The experiment followed a 2 × 2 × 4 design, with *Type* (scene vs. causal system), *Realism* (schematic vs. realistic) and *Presentation Time* (50 vs. 250 vs. 1000 vs. 3000 ms) serving as within-subjects factors.

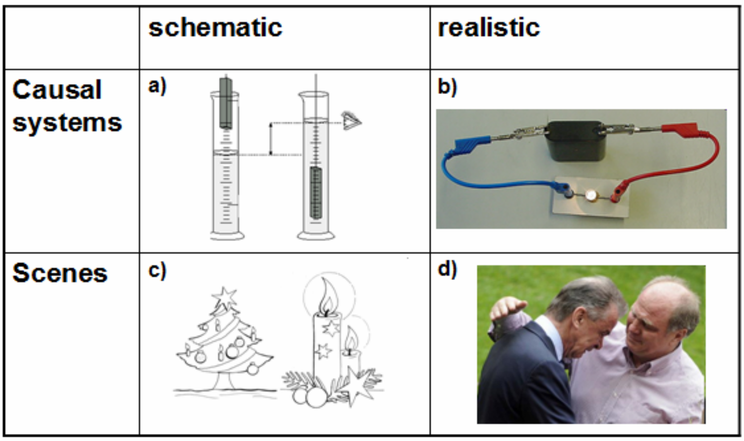



**Fig. 1.** Categorization of pictures used in the experiment. Pictures could either depict a scene or a causal system and could be either schematic or realistic.

## 2.2 Materials and Procedure

The materials comprised 80 pictures of scenes and 80 pictures of causal systems (see Fig. 1), which had previously been pre-tested in a pilot study. A scene depicted an everyday situation. A causal system always had a certain purpose (i.e. pulling weight). It consisted of multiple components, where at least one component was influenced by another component – hence, removing one component would have changed the functioning of the system. For both, scenes and causal systems, half of the pictures were schematic, the other half realistic. This led to four different categories of pictures in the experiment (see Fig. 1). Each picture appeared in the center of the computer screen and covered nearly the whole screen size. An experimental session consisted of 8 training trials and 160 experimental trials. Each experimental trial started with the presentation of the word "ready?", which remained on the screen until a key was pressed. After pressing a key, the word "ready?" was replaced by the fixation cross, which was displayed for 800 ms. Then a picture (scene vs. causal system, schematic vs. realistic) appeared for either 50, 250, 1000 or 3000 ms, respectively, and was immediately masked afterwards. Both pictures and presentation times were presented in a randomized order. After each picture, a statement about the gist, then about details, and then about the functioning of the picture was presented and students were asked to respond to these statements (see Measures section for details). The statement concerning the functioning was presented only after pictures of causal systems. After responding to the last statement (detail or functioning), the trial was over and the word "ready?" reappeared, which marked the beginning of a new trial. An experimental trial for a single picture lasted about 15 seconds. The whole experimental session lasted approximately 45 minutes.

### 2.3 Measures

After viewing each picture, participants had to respond to either two or three statements about the picture depending on the experimental condition. All statements were in a two-alternative forced-choice format, where students had to choose between a "yes" and a "no" response by pressing one of two keys on a keyboard. In half of the trials, "yes" was the correct response, in the other half of the trials "no" was correct. The first statement was about the gist of the picture. For instance, students were asked to decide whether a scene could be identified as "happy people" (see Fig. 1d) or whether a causal system could be identified as "electric circuit" (see Fig. 1b). Statements about the gist always consisted of only one to three words. In the second statement, participants had to judge whether specific details had been present in the scene (e.g. "presents are lying under the tree"; see Fig. 1c) or in the causal system (e.g. "an eye is depicted"; see Fig. 1a) just seen. Four independent raters of the materials made sure that the details were not relevant to either the meaning of the scene or the functioning of the causal system. Moreover, details were depicted in the periphery of the picture so that they were unlikely to be seen within the first fixation. The third statement was presented only after pictures of causal systems, and was about the functioning of the depicted system (e.g. "If the block is pulled out of the test tube, then liquids are at the same level in both test tubes"; see Fig. 1a). The detail and functioning statements consisted of one sentence each.

As the main dependent variable, the mean accuracy was computed. Each correct response (both hits and correct rejections) was coded with 1, each incorrect response with 0. Thus, mean accuracy was 1 at maximum and 0.5 at chance-level and was computed separately for the three types of statements (gist, details, and functioning). Mean reaction times (RT) for responses to the different statements served as a second dependent variable in the experiment. Eye Tracking data were assessed as well, but will not be reported here.

## 3 Results

Overall, results revealed that there was no speed-accuracy trade-off, since accuracy and RT were not significantly correlated ($r$ = .10; $p$ = .65). Thus, only accuracy to statements about the gist, details, and the functioning will be analyzed here.

### 3.1 Gist

T-tests revealed that both in scenes ($M$ = .82, $SD$ = .12; $t(23)$ = 16.57, $p$ < .001) and in causal systems ($M$ = .72, $SD$ = .17; $t(23)$ = 7.98, $p$ < .001), accuracy to gist statements was above chance-level at the shortest presentation time (50 ms), which speaks in favor of an early extraction of gist from both, scenes and causal systems.

A 2 (*Type*: scenes vs. causal systems) × 2 (*Realism*: realistic vs. schematic) × 4 (*Presentation Time*: 50 vs. 250 vs. 1000 vs. 3000 ms) repeated-measures ANOVA was conducted to analyze accuracy for statements about gist. There was a significant main effect of *Type*, meaning that statements about gist were answered more accurately ($F(1, 23) = 65.58$, $p < .001$) in scenes ($M = .92$, $SD = .08$) than in causal systems ($M = .72$, $SD = .17$), which is probably due to a higher difficulty in reporting the general topic of causal systems. There was also a significant main effect of *Presentation Time*, meaning that gist extraction improved with longer presentation times ($F(2.00, 45.98) = 36.00$, $p < .001$). Contrasts of two post-hoc ANOVAs, conducted separately with scenes and causal systems, revealed that much more information about the gist in scenes ($F(1, 23) = 25.36$, $p < .001$) was extracted at 250 ms ($M = .93$, $SD = .08$) than at 50 ms ($M = .82$, $SD = .12$), whereas only slightly more gist information in scenes ($F(1, 23) = 5.87$, $p = .02$) was extracted at 1000 ms ($M = .97$, $SD = .06$) than at 250 ms. Between 1000 ms and 3000 ms ($M = .97$, $SD = .06$), there was no further improvement in scenes ($F < 1$). In causal systems, accuracy further improved ($F(1, 23) = 12.53$, $p = .002$) from 1000 ms ($M = .82$, $SD = .15$) to 3000 ms ($M = .90$, $SD = .13$). To conclude, in scenes, gist extraction reached an asymptotical level at presentation times between 250 ms and 1000 ms. In causal systems, however, no asymptotical level of gist extraction was reached (see Fig. 2). The degree of realism did neither affect gist extraction nor did it moderate the effects of the other independent variables (all $p > .05$).

### 3.2 Details

T-tests revealed that both in scenes ($M = .65$, $SD = .14$; $t(23) = 17.37$, $p < .001$) and in causal systems ($M = .58$, $SD = .16$; $t(23) = 6.49$, $p < .001$), overall accuracy to detail statements was above chance-level. A 2 (*Type*) × 2 (*Realism*) × 4 (*Presentation Time*) repeated-measures ANOVA revealed significant main effects of *Type* ($F(1, 23) = 45.72$, $p < .001$), *Realism* ($F(1, 23) = 4.37$, $p = .048$) and *Presentation Time* ($F(1, 23) = 14.62$, $p < .001$) on accuracy to detail statements. Moreover, the interaction *Type*Realism* was significant ($F(1, 23) = 7.48$, $p = .01$), showing that details were reported more accurately in schematic ($M = .61$, $SD = .14$) compared to realistic

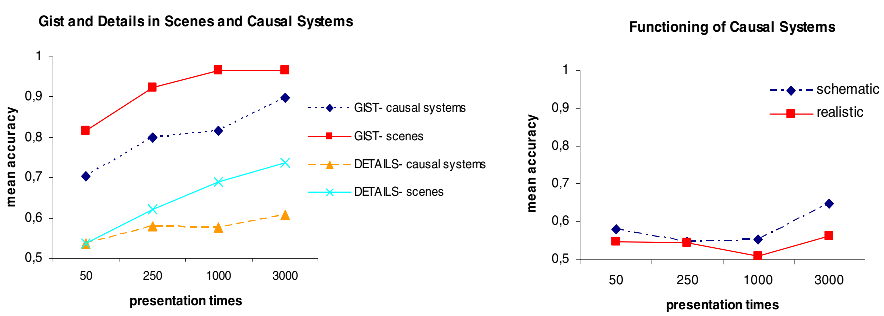



**Fig. 2.** Accuracy to statements about the gist and details in scenes and causal systems (left graph), and about the functioning of causal systems (right graph)

(*M* = .54, *SD* = .17) causal systems, which is possibly due to a higher complexity of realistic than schematic causal systems. On the other hand, there was no effect of *Realism* for scenes. The interaction *Type*Presentation Time* also proved significant ($F(3, 69) = 3.63$, $p = .02$), demonstrating that details in scenes were better extracted at longer presentation times, but not in causal systems (see Fig. 2). This suggests that in causal systems, subjects may use the longer viewing times to try to better understand the system rather than attending to potentially irrelevant details.

### 3.3 Functioning

T-tests revealed that the overall accuracy to statements about the functioning was above chance-level (*M* = .56, *SD* = .16; $t(23) = 5.99$, $p < .001$), which means that the functioning of some causal systems can already be understood at these short presentation times. A 2 (*Realism*) × 4 (*Presentation Time*) repeated-measures ANOVA revealed a significant main effect of *Realism*, meaning that the functioning of schematic causal systems (*M* = .58, *SD* = .16) was understood better ($F(1, 23) = 4.51$, $p = .045$) than that of realistic ones (*M* = .54, *SD* = .16). This can be explained in terms of subjects' lower familiarity with realistic than with schematic depictions of causal systems, since the latter are often used as learning material in technical and scientific domains. There was no effect of *Presentation Time* nor was there an interaction.

## 4 Discussion

The results demonstrate that the gist is extracted very early (< 50 ms) both in scenes and in causal systems, and at least in scenes, reaches an asymptotical level between 250 and 1000 ms. In line with prior research [7], details are better reported with longer presentation times in scenes. Comprehension of the functioning of causal systems did not improve with longer presentation times. Whereas the descriptive values showed a tendency towards better understanding of schematic causal systems at longer presentation times, no such tendency was evident for realistic causal systems. Thus, it can be assumed that with even longer presentation times, comprehension of the functioning of schematic causal systems will improve.

In general, results demonstrated that a well established effect in basic cognitive research (rapid gist extraction in scenes) can be found with instructional material (causal systems) as well. Hence, the function of shortly processing a diagram before reading the respective text presumably is gist extraction [4]. However, it is still unclear whether extracting the gist from a diagram prior to reading the respective text fosters learning. The gist of a diagram may provide a scaffold on which information can be added, as Friedman [11] pointed out. Castelhano and Henderson [9] supplemented that gist extraction leads to priming of the spatial structure of a picture, and this spatial structure "lingers in memory and can facilitate later perceptual and cognitive operations and behavior" (p. 760). Hence, if the gist already provides a structure of a diagram that can be held in memory for some time, than the gist of a diagram possibly fosters learning from subsequent text as well. In a further study, it will be investigated whether these assumptions hold true. As such, the two studies combined are conducted to analyze the processes that take place during learning from text and diagrams by taking findings from basic cognitive psychology into account.

## References


1. Rayner, K., Rotello, C.M., Steward, A.J., Keir, J., Duffy, S.A.: Integrating Text and Pictorial Information: Eye Movements When Looking at Print Advertisements. J. Exp. Psychol. Appl. 7, 219–226 (2001)
2. Carroll, P.J., Young, J.R., Guertin, M.S.: Visual Analysis of Cartoons: A View from the Far Side. In: Rayner, K. (ed.) Eye Movements and Visual Cognition: Scene Perception and Reading, pp. 444–461. Springer, New York (1991)
3. Underwood, G., Jebbett, L., Roberts, K.: Inspecting pictures for information to verify a sentence: eye movements in general encoding and in focused search. Q. J. Exp. Psychol-A 57, 165–182 (2004)
4. Mak, P.: Effects of references from text to picture on the processing of school texts: Evidence from eye tracking. In: Maes, A., Ainsworth, S. (eds.) Proceedings EARLI Special Interest Group Text and Graphics: Exploiting the Opportunities - Learning with Textual, Graphical, and Multimodal Representations, pp. 93–96. Tilburg University, Tilburg (2008)
5. Stone, D.E., Glock, M.E.: How do Young Adults Read Directions with and without Pictures? J. Educ. Psychol. 73, 419–426 (1981)
6. Biederman, I., Rabinowitz, J.C., Glass, A.L., Stacy, E.W.: On the Information Extracted from a Glance at a Scene. J. Exp. Psychol. Gen. 103, 597–600 (1974)
7. Loftus, G.R., Nelson, W.W., Kallman, H.J.: Differential Acquisition Rates for Different Types of Information From Pictures. Q. J. Exp. Psychol.-A 35, 187–198 (1983)
8. Henderson, J.M., Hollingworth, A.: Eye movements during scene viewing: An overview. In: Underwood, G. (ed.) Eye Guidance in Reading and Scene Perception, pp. 269–293. Elsevier, Oxford (1998)
9. Castelhano, M.S., Henderson, J.M.: Initial Scene Representations Facilitate Eye Movement Guidance in Visual Search. J. Exp. Psychol. Human 33, 753–763 (2007)
10. Castelhano, M.S., Henderson, J.M.: The Influence of Color on the Perception of Scene Gist. J. Exp. Psychol. Human 34, 660–675 (2008)
11. Friedman, A.: Framing Pictures: The Role of Knowledge in Automatized Encoding and Memory for Gist. J. Exp. Psychol. Gen. 108, 316–355 (1979)

# Attention Direction in Static and Animated Diagrams

Richard Lowe[1] and Jean-Michel Boucheix[2]

[1] Curtin University, Australia
r.k.lowe@curtin.edu.au
[2] University of Burgundy, France, LEAD-CNRS
Jean-Michel.Boucheix@u-bourgogne.fr

**Abstract.** Two key requirements for comprehending a diagram are to parse it into appropriate components and to establish relevant relationships between those components. These requirements can be particularly demanding when the diagram is complex and the viewers are novices in the depicted domain. Lack of domain-specific knowledge for top-down guidance of visual attention prejudices novices' extraction of task-relevant information. Static diagrams designed for novices often include visual cues intended to improve such information extraction. However, because current approaches to cueing tend to be largely intuitive, their effectiveness can be questionable. Further, animated diagrams with their perceptually compelling dynamic properties pose new challenges for providing appropriate guidance of attention. Using a hydraulic circuit diagram example, this paper considers human information processing influences on the direction of visual attention in complex static and dynamic diagrams. It aims to stimulate a more principled approach to cue design.

**Keywords:** Visual attention, static and animated diagrams, cueing, visual processing, complex content.

## 1 Introduction

In educational and explanatory contexts, visual cues are often added to complex static and animated diagrams with the goal of helping viewers to take account of high-relevance information they may otherwise neglect. For example, an important object may be highlighted using a contrasting colour to make it stand out from the rest of the diagram. The assumption underlying such graphic elaboration appears to be that cues will direct attention to these key aspects, so increasing their likelihood of being noticed, extracted, and incorporated into the viewer's developing mental model of the referent subject matter. Current common practice in the visual cueing of diagrams suggests an approach largely based on designer intuitions rather than a deep understanding of cue functioning. It involves choosing cues from a limited repertoire of pre-existing cueing techniques with little or no detailed associated consideration of how people may process static and animated diagrams. In a research context, inconsistent findings on the effectiveness of cueing reported in recent studies [1] [2] suggest that empirical work in this area may benefit from more substantial theoretical underpinnings. This paper examines cueing of diagrams from a processing

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 250–256, 2010.

perspective with particular emphasis on the direction of visual attention during parsing and the establishment of relationships between diagram components. Although a hydraulic circuit diagram example is used to illustrate our discussion, the points we make are general ones likely to apply beyond this specific case.

## 2 Hydraulic Circuit Diagrams

Hydraulic circuit diagrams are abstract depictions of the components and pathways that comprise hydraulic systems. They typically show how the pressure applied to a hydraulic fluid is distributed to actuators that produce the actions required to perform a given task. Figure 1 depicts a system for clamping then drilling a work-piece.

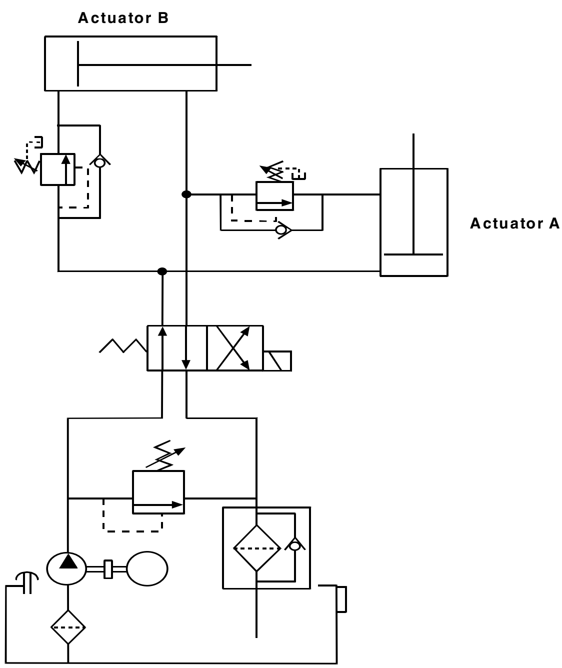



**Fig. 1.** Hydraulic system for clamping and drilling a work-piece

Its overall functioning involves four sequential operations: (i) actuator A applies a clamping force to hold the work-piece in place; (ii) actuator B extends the drill; (iii) the drill is retracted from the work-piece; (iv) the clamp is withdrawn. Note that the latter two movements reverse the former. To sequence this series of operations correctly, the flow of pressurized fluid within the circuit is controlled by several valves. Comprehending the system's functioning involves far more than interpreting the array of abstract symbols used to depict the hydraulic system's components and structure. Viewers must also construct a mental model from the diagram to internally represent how operation of these valves affects fluid pressures, flow directions, and actuator movements. The absence of direct information about these rich dynamic aspects in a static hydraulic diagram makes its interpretation particularly difficult for domain novices. They may fail to extract key task-relevant information that is essential for building a high quality mental model of the depicted system. An increasingly popular way to address difficulties of this type is to convert a static diagram into an animation so that the implicit dynamics are made explicit. The assumed benefit is that viewers can then simply 'read off' this dynamic information rather than having to perform demanding and error-prone mental animations. However, this benefit is not necessarily achieved in practice because animated depictions can actually pose their own distinctive challenges for viewers [3] [4]. Visual cues have the potential to help viewers make sense of both static and animated diagrams. As well as signaling which aspects are most important, suitably designed cues can also indicate how those aspects are related to each other across space and time. Although extraction of this relational information from animated depictions has proven especially difficult for novices, new forms of dynamic cueing appear promising [5].

## 3 Parsing

The interpretation of a hydraulic circuit diagram relies on the viewer being able to parse a complex graphic array into individual pieces that are meaningful at the required level of analysis. For example, Figure 1 can be broken down into two broad sections. The lower part deals with generic aspects common to many hydraulic systems (such as pumping hydraulic fluid, filtering that fluid, and preventing damage due to excessive pressure). In contrast, the upper part deals with aspects specific to the job for which this particular system was designed - clamping and drilling a work-piece.

Knowledgeable viewers could subdivide the diagram into these two parts at a very early stage of their processing and then subsequently devote most of their attention to the upper section where the high relevance information is located. However, domain novices' lack of such knowledge would likely prevent them from making this relevance-based distinction and thereby directing their attention appropriately. Novices tend to be heavily reliant on the raw perceptual features of a graphic display to direct their attention but in this diagram it is not possible to distinguish more relevant and less relevant aspects on the basis of perception alone.

In addition to parsing at a global level of analysis, the high-relevance upper part of the diagram needs to be broken down locally into its key functional aspects. Subsequent finer-grain analysis can identify various operational components such as valves and the actuators they control. These components are depicted via particular configurations of shapes and lines that together indicate specific functions. Parsing the diagram's total set of information into meaningful groups involves both establishing the boundaries of each of the graphic entities and relating these individual entities to others that belong to the same group. Gestalt characteristics such as proximity and similarity may provide novice viewers with some help in establishing these groupings on a perceptual basis. However, without prior knowledge of the configurations and their purposes, it is difficult to determine the boundaries that define functionally-related groups of entities. For example, in Figure 1, the assortment of graphic entities directly below Actuator B depict three functionally distinct components – a drain line, a pilot-controlled sequence valve, and a check valve. A novice has no way of telling from perceptual information alone that the box, the zig-zag line, the two arrows, and the larger of the dotted lines together form a group depicting the pilot-controlled sequence valve.

As noted above, animating a static diagram such as that shown in Figure 1 does not necessarily facilitate its processing. Animated diagrams may be more demanding than their static counterparts because of their transitory depiction of information. If viewer attention is not directed to the right place at the right time, vital information can be missed [6]. The likelihood of missing task-relevant information may be exacerbated in un-cued animations because of the privileged status that dynamic change has in visual perception [7]. For example, our attention may be captured by dramatic movement in a display, irrespective of that dynamic aspect's relevance to the interpretative task at hand. Animated diagrams also require a further form of parsing that goes beyond the visuospatial parsing described above. This additional form of parsing involves analyzing the structure of the material in terms of its dynamics. As well as determining relevant visual and spatial constituents of the display, the viewer needs to carry out a temporal form of parsing that requires segmentation of the animation into key event units [8]. The fleeting nature of animations, the compelling character of their dynamics, and their requirement for task-relevant temporal segmentation suggest that the requirements for designing truly effective cueing techniques for animated diagrams are anything but simple. Rather, it is likely to demand a detailed consideration of the processing factors that may influence the power of different types of cues to steer viewer attention in various contexts. In the remainder of this paper, we first examine the relationship between perception and cueing in static depictions. We then extend the discussion to some special considerations that may apply when designing cues for animated diagrams.

## 4 Perception and Cueing

For the purposes of this paper, cueing is conceptualized as perceptually-based preferential direction of viewer attention to particular aspects of a graphic display. We will restrict our consideration to two types of cueing whose effects rely on differences in perceptual salience. The first, which we term 'intrinsic cueing', is when selective direction of attention takes place incidentally within an unaltered display due to its inherent attributes. For example, in Figure 1 attention may be selectively directed to the 2-position 4-way valve because of its central position in the display and its distinctive graphic properties. The second, which we term 'extrinsic cueing' is a result of deliberate additions to, or manipulations of, the original display. This type of cueing could be introduced into Figure 1 by techniques such as adding an arrow to point out a target aspect of the diagram, or colouring that aspect differently from the rest of the display. The effect of the cueing in these examples should be to make it more likely that viewers will single out particular graphic entities or groups. The potential value of extrinsic cueing depends on its capacity to act as a corrective to misleading intrinsic cues that can distract viewer attention from high relevance aspects of a display.

Perceptually-based cueing relies on contrast between the target aspect and its context to signal the location of high relevance information and to delineate its boundaries. In static diagrams, this contrast concerns visuospatial properties of the depiction.

Two possibilities for increasing the visuospatial contrast are to (i) raise the perceptual salience of the target (cueing), or (ii) lower the perceptual salience of its context (anti-cueing) [9]. However, in animated diagrams, a further resource is available for cueing. Differences in dynamic properties of the display's components may result in the viewer's attention being preferentially attracted to some aspects while others tend to be neglected. Again, this dynamically-based cueing may be either intrinsic or extrinsic. It would clearly be advantageous if the likely perceptual salience of display aspects could be determined *a priori*. The computer algorithms developed by Itti and colleagues [10] show promise in this regard, especially with their recent extension to animated displays. Because both visuospatially-based and dynamically-based cueing can be present in the same animated diagram, it is possible for these two forms of cueing to compete for the viewer's attention. However, it is also possible that animation may sometimes help with visuospatial parsing when coordinated movements of graphic entities signals that they are functionally related. These possibilities have potentially important implications for the effectiveness of cues added to animated diagrams.

## 5 Cue Processing and Design

Cued diagrams need to be designed so that the combined effects of intrinsic and extrinsic cues result in appropriate searching of the available information. To be effective, extrinsic cues employed to signal high relevance aspects of the display must 'out-compete' intrinsic cues offered by low-relevance aspects. However, it is not sufficient merely to provide cues with sufficient power to direct viewer attention to high-relevance aspects. Consideration must also be given to how those cues fit into the viewer's overall processing regime. The processing of static and animated diagrams is constrained by the limits of our human information processing system. It is a cumulative procedure whereby perceptually and cognitively 'manageable' pieces of information are extracted from the external representation and progressively internalized. This internalization is supplemented by background knowledge to build up a mental model of the depicted referent. For those who are familiar with the referent subject matter, background knowledge allows them to fill in information that is missing from the diagram (such as the dynamics of the various sub-systems in a hydraulic circuit). However, it also helps them to give a weighting to the amount of attention each aspect deserves, to sequence their processing of the visual array in an appropriate manner, and to search out typical relationships that the depicted type of system would be expected to involve. When presented with an un-cued diagram from their domain of expertise, such viewers are equipped to marshal their information processing resources in ways that make good used of the limited capacity available.

In contrast, novices are far more reliant on information explicitly provided by the external representation, not only for informational gap-filling, but also for leading them through a content/task-appropriate processing sequence and for processing task-relevant relationships. When presented with a complex depiction such as a hydraulic circuit diagram, they need support that will help them to extract tractable chunks of information suitable for efficient and effective building of a high quality mental model. Visual cueing has the potential to provide this support but only if well designed in terms of the processing affordances it offers the viewer. For example, although perceptual contrast is a fundamental aspect of cue processing, there appears to be an asymmetrical relationship between the dynamic character of a cue and that of its context. In particular, while dynamic cues can be highly effective in static diagrams [11], the reverse does not seem to hold (even when static cues are presented prior to animation of the diagram [9]). This casts doubt on the practice of merely 'borrowing' cueing techniques that have been found effective with static depictions and inserting them into animated diagrams. A possible processing explanation for the failure of these conventional cues within an animated context is that they simply cannot compete with intrinsic cueing produced by the animation's dynamic components. The perceptually compelling effect of temporal change wins the contest for the viewer's attention.

Although there is an extensive legacy of practical experience available to guide the design of cues for static graphics, there is no corresponding legacy about effective cueing of animations [c.f. 12]. In the absence of tried and true approaches, we suggest that the design of cues for animated diagrams be approached from the perspective of the visual processing that viewers need to engage in with these depictions. Current intuitive and simplistic approaches to cue design and usage should be replaced by systematic, detailed consideration of both the content being presented by the diagram and the task that the viewer is to perform with that material. For example, with complex, abstract, and unfamiliar content of the type presented in an animated hydraulic circuit diagram, it would be all too easy for cues added on the basis of designer intuition to have the unintended effect of increasing the level of complexity facing the viewer. Rather than arbitrarily grafting existing cue types onto such a diagram, cues should be tailored for the specific processing demands the diagram poses. For this to occur, designers would need to be far better informed beforehand than they are at present about the likely nature of those demands. A processing-oriented approach to designing cues for animated diagrams should also tackle issues such as how to help domain novices break down the continuous temporal flux of an animated diagram into thematically relevant event units. This form of temporal parsing is likely to be extremely difficult with diagrams of the type described here because the real-world knowledge used as a basis for detecting boundaries between everyday events [13] is not applicable in such a specialized domain. Forms of cueing need to be developed that can draw attention to these boundaries because novices lack the domain-specific knowledge required to identify the predictive discontinuities that signal a transition from one event to another.

Designing approaches that are likely to be effective for cueing animated diagrams will require far more analysis of the content and task than is presently the case. In particular, these approaches must go beyond a consideration of the visuospatial attributes of the graphic display alone to foreground the dynamic character of animated diagrams. If traditional cueing is based on techniques such as manipulation of visuospatial attributes, it may be that cues for dynamic graphics could be based on manipulation of temporal attributes. For example, perhaps existing dynamics could be exaggerated to raise the perceptual salience of high-relevance aspects, or low-relevance aspects could be suppressed by temporarily freezing them in place. Future research needs to examine the effectiveness of a range of possibilities for manipulating temporal attributes of animated diagrams in directing learner attention more productively.

## References


1. Boucheix, J.-M.: Young Learners' Interaction with Technical Animations: User Control and Attentional Cues. In: Lowe, R.K., Schnotz, W. (eds.) Learning with Animation: Research and Design Implications, pp. 208–234. Cambridge University Press, New York (2008)
2. de Koning, B.B., Tabbers, H.K., Rikers, R.M.J.P., Paas, F.: Attention Guidance in Learning from a Complex Animation: Seeing is Understanding? Learning and Instruction 20, 112–122 (2010)
3. Kriz, S., Hegarty, M.: Top-Down and Bottom-Up Influences on Learning from Animations. International Journal of Human Computer Studies 65, 911–930 (2007)
4. Schnotz, W., Lowe, R.K.: A Unified View of Learning from Animated and Static Graphics. In: Lowe, R.K., Schnotz, W. (eds.) Learning with Animation: Research Implications for Design, pp. 304–356. Cambridge University Press, New York (2008)
5. Boucheix, J.-M., Lowe, R.K.: An Eye Tracking Comparison of External Pointing Cues and Internal Continuous Cues in Learning with Complex Animations. Learning and Instruction 20, 123–135 (2010)
6. Lowe, R.K.: Learning from Animation: Where to Look, When to Look. In: Lowe, R.K., Schnotz, W. (eds.) Learning with Animation: Research Implications for Design, pp. 304–356. Cambridge University Press, New York (2008)
7. Wolfe, J.M., Horowitz, T.S.: What Attributes Guide the Deployment of Visual Attention and How Do They Do It? Nature Reviews Neuroscience 5, 1–7 (2004)
8. Lowe, R.K., Boucheix, J.-M.: Learning from Animated Diagrams: How are Mental Models Built? In: Stapleton, G., Howse, J., Lee, J. (eds.) Diagrams 2008. LNCS (LNAI), vol. 5223, pp. 266–281. Springer, Heidelberg (2008)
9. Lowe, R.K., Boucheix, J.-M.: Complex Animations: Cues Foster Better Knowledge Structures? In: Research Paper, 13th EARLI Conference, Amsterdam, Netherlands (2009)
10. Peters, R.J., Itti, L.: Applying Computational Tools to Predict Gaze Direction in Interactive Visual Environments. ACM Transactions of Applied Perception, Article 9, 5 (2008)
11. Grant, E.R., Spivey, M.J.: Eye Movements and Problem Solving: Guiding Attention Guides Thought. Psychological Science 14, 462–466 (2003)
12. Tversky, B., Heiser, J., Lozano, S., MacKenzie, R., Morrison, J.: Enriching Animations. In: Lowe, R.K., Schnotz, W. (eds.) Learning with Animation: Research Implications for Design, pp. 304–356. Cambridge University Press, New York (2008)
13. Zacks, J.M., Speer, N.K., Swallow, K.M., Braver, T.S., Reynolds, J.R.: Event Perception: A Mind/Brain Perspective. Psychological Bulletin 127, 273–293 (2007)

# Tactile Diagrams: Worth Ten Thousand Words?

Cagatay Goncu[1,⋆], Kim Marriott[1], and Frances Aldrich[2,⋆⋆]

[1] Clayton School of IT, Monash University
[2] Department of Psychology, University of Sussex
cagatay.goncu@infotech.monash.edu.au,
kim.marriott@infotech.monash.edu.au,
f.k.aldrich@sussex.ac.uk

**Abstract.** The properties that make diagrams more effective than text in certain circumstances have been investigated by researchers for over 20 years. However, this research has focused on visual diagrams. To the best of our knowledge, no research has yet investigated whether the same benefits and properties hold for tactile diagrams, which are blind people's primary means of access to diagrams. We present a consideration of similarities and differences in the properties and potential benefits of visual and tactile diagrams; and suggest where experimental investigation would be useful.

## 1 Introduction

Starting with Larkin and Simon [1] many researchers have investigated the differences between diagrams and text and the benefits that sometimes makes a diagram more effective than text [2,3,4]. Identified benefits of diagrams include: geometric and topological congruence, homomorphic representation, computational off-loading, indexing, mental animation, macro/micro view, analogue representation and graphical constraining. These benefits have been explained in terms of visual perception.

Not everyone is able to perceive a diagram visually, however. People who are blind most frequently access diagrams through versions they can *feel*: such tactile diagrams have been in use for over 200 years [5]. Verbal descriptions and explanations have always been an indispensable supplement to tactile diagrams, and now technological advances such as touchpads are allowing increasingly sophisticated use of audio feedback in conjunction with static tactile displays [6,7]. For a brief introduction to tactile diagrams see [8], and for an overview of research and development in the field see http://www.nctd.org.uk/conference

In this paper we present an initial analysis of similarities and differences of the benefits provided by visual and tactile diagrams, together with further issues for experimental investigation. Our observations on blind people using tactile diagrams is the main motivation for this work. We take diagrams to include tables,

⋆ We acknowledge the support of the ARC Discovery Project Grant DP0987168.
⋆⋆ We acknowledge the support of R. M. Phillips Settlement.

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 257–263, 2010.

graphs, plots, charts, structure diagrams, process diagrams, maps, cartograms, and illustrations [9]. We believe we are the first researchers to make this comparison, called for at Diagrams 2008 by [10]. We hope, by doing so, to provide insights into the cognitive leverage that blind people can gain from tactile diagrams, so increasing the effectiveness of their design. While there has been some research into guidelines for tactile diagram design(e.g. [11]) this has not considered research into the benefits of visual diagrams.

## 2 Visual and Haptic Perception

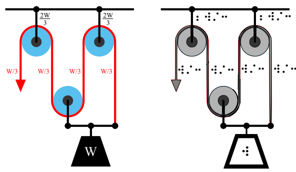
The visual subsystem has sensors that receive light and provide visual information such as shape, size, colour, intensity and position [12]. It needs no physical contact with objects to acquire this information. It has a wide area of perception that provides parallel information in a continuous flow [13], and within this is a narrow area (the fovea) capable of detecting highly detailed information [12].

The haptic subsystem is specialized to receive stimuli about touch, temperature, and motion [12]. It requires physical contact with objects to acquire information. Cutaneous actuators on the skin detect touch and temperature, while the kinesthetic actuators on the muscles and joints of the body sense motion [14]. Because of the sequential movement of hands and fingers involved in perception, acquisition of information is less parallel than vision [13]. While the haptic subsystem can provide much of the same information as the visual subsystem (shape, size, texture, and position) [13], it is not as fast at doing so [15]. However, it is better than the visual system at detecting properties of objects, such as texture and hardness [16]. The haptic perceptual field is much smaller than the visual field and is centred on the hands. Because there is no haptic equivalent of peripheral vision, the position of previously encountered objects must be stored in memory thus there are no perceptual cues to recapture attention [17]. Nonetheless, haptic input can lead to internal spatial representations that are functionally equivalent to those obtained from visual input [18].

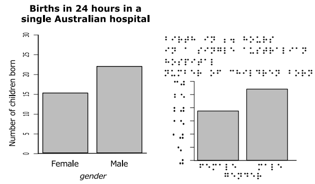
## 3 Likely Benefits of Tactile Diagrams

In this section we consider which of the benefits previously identified for visual diagrams are likely to apply to tactile diagrams, given differences in visual and haptic perception. We will use the example diagrams shown in Figure 1 to illustrate our discussion.

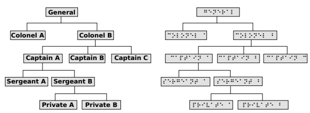
**Topological and geometric resemblance:** It has been claimed that diagrams are effective because they resemble the entity they represent. The simplest example of this is when the diagram preserves the topology and geometric relationships, including relative size or position, of the objects being represented [1] (as in the pulley system and floor plan examples).




(a) Pulley System




(b) Histogram

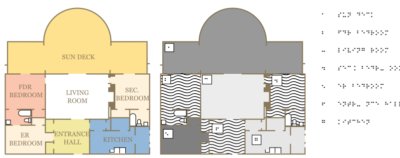



(c) Organization Chart

Truth Table

| $A_1$ | $A_0$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

(d) Table




(e) Floor Plan

**Fig. 1.** Sample diagrams: Each figure shows both visual and a tactile version. Note that the floor plan has also got a key for its labels. (images are taken from http://en.wikipedia.org)

Tactile presentation preserves topology and geometry and so potentially has the same benefit. However, whereas the visual subsystem identifies these relationships almost instantaneously, the haptic subsystem must obtain the information through sequential hand movements. Storing and integrating this information introduces a memory load which presumably detracts from the potential benefit of geometric and topological congruence, to an extent yet to be investigated.

More generally, "resemblance" occurs when there is a *homomorphism* between the diagrammatic representation and the target [19,20], with topology and geometric relationships being used to represent abstract relationships with similar abstract properties (e.g. the use of containment to represent order in a hierarchy). Such homomorphism works well when the representation allows computational off-loading–see below. It is worth pointing out that tactile diagrams may differ markedly from visual diagrams in the forms of "resemblance" they employ [21].

One issue that affects the value of the homomorphism is the level of virtual realism. Studies suggest that people reason differently when the entities in a diagram have different fidelity. When people are asked to determine whether the marks of a hinge will meet when it is closed, the results of accuracy and latency data indicates that they simulate the movement in the high fidelity diagrams, whereas they use abstract features such as length, and angles for the low fidelity one [22]. It is worth investigating which fidelity level tactile diagrams do have.

**Computational off-loading:** One of the major benefits of visual diagrams over text is that diagrams support perceptual inferencing. Perceptual inferencing depends on the visual and spatial properties of the diagram encoding the underlying information in a way that capitalises on automatic processing to convey the intended meaning. For example, an organisation chart allows perceptual inferencing of relative seniority by vertical positioning. Perceptual inferencing was first identified as a benefit of diagrams by Larkin and Simon, and has been explored by other researchers under the names of *computational off-loading* and "free rides" [2,4].

Tactile diagrams would seem to provide some of the same advantages. For instance, the benefits of perceptual inferences based on topology and geometric information will hold to some extent for tactile diagrams, subject to the fact that more cognitive effort is required to discover the geometric and topological relationships. Clearly colour cannot be used in a tactile diagram, but line and texture heights and patterns can be used to indicate relative magnitude.

For sighted people the elements of a diagram help reduce working memory load, allowing them to focus more fully on the meaning of the diagram [23]. As we noted earlier, however, there is no haptic equivalent of peripheral vision, and so blind people will need to expend more effort in keeping all the diagram elements in mind at the same time as working on their meaning. The extent to which there may still be "free rides" would be worth exploring experimentally.

**Indexing:** A benefit identified by Larkin and Simon [1] is that diagrams group information that is relevant and, in particular, information about a single element is often placed together. Such *locational indexing* avoids the need to use and match symbolic labels, as is often required in textual representations referring to the different properties/relationships of an object. This is a feature of all of our example visual diagrams and most of our tactile ones. The usefulness of locational indexing is arguably even greater for tactile diagrams, as finding information with the haptic system is slower than with the visual system. However, a drawback is that Braille text is very space consuming. To avoid Braille labels overwriting elements of the diagram, keys must often be used instead, requiring cross indexing, as seen in the map example. Tactile-audio presentation can avoid this by providing audio labels in situ.

A related benefit of visual and tactile diagrams is that they allow *two dimensional indexing*: users can locate an object by its $(x, y)$ position on the diagram. This is useful with the table and histogram examples. For instance, when finding the value of a point on a visual line graph or histogram, the eyes move left until the axis is found and then read a y value from the axis. Tactile line graph users do the same with their fingers. There has been some work to improve speed and accuracy of indexing by altering tactile diagram format [24] to facilitate this.

Another kind of indexing benefit provided by diagrams is the use of line segments to guide navigation. An example is the use of lines to connect text to the object it labels, as in grid lines in scatter plots and bar charts, and the lines connecting the elements in the example organization chart. The benefit of such *guidelines* is similar for both visual and tactile diagrams, although they can lead to "clutter" which makes tactile diagrams more difficult to understand [24].

**Mental animation:** Diagrams allow *mental animation* [25]: mentally activating components in the representation of a system (e.g. the pulley example), in a serial manner, in order to reason about the system. Experiments have validated the benefits with visual diagrams. Mental animation can be more effective when it is supported by an external representation that helps people to remember the elements easily. One might expect similar benefits to hold for tactile diagrams but experimental investigation is needed to test this supposition.

**Macro/micro view:** One of the most important benefits of diagrams for data visualization is that parallel information acquisition from a wide area, as well as a narrow detailed focus of attention, provides a seemingly simultaneous *micro/macro* view of the diagram. As noted by Tufte [26], a well designed graphic allows the viewer to get an overview, as well as detailed information about the underlying data. A diagram enables a picture of the whole problem to be maintained *simultaneously*, whilst allowing the user to work through the interconnected parts [27]. A macro view also assists the reader in using visual features of the diagram for navigation. Modern computer visualization applications support interactive micro/macro viewing of visual diagrams: using fish-eye techniques, allowing viewers to pan and zoom at will, and providing an overview window in addition to a detailed view [28,29].

Because of the characteristics of the haptic subsystem, obtaining a macro view of a tactile diagram is much harder. Users are recommended to *sweep* the display with their hands, or place their hands side-by-side on the display, to try to get an overview of the location of information, and then use this information to frame more detailed exploration (e.g. [8]). This might be thought of as a mental macro view. Additionally, some micro/macro functionality can be offered in the design of the materials themselves. Tactile diagrams are sometimes designed as a set which consists of an overview diagram together with additional, more detailed, diagrams. Similarly, verbal descriptions of a tactile diagram can be *layered*, for example focusing initially on the physical layout and then switching to an interpretation of its meaning. Because of current limitations on the technology of dynamic tactile displays, the variety of graphic views available to sighted people through interactive computer technology are not accessible to blind users. However, when tactile diagrams are displayed on touch pads, there is some scope for user choice to select which "layer" of audio description to listen to [6,7].

As noted in [4], we do not have an adequate understanding of the internal representation of a visually presented diagram yet, much less the internal representation of a tactile diagram. At this stage it is unclear how effectively tactile diagrams provide a macro/micro view. This is another area for investigation.

**Other properties:** There are a number of other properties that have been identified as distinguishing diagrams from text. We shall not discuss these in great detail since in our view these properties hold equally for both tactile and visual diagrams. The first is the observation that diagrams can be classified as analog(infinite class of states represents infinite class of information), digital (discrete class of states represents discrete class of information), and mixed [30].

The second is that many diagrams are graphically constraining in the sense that it is impossible to specify partial information, like drawing an object without giving it a size and position on a map [31]. A third is that diagrams allow the use of more memorable and understandable symbols to represent the same concepts as phonetic text since the symbols of text are used to represent sound rather than meaning [3].

## 4 Conclusion and Future Work

We have set out those differences in visual and haptic perception which seem to us relevant to the use of visual and tactile diagrams. We have also presented a number of ways in which visual diagrams are recognised as offering advantages over text, focusing on "resemblance", "computational off-loading", "indexing", "mental animation", and "macro/micro views". For each of these advantages of visual diagrams, we have considered the extent to which it might hold for tactile diagrams.

From these considerations it is evident that there is much research to be done before we have a proper understanding of the effectiveness of tactile diagrams, in comparison both to visual diagrams and to text. We have indicated where this work might begin.

It is already clear, however, that one way to increase the effectiveness of tactile diagrams should be by enabling users to interact more with the displays. For example, facilities for users to switch between different views of a tactile diagram at will, and to add their own annotations [4] to reduce working memory load, should pay dividends in cognitive terms. Unfortunately, the commonly available technologies for presenting tactile diagrams do not facilitate this, although some low tech approaches might well be helpful (e.g. pins and blu tac).

## References

1. Larkin, J., Simon, H.: Why a Diagram is (Sometimes) Worth Ten Thousand Words. Cognitive Science 11(1), 65–100 (1987)
2. Shimojima, A.: The graphic-linguistic distinction: Exploring alternatives. Artif. Intell. Rev. 13(4), 313–335 (1999)
3. Tversky, B.: Spatial schemas in depictions. In: Spatial Schemas and Abstract Thought, pp. 79–111. MIT Press, Cambridge (2001)
4. Scaife, M., Rogers, Y.: External cognition: how do graphical representations work? Int. J. Hum.-Comput. Stud. 45(2), 185–213 (1996)
5. Eriksson, Y.: Tactile Pictures: Pictorial Representations for the Blind, pp. 1784–1940 (1998)
6. Gardner, J., Bulatov, V.: Scientific Diagrams Made Easy with IVEO. In: Miesenberger, K., Klaus, J., Zagler, W.L., Karshmer, A.I. (eds.) ICCHP 2006. LNCS, vol. 4061, pp. 1243–1250. Springer, Heidelberg (2006)
7. Landau, S., Gourgey, K.: Development of a talking tactile tablet. Information Technology and Disabilities 7(2), 4 (2001)
8. Sheppard, L., Aldrich, F.: Tactile graphics: a beginner's guide to graphics for visually impaired children. Primary Science Review, 29–30 (2000)
9. Lohse, G.L., Biolsi, K., Walker, N., Rueter, H.H.: A classification of visual representations. Commun. ACM 37(12), 36–49 (1994)

10. Aldrich, F.: Talk to the Hand: An Agenda for Further Research on Tactile Graphics. In: Stapleton, G., Howse, J., Lee, J. (eds.) Diagrams 2008. LNCS (LNAI), vol. 5223, pp. 344–346. Springer, Heidelberg (2008)
11. Challis, B., Edwards, A.: Design principles for tactile interaction. In: Brewster, S., Murray-Smith, R. (eds.) Haptic HCI 2000. LNCS, vol. 2058, pp. 17–24. Springer, Heidelberg (2001)
12. Coren, S., Ward, L., Enns, J.: Sensation and perception, Hoboken (2004)
13. Hatwell, Y.: Images and non-visual spatial representations in the blind, pp.13–35, John Libbey Eurotext (1993), Source, `http://books.google.com`
14. Gibson, J.: The Senses Considered as Perceptual Systems, Greenwood, New York (1966)
15. Klatzky, R., Lederman, S.: The intelligent hand. The Psychology of Learning and Motivation 21, 121–151 (1988)
16. Lederman, S., Thorne, G., Jones, B.: Perception of texture by vision and touch: multidimensionality and intersensory integration. J. Exp. Psychol. Hum. Percept. Perform. 12(2), 169–180 (1986)
17. Ungar, S.: Cognitive mapping without visual experience. Kitchin and Freundschuh, 221–248 (2000)
18. Cattaneo, Z., Vecchi, T., Cornoldi, C., Mammarella, I., Bonino, D., Ricciardi, E., Pietrini, P.: Imagery and spatial processes in blindness and visual impairment. Neuroscience and Biobehavioral Reviews 32(8), 1346–1360 (2008)
19. Barwise, J., Etchemendy, J.: Visual Information and Valid Reasoning, pp. 9–24 (1991)
20. Sloman, A.: Afterthoughts on analogical representations. In: Proc. Theoretical Issues in Natural Language Processing, pp. 164–168. Morgan Kaufmann, San Francisco (1975)
21. Thompson, L., Chronicle, E.: Beyond visual conventions: Rethinking the design of tactile diagrams. British Journal of Visual Impairment 24(2), 76 (2006)
22. Schwartz, D.: Reasoning about the referent of a picture versus reasoning about the picture as the referent: An effect of visual realism. Memory and Cognition 23, 709–709 (1995)
23. Suwa, M., Tversky, B.: External representations contribute to the dynamic construction of ideas. In: Hegarty, M., Meyer, B., Narayanan, N.H. (eds.) Diagrams 2002. LNCS (LNAI), vol. 2317, pp. 341–343. Springer, Heidelberg (2002)
24. Aldrich, F.K., Parkin, A.J.: Tangible line graphs: an experimental investigation of three formats using capsule paper. Human Factors 29(3), 301–309 (1987)
25. Hegarty, M.: Mental animation: Inferring motion from static displays of mechanical systems. Journal of Experimental Psychology: Learning, Memory, and Cognition 18(5), 1084–1102 (1992)
26. Tufte, E.: The visual display of quantitative information. Graphics Press (1986)
27. Johnson-Laird, P.: Mental models: Towards a cognitive science of language, inference and consciousness. Harvard Univ. Press, Cambridge (1983)
28. Shneiderman, B.: The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In: Proc. 1996 IEEE Symp. on Visual Languages. IEEE Computer Society, Washington (1996)
29. Chi, E.: A taxonomy of visualization techniques using the data statereference model. In: IEEE Symp. on Information Visualization, pp. 69–75 (2000)
30. Goodman, N.: Languages of art: an approach to a theory of symbols. Bobbs-Merrill, Indianapolis (1968)
31. Stenning, K., Oberlander, J.: A cognitive theory of graphical and linguistic reasoning: Logic and implementation. Cognitive Science 19(1), 97–140 (1995)

# The Effects of Signals on Learning from Text and Diagrams: How Looking at Diagrams Earlier and More Frequently Improves Understanding

Katharina Scheiter and Alexander Eitel

Knowledge Media Research Center, Konrad-Adenauer-Str. 40,
72072 Tübingen, Germany
{k.scheiter,a.eitel}@iwm-kmrc.de

**Abstract.** In an eye tracking study 35 students learned about the functioning of the heart. In a no-signals condition, a text and diagram were presented in an unaltered fashion. In the signals-condition, correspondences between the representations were highlighted by means of labels, color coding, and deixis. The signals improved understanding of the correspondences between verbal and diagrammatic information as well as led to more attention being devoted to the diagrams. Moreover, diagrams were fixated earlier in the signals- compared to the no-signals condition. A mediation analysis showed that the changes in visual attention were sufficient to completely explain the effect of signals on learning outcomes. Hence, signals improve learning from text and diagrams by fostering learners' early reference to diagrams and by increasing the amount of attention devoted to them.

**Keywords:** learning, multimedia, signaling, eye tracking.

## 1 Introduction

The use of text and diagrams has a long-standing tradition in education. Various arguments have been brought forward for why diagrams may aid understanding. Diagrams facilitate visuo-spatial reasoning [1] by representing related information in a spatially contiguous manner. Moreover, they support inferences and reasoning processes that are grounded in perception, allowing perceptual judgments to substitute for more demanding logical inferences [2; 3; 4]. Moreover, diagrams may serve additional instructional functions in the service of text comprehension. Levin, Anglin and Carney [5] have suggested that diagrams that depict objects and relations already mentioned in a text make the meaning of the text more concrete for learners, provide an organizational framework for a text by highlighting its structure, and make especially complex text better comprehensible for learners.

In order to benefit from text and diagrams, however, it is important that learners build a coherent mental representation of the content, where information from the text is related to information from the diagram by identifying the structural correspondences between the two representational formats [6; 7]. One important question is how this coherent mental representation is constructed. For instance, in the Cognitive

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 264–270, 2010.

Theory of Multimedia Learning [7] it has been suggested that learners will first build two separate mental models of both information sources in isolation, before integrating these models into a single coherent mental representation with the help of prior knowledge. This suggests that the processing of text and diagrams occurs independently of each other and only after each of the two representations has been understood in isolation, references between the two are established. However, it seems more likely that interactions between the processing of either text or diagrams will occur in that a diagram may aid the understanding of the text and vice versa.

Evidence in favor of the latter assumption comes from studies that have used eye tracking to assess text and diagram processing. By recording a person's eye movement while s/he is studying a text-diagram presentation, information is obtained on when and how much visual attention was devoted to processing the representations [8; 9]. Most importantly, Just and Carpenter [10] have suggested that what is being attended or fixated reflects what is being processed at the cognitive level. Therefore, eye tracking can provide valuable insights into the perceptual as well as presumably the conceptual processing of text and diagrams. For instance, an eye tracking study by Hegarty and Just [6] on learning about pulley systems suggests that the processing of diagrams is largely guided by the accompanying text, where interruptions of the reading process occur mostly at the end of a larger semantic unit. Then attention is shifted to those components in the diagram that have been mentioned in the previously read text. Afterwards, attention is again devoted to the text.

More recently, there have been a number of eye tracking studies that investigated learning from text and/or diagrams depending on the degree of instructional support provided to learners [e.g., 11; 12; 13; 14; 15; 16]. In particular, these studies have focused on the role of signaling in instruction. Signals (or cues) are instructional methods that are based on manipulations of the layout of the representations (but not their content) in a way that relevant information can be more easily identified. Their effect is presumably due to the fact that attention is guided towards this information and the need for visual search is reduced [15]. Signals can refer to highlighting relevant information in the text (e.g., underlining, italics, bold-face, enumeration etc.) or in the diagram (e.g., by inserting arrows or using color and contrasts; [11; 12; 14]). Most importantly, signals can be used to emphasize the relation between verbal and diagrammatic information, for instance, by making deictic references (i.e., words or sentences that specify the referent in the diagram, e.g., "as can be seen in the upper part of the diagram), corresponding labels that appear in the text as well as in the diagram, or color coding, where the same color is applied to print a relevant word or phrase as to depict its corresponding diagram element [13; 16].

Despite the increasing number of studies devoted to studying the impact of signaling on text-diagram comprehension, it is not yet clear how exactly and when they may foster learning. In particular, signals have been shown to lead to changes in visual attention distribution (i.e., more fixations and/or longer overall dwell times on the signaled information elements) without yielding the expected benefits for learning outcomes [e.g., 12; 13; 14], thereby emphasizing that looking at relevant information does not guarantee deeper conceptual processing of that information. Moreover, it can be expected that effects of signaling depend on the type of instructional materials with effects of signaling presumably becoming, for instance, more pronounced for more complex materials. Also the granularity of what is being signaled may matter. That is, in many instructional materials and especially those on causal systems (e.g., on the functioning of a mechanical device or of a biological system like the heart) one does not simply want students to attend to an isolated element of that system. Rather, better learning outcomes as a function of signaling are to be expected only if the signaling leads to an increase in the amount of attention devoted to this element *and* its functional relation to other elements so that learners can elaborate on the role of the signaled element within the causal systems. Hence, the signaling of isolated elements may not be as effective as the signaling of larger structural units, unless learners use the signaled isolated element as a starting point for their inspection of the functional aspects depicted in the diagram. In the latter case, signaling should not lead to an increase in the amount of attention devoted to signaled elements only, but rather increase attention to non-signaled, but functionally related elements as well. Finally, current research on signaling has not yet answered the question satisfactorily of whether changes in visual attention cause (or at least are related to) changes in learning outcomes. Rather, it has been inferred from the fact that both, changes in visual attention and comprehension co-occurred, that the prior changes must have been responsible for having caused the latter (for exceptions see [15; 16]. Hence, the current study aimed first at investigating the effect of signaling related to text-diagram integration on learning outcomes and visual attention for signaled vs. non-signaled diagrammatic elements as well as second at determining whether changes in visual attention can explain improvements in comprehension achieved by means of signaling.

# 2 Experiment

## 2.1 Method

**Participants and design.** Thirty-five university students (16 male and 19 female, M = 24.94 years) with equal levels or prior knowledge participated in the study. The experiment consisted of a control (no-signals) condition and a signaling condition.

**Materials and procedure.** Students had to learn about the functioning of the heart. The instructional materials consisted of 16 slides with each slide containing a written text and a related diagram (Figure 1). The first slide contained a diagram of the heart with labels for all the technical terms that were used in the subsequent learning materials. It was the same for both conditions. For the remaining slides the text and diagrams were presented in an unaltered fashion in the control condition. In the signaling-condition the text was structured into paragraphs to facilitate the identification of larger semantic units that may trigger diagram inspections [6]. In addition, three signaling methods for highlighting text-diagram correspondences were used, whereby their use on a particular slide depended on the appropriateness of the method in the given context. First, deictic references to the diagram were made. Second, important words were highlighted in the text (italics) and used as labels in the diagram. Third, color coding was implemented. Students in both conditions were instructed that they could navigate forward (but not backward) through the slideshow at their own pace by pressing the space bar. During learning, eye movements were assessed with a video-based eye tracking system (SMI iViewX$^{TM}$ Hi-Speed; 500 Hz). Subsequently, students had to work on a posttest assessing learning outcomes.

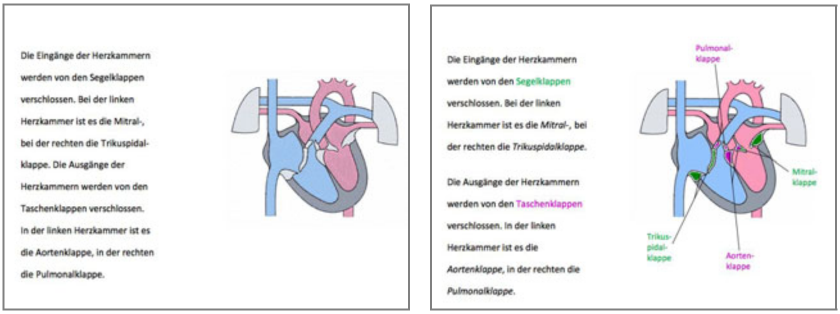



**Fig. 1.** Sample materials for the control (*left panel*) and the signaling condition (*right panel*). The signals for this slide were color coding for the terms in the text, labels, and corresponding diagram parts as well as highlighting the labels used for the diagram in the text (italics).

**Measures.** We assessed learning outcomes in terms of verbal recall, transfer, and text-diagram integration. Nine items were multiple-choice questions assessing the recall of verbal information. Six multiple-choice items assessed the students' ability to draw inferences based on the given information (transfer). Always one point was awarded for each correct answer. In the final 4 questions, diagrams from the learning phase (without signals) were shown to participants and they were asked to recognize the phase in which the heart was depicted, answer specific questions related to the diagram and draw single elements as well as the blood flow into the diagram. These items assessed the ability to integrate verbal and diagrammatic information. A total of 10 points could be obtained in this part of the posttest. Moreover, as indicators for the processing of text and diagrams the number of fixations on either text or diagrams, the overall dwell time on both (in seconds), as well as the time that had elapsed until attending to either of the two representations (time until first fixation in milliseconds) were recorded. Finally, we defined areas of interest (AoIs) on the diagram for those elements that had been signaled on a particular slide and for the remaining, non-signaled elements. For both AoI types, the dwell time in milliseconds was recorded. Because different elements of the diagram had been signaled over the course of instruction, this was done separately for each slide.

## 2.2 Results

Learning outcomes were analyzed by means of a one-factorial MANOVA yielded a marginal overall main effect, $F(3, 31) = 2.64$, $p = .07$. The univariate ANOVAs showed a positive effect of signaling for the text-diagram questions (control condition: $M = 4.33$, $SD = 2.03$; signaling condition: $M = 6.41$, $SD = 2.79$, $F(1, 33) = 6.42$, $p = .02$), but not for verbal recall, $F(1, 33) = 2.85$, $p = .10$, or transfer, $F < 1$.

To investigate whether signaling enhanced visual attention for the signaled information only or also for the non-signaled elements, three slides were selected for analysis, which each contained only one of the signaling methods (either deixis, italics + labels, or color coding). Hence, the effects could be traced back unambiguously to one of these signaling methods. A 2x2-mixed design ANOVA with experimental condition serving as a between-subjects factor and element type (signaled/non-signaled) as a within-subjects factor was conducted for the dwell time in the respective AoIs. Only the main effects for experimental condition and the interaction will be reported here, because comparisons between AoIs (i.e., main effects of element type) are not interpretable as the AoIs had been of different sizes. All three signaling methods increased the overall dwell time irrespective of element type (deixis: $F(1, 33) = 10.87$, $p = .002$; italics + labels: $F(1, 33) = 19.20$, $p < .001$ color coding: $F(1, 33) = 5.86$, $p = .02$;, see Table 1 for means and standard erors). However, there were either no interactions of signaling and element type (deixis: $F < 1$) or there was one, but in an unexpected direction (italics + labels: $F(1, 33) = 21.34$, $p < .001$; color coding: $F(1, 33) = 6.71$, $p = .01$). That is, italics + labels did not have an effect for visual attention devoted to the signaled element (control condition: $M = 2467.89$, $SD = 3100.20$; signaling condition: $M = 2183.61$, $SD = 2011.62$, F(1, 33) = 1.02, $p = .75$), but increased attention concerning the non-signaled elements only (control condition: $M = 2244.25$, $SD = 2541.93$; signaling condition: $M = 8706.56$, $SD = 4003.03$, F(1, 33) = 32.90, $p < .001$). The same pattern was true for the slide containing color coding, which had no effect on the signaled elements (control condition: $M = 9178.64$, $SD = 7768.59$; signaling condition: $M = 9805.53$, $SD = 8948.44$, F < 1), but yielded longer dwell times on non-signaled elements only (control condition: $M = 984.45$, $SD = 3566.67$; signaling condition: $M = 9452.82$, $SD = 7230.33$, F(1, 33) = 19.65, $p < .001$). Thus, signals had stronger effects on guiding attention to non-signaled versus signaled elements. These effects were not limited to elements in close spatial or functional proximity to the signaled elements only, but extended to the whole diagram.

**Table 1.** Means and standard errors for the effects of signaling on selected slides (in ms)

| | Control condition | Signaling condition |
|---|---|---|
| Deixis | 1395.39 | 2705.86 |
| | (276.80) | (284.83) |
| Italics + labels | 2356.07 | 5445.09 |
| | (491.26) | (505.51) |
| Color coding | 5081.55 | 9629.18 |
| | (1309.82) | (1347.80) |

To investigate how learners had distributed their attention across both representations, the number of fixations, the overall dwell time, and the time elapsing until the first fixation were analyzed by separate MANOVAs for the text-related measures and for the diagram-related measures, respectively. Both analyses revealed significant overall main effects (text-related: $F(3,31) = 4.99$, $p = .006$; diagram-related: $F(3,31) = 4.53$, $p = .01$). Subsequent ANOVAs showed that signaling had no impact on the number of fixations on text, $F(1,33) = 1.61$, $p = .15$, or on the overall dwell time on text, $F(1,33) = 2.13$, $p = .21$, but that it increased the time until text was fixated for the first time (control condition: $M = 215.91$ ms, $SD = 144.13$; signaling condition: $M = 549.13$ ms, $SD = 325.40$, $F(1,33) = 15.65$, $p < .001$). Regarding the diagram-related measures, signaling increased the number of fixations on diagrams (control condition: $M = 279.94$, $SD = 107.53$; signaling condition: $M = 459.88$, $SD = 185.78$, $F(1,33) = 12.48$, $p = .001$), the overall dwell time (control condition: $M = 81.41$ s, $SD = 32.44$;

signaling condition: $M = 122.93$ s, $SD = 47.88$, $F(1,33) = 9.12$, $p = .005$), and reduced the time that elapsed until a diagram was fixated for the first time (control condition: $M = 4391.79$ ms, $SD = 1883.99$; signaling condition: $M = 2916.97$ ms, $SD = 1546.53$, $F(1,33) = 6.36$, $p = .02$). Hence, signaling had a strong impact on the processing of text and diagrams and particularly fostered the processing of the diagrams, which were attended more frequently, longer, and earlier during learning.

To investigate whether the changes in the processing of text and diagrams could explain the improvement in learning outcomes achieved by signaling, we conducted a mediation analysis [17]. In this analysis, which is based on multiple regression, the *total effect c* that signals had on text-diagram integration questions (see ANOVA results reported earlier) was separated into the *indirect effect c-c'* that was mediated by the changes in processing text and diagrams and the remaining *direct effect c'* that could not be explained by the mediating processing variables (Figure 2). The number of fixations on diagrams, time until fixating the diagram for the first time, and time until fixating the text for the first time acted as mediators. The chosen variables completely mediated the effect of signals on text-diagram integration questions (indirect effect: $p = .03$; $R^2 = .30$; remaining non-mediated, i.e., direct effect of signals on text-diagram integration questions: $p = .52$).

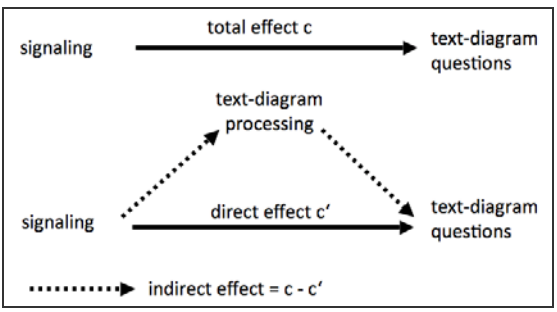



**Fig. 2.** Mediation analysis for determining the indirect (i.e., mediated) effect of signaling on text-diagram questions

## 3 Summary and Discussion

In the current study it was shown that signals highlighting the relation between text elements and corresponding diagrammatic elements improved the ability to answer questions requiring the integration of both representational formats. The analysis of the eye tracking data revealed that this effect could be explained by changes in the distribution of attention. In particular, signals increased the overall attention to the diagram and led to attention being guided towards the diagram earlier in the process of studying the representations. Our results differ from those obtained by Oczelik and colleagues [15; 16] in that the effects of signals were not limited to paying more attention to signaled elements; rather, they were even stronger for non-signaled elements. This suggests that especially in diagrams about causal systems, it is important to not only attend to isolated elements, but to understand their functional role within the system. Hence, students may use the signaled element as an entry point to study the system and its interrelated components as a whole. However, to prove this assumption, we need to analyze the time course of attending to the text and diagrams at a more fine-grained level yet.

## References

1. Larkin, J.H., Simon, H.A.: Why a Diagram is (Sometimes) Worth Ten Thousand Words. Cognitive Sci. 11, 65–99 (1987)
2. Goldstone, R.L., Son, J.Y.: The Transfer of Scientific Principles using Concrete and Idealized Simulations. J. Learn. Sci. 14, 69–110 (2005)
3. Schwartz, D.L.: Reasoning about the Referent of a Picture versus Reasoning about the Picture as a Referent: An Effect of Visual Realism. Mem. Cognition 23, 709–722 (1995)
4. Salomon, G.: Media and Symbol Systems as Related to Cognition and Learning. J. Educ. Psychol. 71, 131–148 (1979)
5. Levin, J.R., Anglin, G.J., Carney, R.N.: On Empirically Validating Functions of Pictures in Prose. In: Willows, D.M., Houghton, H.A. (eds.) The psychology of illustration, vol. 1, pp. 51–85. Springer, New York (1987)
6. Hegarty, M., Just, M.A.: Constructing mental models of machines from text and diagrams. J. Mem. Lang. 32, 717–742 (1993)
7. Mayer, R.E.: Multimedia Learning, 2nd edn. Cambridge University Press, New York (2009)
8. Scheiter, K., van Gog, T.: Using Eye Tracking in Applied Research to Study and Stimulate the Processing of Information from Multi-representational Sources. Appl. Cognitive Psych. 23, 1209–1214 (2009)
9. Van Gog, T., Scheiter, K.: Eye Tracking as a Tool to Study and Enhance Multimedia Learning. Learn. Instr. 20, 95–99 (2010)
10. Just, M.A., Carpenter, P.A.: A Theory of Reading: From Eye Fixations to Comprehension. Psychol. Rev. 87, 329–355 (1980)
11. Boucheix, J.-M., Lowe, R.K.: An Eye Tracking Comparison of External Pointing Cues and Internal Continuous Cues in Learning with Complex Animations. Learn. Instr. 20, 123–135 (2010)
12. De Koning, B.B., Tabbers, H.K., Rikers, R.M.J.P., Paas, F.: Attention Guidance in Learning from a Complex Animation: Seeing is Understanding? Learn. Instr. 20, 111–122 (2010)
13. Folker, S., Ritter, H., Sichelschmidt, L.: Processing and Integrating Multimodal Material – The Influence of Color-Coding. In: Bara, B.G., Barsalou, L., Bucciarelli, M. (eds.) Proceedings of the 27th Annual Cognitive Science Conference, pp. 690–695. Erlbaum, Mahwah (2005)
14. Kriz, S., Hegarty, M.: Top-Down and Bottom-Up Influences on Learning from Animations. Int. J. Hum-Comput. St. 65, 911–930 (2007)
15. Ozcelik, E., Arslan-Ari, I., Cagiltay, K.: Why Does Signaling Enhance Multimedia Learning? Evidence from Eye Movements. Comp. Hum. Behav. 26, 110–117 (2010)
16. Ozcelik, E., Karakus, T., Kursun, E., Cagiltay, K.: An Eye-Tracking Study of How Color Coding Affects Multimeda Learning. Comp. Educ. 53, 445–453 (2009)
17. Preacher, K.J., Hayes, A.F.: SPSS and SAS Procedures for Estimating Indirect Effects in Simple Mediation Models. Behav. Res. Meth. Instr. C 36, 717–731 (2004)

# An Attention Based Theory to Explore Affordances of Textual and Diagrammatic Proofs

Peter Coppin[1,⋆], Jim Burton[2,⋆⋆], and Stephen Hockema[1]

[1] Faculty of Information, University of Toronto, Ontario, Canada, M5S 3G6
{peter.coppin,steve.hockema}@utoronto.ca
[2] Visual Modeling Group, CMIS, University of Brighton, UK BN2 4GJ
j.burton@brighton.ac.uk

**Abstract.** Shimojima and Katagiri have demonstrated that diagrams reduce "inferential load" during reasoning by scaffolding visual-spatial aspects of memory. In response, we wondered why, if this is true, that proofs are usually text based? The purpose of this paper is to explore ergonomic affordances of text that may encourage its use in the communication of proofs by building on prior work in attention. We claim that textual notations may focus a reasoner's "spotlight" of attention through serialized sequential chunks, whereas many diagrams may "diffuse" attention and that a diagrammatic notation system that serialized information in chunks amenable to focused attention could leverage the power of textual notations. We present such an example through a case study focused on generalized constraint diagrams, a visual logic with attributes that may support focused attention and extract ergonomic principles that may transcend each notation system.

**Keywords:** attention, visual thinking, proof, logic, geometry.

## 1 Introduction

Why are most geometric proofs usually "non-visual", i.e., textually based? In previous work [3] we asked this question because Shimojima and Katagiri [10] demonstrated that diagrams reduce "inferential load" during reasoning by scaffolding visual-spatial aspects of memory and attention (cf. [1,11]). We reasoned that, if diagrams reduce inferential load, then notation systems that include diagrams as "first-class citizens" should be inherently useful. However, as Tennant [13] described: "[The diagram] is only an heuristic to prompt certain trains of inference; ... it is dispensable as a proof-theoretic device; indeed ... it has no proper place in a proof as such. For the proof is a syntactic object consisting only of sentences arranged in a finite and inspectable array." (quoted in [2]). Based on this trend, we sought to explore ergonomic affordances of text relative to diagrams that may encourage the use of diagrams in proofs. We did this by

⋆ Peter Coppin is supported by an NSF Graduate Research Fellowship.
⋆⋆ Jim Burton is supported by a UK EPSRC doctoral grant.

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 271–278, 2010.

examining the role of text and diagrams in Proposition 35 of Euclid's elements (detailed in [3]).

Before going much further, it is important to review some common distinctions; one must distinguish between formal and informal proofs, and between different cognitive tasks such as proof generation, presentation and comprehension. This paper investigates features that distinguish the comprehensibility of diagrams from a human cognitive perspective; it follows therefore that we are interested in informal proofs, since formal proofs are not primarily designed for human use. Furthermore, we are interested in diagrams as tools of communication to present and communicate proofs, rather than in their use as primary reasoning tools used to generate proofs.

To use Shimojima's example to demonstrate the distinction between text and diagrams, if block $A$ is under block $B$ and block $C$ is above block $B$, then logic tells us that $A$ is below $C$. Alternatively, we can easily induce that $A$ is below $C$ by observing a diagram, yet the logical text-based proof is considered more rigorous than a diagram [2]. Indeed, for generations, Euclid's Elements was considered to be flawed because of its reliance on diagrams. As Mumma [7] described: "for some of Euclid's steps, the logical form of the preceding sentences is not enough to ground the steps. One must consult the diagram to understand what justifies it." For this reason it is commonly felt that Euclid "failed in his efforts to produce (an) exact, full explicit mathematical proof" [7].

By building on prior work in attention [9,14], we claimed that textual notations may focus a reasoner's "spotlight" of attention through serialized sequential chunks, enabling the methodical presentation of a rational argument in a way that is not possible (or at least very difficult given current understandings and practices) when using a diagrammatic notation that may "diffuse" attention; such notations may enable a reasoner to discern how elements fit together holistically, but place less focus on individual steps in the argument, or on reifying and explicitly representing connections/relationships between steps.

Given that there are social and psychological considerations associated with the uptake and widespread acceptance of a notation and inference system, our theory can help to explain why some systems appeal to mathematicians more than others in the context of specific tasks. Though text was described as a focal notation system in [3], we also suggested that a diagrammatic system could be constructed that also afforded focal attentional processes. One purpose of this paper is to explore such an example through a case study focused on the visual logic of generalized constraint diagrams [12]. We will describe the affordances of this notation system in relation to our findings from our past exploration of Euclid's proof.

## 2 Literature Review

Larkin and Simon claimed in [5] that a cognitive dimension of sentences is their list-like structure, in that each item on the list is only adjacent to the item before or after it on the list. In contrast, items in a diagram are adjacent to many items on a list. This view is synergistic with Barwise and Etchemendy [2], who suggested that a picture or diagram can support "countless facts". In other words, many sentences could be created by linking together elements in a diagram into a sentence (list-like structure). Each sentence inferred from a diagram is a path that guides attention through visual-spatial relationships in a diagram. This observation is demonstrated by Shimojima and Katigiri's eye tracking study in [10] that showed how reasoners mentally guide their attention through a "non-physical drawing." They suggest that actual drawings support non-physical (mental) drawings, thus reducing inferential load. To summarize, it appears that sentences guide attentional paths through both physical and non-physical (mental) visual-spatial structures. Further, Shimojima and Katigiri demonstrated that rational language/propositional logic guides attention and motor movements through non-physical visual-spatial representations.

Mumma approached our question by examining the "flaws" in Euclid's picture proofs and demonstrated that they are rigorous through his proof Eu [8]. Mumma locates the answer in the subject matter and norms of the field (generality problem, modern understanding of continuity, and the modern axiomatic method); a detailed review can be found in [3].

Treisman [14] described focused attention (FA), suggesting that "attention must be directed serially to each stimulus in a display whenever conjunctions of more than one separable feature are needed to characterize or distinguish the possible objects presented." By describing features as separable, she means primitives such as basic shapes and colors prior to integration into a conceptual whole. Treisman was using a spotlight/zoom lens metaphor, claiming that attention can either be narrowed to focus on a single feature, when we need to see what other features are present and form an object, or diffused over a whole group of items which share a relevant feature [14]. Relative to diffused attention (DA), FA is more precise and detailed. We suggest that textual language guides FA through such structures to build more precisely focused but less holistic ideas, thus responding to our previous question on the origin of an experience of rigor.

So-called object-based attention [9] claims that attention may automatically ("pre-attentively") spread as a "bottom up" process within "groups" of objects. However, specific queries and other factors can influence a "top down" process that changes which groups are perceived as objects. For example, Scholl [9] described how an initial gestalt grouping of two intersecting lines can change to a different gestalt grouping based on a statement (e.g. identify the bird beaks in a field of intersecting lines). A text description in a proof (e.g., "consider vertices A-B-C-D") could cause one to perceive a quadrilateral within a field of many intersecting lines. For our purposes, this maps to our usage of FA and DA.

## 3 Theory: Text Affords Rigor by Directing Focal Attention

Before presenting our theory we need to define what we mean by the relative feeling of "rigor" experienced by a user with regard to two mathematical notations.

The rigor of a certain notation is a mathematical quality which is either present or missing. In the context of this work, however, the notion of rigor is used to quantify the clarity and persuasiveness of a notation, resulting in a greater or lesser degree of confidence in the users of the notation. The users we have in mind are mathematicians, and the various notations are used to communicate with colleagues via proofs. An experience of rigor may be related to but separate from logical soundness. A mathematician may make use of a visual notation with a formal semantics as an integral part of a proof with confidence that the results are sound. If a formal diagrammatic element of a proof fails to inspire a feeling of rigor amongst readers, perhaps because the notation comes with semantics that is complex and unfamiliar, the proof may have failed to fulfil its purpose as communication tool. Conversely, the same person may deploy an informal notation as an aid to reasoning with confidence, when she expects the meaning will be clear in the context it is used. Our theory aims to provide a partial explanation for the clarity and confidence experienced by users of textual proofs and extract features which can be designed for in primarily visual notations.

**Why proofs are often sequential:** We suggest that focal attention must "walk" through different parts of a conceptual visual-spatial structure due to "narrower," but more intense focus (rather than attending to the whole structure at once).[1]

**Why a diagram cannot usually constitute a convincing "holistic" proof:** The need for symbols to fall within the narrow spotlight of focal attention means that diagrams, and the spatial relationships they embody, cannot usually be attended to all at once. So they should instead be processed in a way that allows linkages between earlier perceptual memories and later percepts.

**Why text is effective for proofs:** External symbolic representations such as text enable each symbol to sequentially fall within the narrow focus of focal attention.

**Why propositional logic is used for proofs:** Propositional structures in proofs may provide symbolic "short-cuts," serving as stand-ins for visual-spatial relationships that cannot all simultaneously be in the narrow spotlight of focal attention. For example, the statement "if C is below B" references a perceptual symbol constructed from a previously considered image, as does the statement "if B is below A". The conclusion "therefore C is below A" references the two previous symbols in order to support construction of a new mental image that can serve as the basis for a new perceptual symbol and be used in future propositional statements.

To summarize, we suggest that a visual-spatial structure such as a geometric structure (irregardless of whether it is presented as a diagrammatic representation) is often beyond the narrow spotlight of FA. The purpose of sequential symbolic representations such as textual propositional statements is to guide

[1] The sequence implied here does not imply a particular ordering. We assume that many factors play into determining the order in which structures must be attended.

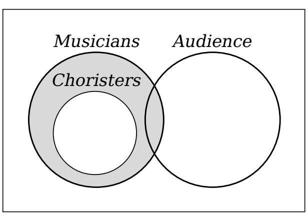
FA through a sequence of patterns in order to create perceptual symbols that are amenable to analytical neurological machinery. Hence, in addition to being derived from the norms of the field of mathematics and the other social and technical reasons that have been proposed, we argue the answer to our initial question is related to basic facts about human cognition. Based on our theory above, we summarize the design criteria of a focal notation system:

1. Atoms of meaning that are sparse enough, relative to DA, to be perceived using focal attention.
2. A sequential relationship of sentences.
3. A hierarchical structure whereby complex relationships can be explored by referencing previous experiences (perceptual symbols) in the sequence.

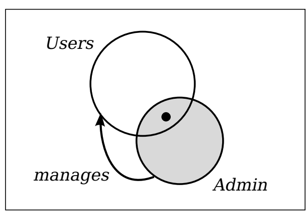
In other words, a focal notation system could resemble the ergonomics of textual proofs. The next section examines a primarily diagrammatic notation which, we argue, is aligned with these design criteria and thus includes attributes that may support FA. A comparison to findings from [3] enables the extraction of ergonomic principles that transcend each system.

## 4 Applying the Theory to Constraint Diagrams

Constraint diagrams [4] are a primarily diagrammatic logic based on the Euler diagram notation which makes use of topological properties of separate and overlapping circles to represent sets, subsets, disjointness and intersection. Shading represents emptiness. Circles are labeled to indicate the set they represent. The Euler diagram in figure 1a asserts relationships between three groups, *Musicians*, *Choristers* and *Audience*. Constraint diagrams build on Euler diagrams




(a) An Euler diagram




(b) A constraint diagram

**Fig. 1.** Euler diagrams and constraint diagrams

by adding syntax to represent quantification and binary relations. An example of a constraint diagram is shown in figure 1b. The black dot is called a *spider* and represents the existence of an element in the sets within both *Users* and *Admin*. This is a so-called *existential* spider; asterisks represent universal quantification. The arrow tells us that there is a relation called *manages* and that, informally, every administrator manages users and every user is managed.

Generalized constraint diagrams were proposed by Stapleton and Delaney [12] as the result of an analysis of the positive and negative features of the original notation; a full description of their syntax and semantics can be found in [12], and we draw attention to only those features which are necessary to follow our argument. GCDs impose a reading order on diagrammatic elements by allowing them to be introduced piece-wise, in what may be thought of as a movie strip or timeline. Figure 1b is a *unitary* diagram; a GCD may include more than one unitary diagram in a directed tree structure. The root node of such a tree is labeled by a unitary diagram and each node below the root is labeled by either a unitary diagram or by one of the logical connectives $\wedge$ and $\vee$, represented by a fork and vertical bar respectively. Figure 2 shows an example, whose informal meaning is that either all dogs bury bones or none of them do. Each unitary

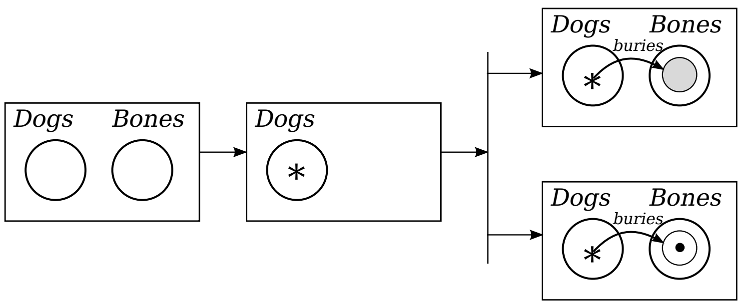



**Fig. 2.** A generalized constraint diagram

diagram labeling a node below the root of a GCD may add some new piece of information to the sum of the information provided by its ancestors. Crucially for our purposes, it need not restate the information provided by its ancestors and, by removing syntax, it may represent less information. Constraints are placed on what may be added and removed to ensure that no ambiguity arises.

### 4.1 Discussion: GCDs in Relation to Our Theory

The usability of GCDs was addressed in [12], where the authors state five principles that guided the design of the notation:

1. *Well-matchedness principle*: that syntactic relations mirror semantic ones.
2. *Explicitness principle*: make the informational content explicit, not implicit.
3. *Interpretation principle*: ensure that the semantics assigned to each piece of syntax are independent of context.
4. *Expressiveness principle*: allow statements to be made naturally.
5. *Construction principle*: impose only the necessary restrictions on what constitutes a well-formed statement [12].

The authors justify the claim that GCDs satisfy these principles by matching each principle to several aspects of the syntax and meaning of GCDs. Here, we are more interested in aspects of GCDs that may support focal attention; we believe that the most significant of these are the tree structure and the ability to add information piece-wise. These properties allow GCDs to match the design criteria of our ideal notation from section 3.

1. Atoms of meaning that are sparse enough to be perceived using focal attention: Each diagram-labeled node may present a single piece of information in a way that allows it to be understood independently of context. In figure 2 chunks of information are added at each step. Redundant information is suppressed so that diagrams remain optimally sparse and the most relevant information is highlighted; for instance the circle labeled "Bones" in the first node identifies a set. It would be legitimate to represent that set in the second node, but it would add no new information and result in a more cluttered diagram. New information may be introduced, and existing information highlighted, strategically to convey a step in an argument. This property is related to Stapleton and Delaney's *interpretation* principle.
2. Support for a means to deliver information serially in a way that guides the reader to understanding: The ability to introduce information piece-wise and to restate information at key junctures matches the perceptual chunking of text-based proofs well. This gives GCDs a strong potential to perform the "story telling" duty of a proof, in which the objective is to persuade the reader of the logical argument the proof contains. This property is related to the principles of *well-matchedness* and *explicitness*.
3. Sentences composed in a hierarchical structure: The tree structure of GCDs supports this property. As stated, the underlying meaning of figure 2 has a disjunctive structure; this is mirrored by the fork in the tree, which leads the viewers gaze along each branch.

A defining characteristic of GCDs is that few restrictions are imposed on the user; this is reflected in Stapleton & Delaney's *construction principle*. It is quite possible, therefore, to use them in an undisciplined way which tends to diffuse attention and negate the positive features identified above. However, the same may be said of any notation, including symbolic logic. Similarly, there are aspects of GCDs which mitigate against focal attention; one example are the so-called "free rides" which occur when a diagram enables an inference to be made based on implicit information [6]. In both of these cases it is the *potential* of GCDs to support the design criteria which interests us, rather than an unerring ability to do so.

## 5 Conclusion

In this paper we identified ergonomic properties which, we have argued, support focal attention and which are therefore desirable for certain reasoning tasks. We have discovered these properties in an analysis of generalized constraint diagrams, which are primarily diagrammatic. In this sense, GCDs share properties with more symbolic notations which are likely to enhance their suitability to convey arguments composed of axioms and steps of inference. It seems likely that certain notation systems and perhaps all, to some degree, may be used in a way which shifts between the focal and diffuse modes; on this theme, we conclude with a question that will guide our unfolding research: "What is the boundary between focal and diffuse notations?"

**Acknowledgements.** The authors wish to thank the anonymous reviewers for their feedback, which was extremely helpful.

## References

1. Ballard, D.H., Hayhoe, M.M., Pook, P.K., Rao, R.P.N.: Deictic codes for the embodiment of cognition. Behavioral and Brain Sciences 20, 723–767 (1995)
2. Barwise, J., Etchemendy, J.: Visual information and valid reasoning. In: Logical Reasoning with Diagrams, pp. 3–25. Oxford University Press, Inc., Oxford (1996)
3. Coppin, P., Hockema, S.: A cognitive exploration of the "non-visual" nature of geometric proofs. In: Cox, P., Fish, A., Howse, J. (eds.) Visual Languages and Logic, pp. 81–95 (September 2009)
4. Gil, J., Howse, J., Kent, S.: Towards a formalization of constraint diagrams. In: Proc. IEEE Symposia on Human-Centric Computing (HCC 2001), Stresa, Italy, pp. 72–79. IEEE Computer Society Press, Los Alamitos (2001)
5. Larkin, J.H., Simon, H.A.: Why a diagram is (sometimes) worth ten thousand words. Cognitive Science 11(1), 65–100 (1987)
6. Lemon, O., de Rijke, M., Shimojima, A.: Editorial: Efficacy of diagrammatic reasoning. Journal of Logic, Language and Information 8(3), 265–271 (1999)
7. Mumma, J.: Ensuring generality in euclid's diagrammatic arguments. In: Diagrammatic Representation and Inference, pp. 222–235 (2008)
8. Mumma, J.: Proofs, pictures, and euclid. In: Synthese (2009)
9. Scholl, B.J.: Objects and attention: The state of the art. Cognition 80(1-2), 1–46 (2001)
10. Shimojima, A., Katagiri, Y.: An Eye-Tracking study of exploitations of spatial constraints in diagrammatic reasoning. In: Diagrammatic Representation and Inference, pp. 74–88 (2008)
11. Spivey, M.J., Geng, J.J.: Oculomotor mechanisms activated by imagery and memory: eye movements to absent objects. Psychological Research 65(4), 235–241 (2001)
12. Stapleton, G., Delaney, A.: Evaluating and generalizing constraint diagrams. Journal of Visual Languages and Computing 19(4), 499–521 (2008)
13. Tennant, N.: The withering away of formal semantics? Mind & Language 1(4), 302–318 (1986)
14. Treisman, A.M., Gelade, G.: A feature-integration theory of attention. Cognitive Psychology 12(1), 97–136 (1980)

# Effects of Graph Type in the Comprehension of Cyclic Events

Özge Alaçam[1], Annette Hohenberger[2], and Kürşat Çağıltay[3]

[1] Human Computer Interaction Rsrch. and Appl. Laboratory, Computer Center,
[2] Department of Cognitive Science, Informatics Institute,
[3] Departments of Computer Education and Instructional Technologies,
Middle East Technical University, 06531, Ankara/Turkey
{ozge,kursat}@metu.edu.tr, hohenberger@ii.metu.edu.tr

**Abstract.** This study presents an analysis of the effect of different graph types on the comprehension of cyclic events. The results indicated that although round and linear graph designs are informationally equivalent, the round graphs are computationally better suited than linear graphs for the interpretation of cyclic concepts.

**Keywords:** Graph comprehension, eye tracking, and cyclic concept, round graphs.

## 1 Introduction

Graphs are a very efficient way of representing and conveying relations between variables. They can be used for extracting a single piece of information from the graph, comparing two or more pieces of information, or they are also frequently used for determining trends. They are also used for extracting information that may not even be explicitly represented in the graph [1]. Most popular graphs are based on three basic types: line graph, bar chart or pie chart. Different graph types differ in pointing out specific features about the data [2]. There are lots of studies that generally lead to the conclusion that the given tasks or the aims of the graph readers affect the comprehension [3]. In addition to the effect of task and graph type, the matching of particular event types with appropriate graph types and graph designs should play a role in graph comprehension. However, while the effects of task and graph type have been well- studied, the effects of event type and graph design need more investigation.

This study[1] aims to investigate whether the circularity of the graph design affects the comprehension of cyclic events. The winter season comprising December, January and February is an example of a cyclic concept. It reoccurs each year. In order to extract information presented by a linear graph about what change happens in winter, the reader should firstly read the data which is presented separately at the two

[1] The study was a part of a series of analyses on comprehension of cyclic and trend events in different tasks in different graph designs (round vs. linear). For the full version of the study visit the http://blog.metu.edu.tr/ozge/my-publications/).

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 279–281, 2010.

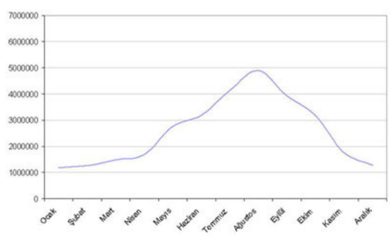
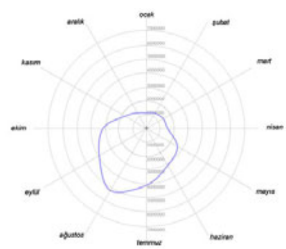
opposite sides of the x-axis before integrating the information represented by these points. Seasonal data is not captured directly in a linear graph. In order to allow for an immediate comprehension of cyclic concepts like seasons or times of day in a graph, these related data should be presented together. This feature is called “proximity” and is one of the Gestalt principles [4]. In order to investigate this phenomenon, a novel graph type was designed by taking the most relevant features of the Cartesian coordinate system (linearity of y-axis value) and the Polar coordinate system (the sphericity). Consequently, most related entities such as months which constitute the winter season are presented together in this graph.

## 2 Methodology

40 participants’ eye movements were collected by a Tobii 1750 Eye Tracker. Three types of simple graphs (bar, line and area) and two graph designs: linear and round graphs (between subjects variable) were used in this experiment (see Table 1). The sentence-graph verification paradigm, in which participants see the graph and sentence concurrently and are required to decide whether the sentence is an accurate description of the graph, was used. There are two different task type (Task-1 for winter and noon, Task-2 for summer and noon), which is also a within-subject variable. Sample sentences are given below:

Task-1: In Lake Eymir, the amount of zooplankton increases at night.
Task-2: In Lake Eymir, the amount of zooplankton decreases at noon.

**Table 1.** Sample line graphs for each graph design

| Linear Graph | Round Graph |
|---|---|
|  |  |

## 3 Results

The three consequent analyses were conducted. In first analysis, Task-1 and Task-2 sentences were evaluated. The first analysis investigates how the comprehension of temporal concepts is affected by their spatial location in the graph. The results showed that there were no significant main effects of graph design (F (1, 29) =1.369, and task (F (1, 29) =.042), nor was there a significant interaction between task and graph design (F (1, 29) =.996) in terms of fixation length. The insignificant interaction indicates that the linear graph design was not different from the round graph design. The results of fixation count parameter also indicated same results (for the statistical analysis visit http://blog.metu.edu.tr/ozge/diagram2010/), although the differences had been predicted in the linear group. For that reason, the gaze patterns of the participants in the linear group were investigated individually. This additional analysis examined whether the participant in the linear group actually looked at both sides of the graph when the task asks for edge information. The eye-tracking results were combined with the results from the concept evaluation form. As a result of this analysis, the subjects were divided into three categories. The results of the participants who did not look at both sides of the graph although in the questionnaire they reported months or hours from both sides of the timeline in the linear graph and the participants associated these concepts with the time units that are located on just one side of the graph were combined into the “one-sided linear group”. On the other hand, the results of the subjects who made two-sided reports and also looked at two sides were evaluated under the category of the “two-sided linear group”. After dividing the participants into three categories (one-sided linear, two-sided linear and round), the analysis of the task-1 was repeated. The analysis on fixation length showed that now there was a significant effect of graph group ($F(2, 34) = 10.092$). Post hoc comparisons test indicated that the one-sided linear group ($M = 5.962$) was significantly different than the mean score for the two sided-linear group ($M = 9.165$). Additionally, the mean score for the round group ($M = 7.237$) was significantly different from the two-sided linear group; however, the difference between the one-sided linear group and the round graph group was not significant. The results of fixation count parameter also indicated same interaction results. Furthermore, the number of the errors committed by the participants in the decision task indicates that there was a significant main effect of graph type. One-sided linear group ($M = 1.55$) made significantly more error in the judgment of the cyclic concepts than the two-sided linear group ($M = .75$) and the round group ($M = .55$).

## 4 Discussion and Conclusion

The overall results of all three analyses revealed that some of the linear graph readers just looked at one side of the graph although they had reported in the questionnaire that the relevant concept involved entities presented at both sides of the timeline. This result suggests that linear graphs representing cyclical events may either misguide the interpretation of the graph since the event that they represent is not coherent with the graph’s features or may lead to a truncated interpretation that only considers only partial evidence from one of the two sides of the graph. The re-analysis conducted after re-grouping the subjects showed that cyclic events are comprehended in less fixation time, and with fewer fixations in the round graph than in the two-sided linear group, while task performance is the same for the round graph group and the one-sided linear graph group that might have been misguided in the evaluation of cyclic concepts, though. The number of errors done by the participants also supports this conclusion. To sum up, all eye-tracking data results suggest that grasping trend information in cyclic events can be achieved with less effort in round graphs and the round design are computationally superior to the linear graph in the interpretation of cyclical concepts. This result is not trivial at all, given the fact that participants were not familiar with the round graph design.

For Reference Section, visit the http://blog.metu.edu.tr/ozge/my-publications/

# VCL, a Visual Language for Modelling Software Systems Formally

Nuno Amálio and Pierre Kelsen

University of Luxembourg, 6, r. Coudenhove-Kalergi, L-1359 Luxembourg
{nuno.amalio,pierre.kelsen}@uni.lu

**Abstract.** This paper overviews design of VCL, a new visual language for abstract specification of software systems at level of requirements. VCL is designed to be visual, formal and modular, and aims at expressing precisely structural and behavioural properties of software systems. Novelty of VCL design lies in its emphasis on modularity.

**Keywords:** formal modelling, visual languages, modularity.

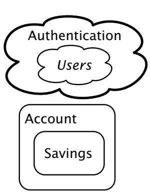
## 1 Introduction

Visual languages are widely used to describe software systems. Mainstream visual languages, like UML, however, have several shortcomings:

– They were mostly designed to be semi-formal (with a formal syntax, but no formal semantics). Although, there have been successful formalisations of semantics (e.g subsets of UML, see [1]), they are mostly used semi-formally. This brings numerous problems: it is difficult to be precise, unambiguous and consistent, and resulting models are not mechanically analysable.
– They cannot express a large number of properties diagrammatically; hence, UML is accompanied by the textual Object Constraint Language (OCL).
– They lack effective mechanisms to support coarser-grained modularity and separation of system-specific concerns.

To address these problems, this paper proposes the visual contract language (VCL) [2,3], designed to be formal and modular, and to target abstract specification at level of requirements. The paper presents an overview of VCL's design.

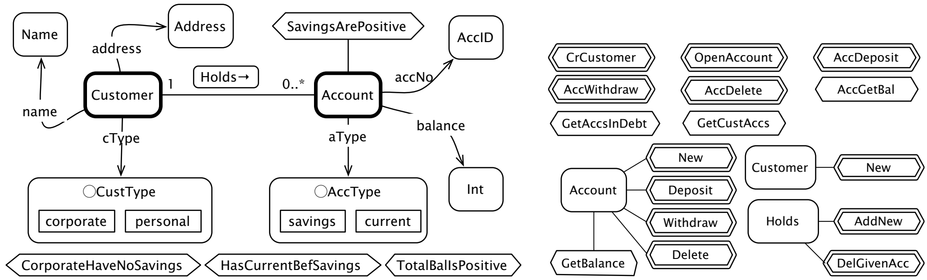
## 2 Visual Primitives




A VCL model is organised around *packages*, whose symbol is the *cloud*. Packages encapsulate structure and behaviour, and are built from existing ones using *extension*. They are VCL's coarse grained modularity construct. A package extends those packages that it encloses (e.g. *Authentication* to the left).

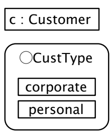
VCL *blobs* are labelled rounded contours denoting a *set*. They resemble Euler circles because topological notion of *enclosure* denotes subset relation (e.g. to the left, *Savings* is subset of *Account*).

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 282–284, 2010.

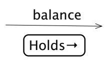


(a) Structural Diagram (b) Behavioural Diagram

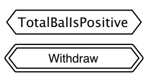
**Fig. 1.** Sample VCL structural and package diagrams (taken from [2])




*Objects*, represented as rectangles, denote an element of some set. Their label includes their name and may include set to which they belong (e.g. *c* to the left). Blobs may also enclose objects, and be defined in terms of things they enclose by preceding blob's label with symbol ◯. To the left, *CustType* is defined by enumerating its elements.




*Property edges* are represented as labelled directed arrows; they denote some property possessed by all elements of a set, like *attributes* in the object-oriented (OO) paradigm (e.g. *balance* to the left). *Relational edges*, represented as directed lines where direction is indicated by arrow symbol above line, denote some relation between blobs (*associations* in OO) – e.g. *Holds* to the left.

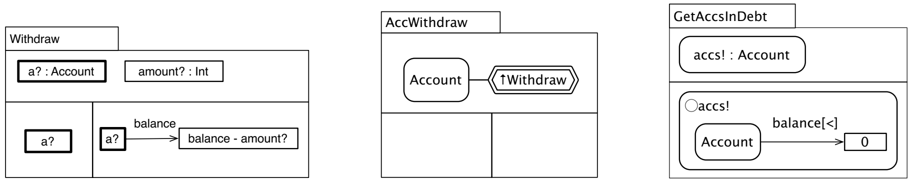



Represented as labelled hexagons, *constraints* denote some state constraint or observe operation (e.g *TotalBalIsPositive* to the left). They refer to a particular state of some structure or ensemble. *Contracts*, represented as labelled double-lined hexagons, denote operations that change state; hence, their representation as double-lined hexagons as opposed to single-lined constraints (e.g. *Withdraw* to the left).

## 3 VCL Diagrams

VCL diagrams are constructed using the visual primitives presented above. VCL *package diagrams* define packages. VCL *structural diagrams* (SDs) define structures of a package; the ensemble of structures makes the package's state space. Structures and their ensembles are subject to constraints (invariants), which are identified in SDs and defined in *constraint diagrams.* A sample SD is given in Fig. 1(a); see [2,3] for details.

*Behavioural diagrams* (BDs) identify all operations of a package. Operations are either local (operate upon individual structure), or global (scope of overall package, operate upon ensemble of structures). Update operations are represented in BDs as contracts, observe operations as constraints; these are defined in constraint and *contract* diagrams. A sample BD is given in Fig. 1(b); sample constraint and contract diagrams are given in Fig. 2; see [2] for details.




(a) Contract Diagram of *Account.Withdraw* (b) Contract Diagram of global *AccWithdraw* (c) Constraint Diagram of global *GetAccsInDebt*

**Fig. 2.** Sample VCL contract and constraint diagrams (taken from [2])

## 4 Semantics

VCL's design overviewed here is accompanied by a formal semantics. VCL takes a *generative* (or *translational*) approach to semantics. Currently, VCL diagrams are mapped into ZOO [4,1], a OO semantic domain expressed in formal language Z. We intend to support other formal languages in the future.

## 5 Using VCL

VCL has been applied to several case studies. Sample VCL diagrams of Figs. 1 and 2 are part of VCL model of *secure simple Bank* case study documented in [2]. In [5], VCL is used to model a large-scale case study.

## 6 Conclusions and Future Work

This paper overviews design of VCL, a visual language for formal abstract specification of software systems. A prominent feature of VCL is its support for modularity: constraints, contracts and packages are all modular constructs. Currently, we are completing formal definition of VCL, and developing VCL's tool[1].

## References

1. Amálio, N.: Generative frameworks for rigorous model-driven development. Ph.D. thesis, Dept. Computer Science, Univ. of York (2007)
2. Amálio, N., Kelsen, P., Ma, Q.: The visual contract language: abstract modelling of software systems visually, formally and modularly. Tech. Report TR-LASSY-10-03, Univ. of Luxembourg (2010), `http://bit.ly/9c5YwQ`
3. Amálio, N., Kelsen, P., Ma, Q.: Specifying structural properties and their constraints formally, visually and modularly using VCL. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Ukor, R. (eds.) EMMSAD 2010. LNBIP, vol. 50, pp. 261–273. Springer, Heidelberg (2010)
4. Amálio, N., Polack, F., Stepney, S.: An object-oriented structuring for Z based on views. In: Treharne, H., King, S., Henson, M.C., Schneider, S. (eds.) ZB 2005. LNCS, vol. 3455, pp. 262–278. Springer, Heidelberg (2005)
5. Amálio, N., Kelsen, P., Ma, Q., Glodt, C.: Using VCL as an aspect-oriented approach to requirements modelling. Transactions on Aspect Oriented Software Development 7 (to appear 2010)

[1] *Visual Contract Builder*, `http://vcl.gforge.uni.lu`

# Visualizing Student Game Design Project Similarities

Ashok Basawapatna and Alexander Repenning

University of Colorado Boulder
Dept. Of Computer Science
Boulder, CO. 80309-0430 USA
basawapa@colorado.edu, ralex@cs.colorado.edu

**Abstract.** High dimensional cosine calculation is a tool that is often used to discover the similarity between two vectors in semantic space. This research uses vector similarities to create a novel way of visually representing the submitted work of a whole classroom of students over the course of a semester. Using a high dimensional cosine calculation, every student assignment submission is compared to one another in the Educational Game Design Class, an undergraduate/graduate programming class taught in Spring 2009 at the University of Colorado Boulder. This is accomplished by first converting every student submission into a representative vector based on submission project code. Through creating a visualization of these similarity scores, called a 'Similarity Matrix', interesting patterns begin to emerge indicating notable phenomena such as class 'watershed moments' and relative in-class effectiveness of presented programming concepts.

## 1 The Similarity Matrix

In spring semester 2009, the University of Colorado at Boulder offered the Educational Game Design Class. The Educational Game Design Class teaches students the theory behind making engaging and educational "gamelets" (mini games) through using AgentSheets, an agent-based visual programming environment [1]. The first four assignments are the same for every student; they include Frogger, Sokoban, Centipede, and a 'Sims' type simulation. The next four open-ended assignments are called 'gamelet madness' wherein students create a new gamelet, of their choosing, every week. The remainder of the class is spent on creating a final project and play testing this project with middle school students.

Previous research talks in-depth about the Educational Game Design Class structure including the use of a cyberlearning infrastructure, called the Scalable Game Design Arcade, that enables open classroom interactions among students [2]. At semester's end, the Educational Game Design Class had over 300 unique games submitted. To better understand and gain insights into the class necessitates a meaningful way to represent this enormous amount of data such that it:

- Is easy to see student game progression over the course of the semester.
- Enables the ability to better understand the types of games students choose to make during the open-ended part of the class.
- Gives insight into the assignments themselves and their relation, similarity or dissimilarity, to other assignments in the class.

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 285–287, 2010.

To this end we developed a ‘Similarity Matrix’, depicted in Figure 1a and Figure 1b, which uses the high dimensional cosine calculation to compare every student’s game submission to every other student submission [3].

AgentSheets programs consist of ‘agents’ which are the game characters. Each agent has a depiction and behaviors, which dictate how the agent acts in a given situation. Behaviors are accomplished by “If/Then” conditionality statements. AgentSheets allows you to use 16 different “If” conditions and 23 different “Then” actions. It is possible to represent every game created by a numeric vector of length 39, wherein each vector element represents how many of each individual conditions and actions are used to implement a given game. Using these vectors, each student game can be compared to every other student’s game for similarity using the high dimensional cosine formula; every vertical and horizontal line in Figure 1a depicts one student’s game compared to every other student’s game. The whiter squares represent more similar games and the darker squares represent a greater difference between two games. The diagonal white line in Figure 1a is every game compared to itself (identical similarity, similarity score of 1), and the diagram is symmetric about the diagonal. Figure 1b is the average of similarity scores within every game comparison block.

The following are a few notable initial results of this diagram analysis; each result is in bold followed by a brief discussion.

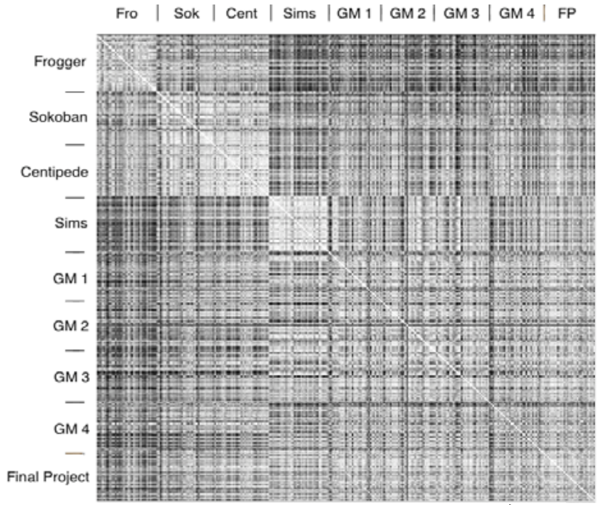
- **Frogger is the most dissimilar to the rest of the games**

In Figure 1a and 1b we see that Frogger is always one of the most dissimilar games to every subsequent game, especially the student created gamelet madness games and Final Project. Frogger is the first assignment students work on in the class. One reason for the dissimilarity between Frogger and every other game is that students often use implementations in Frogger of various programming patterns that they would not use once they figured out correct or simpler implementations.

- **Sims project is a ‘watershed moment’ wherein the projects before it are dissimilar and the projects after are more similar.**

The Sims assignment implements ‘agent-diffusion’ and ‘hill climbing’ where an agent automatically follows the diffused “scent” of another agent. The diffusion and hill climbing patterns are the first artificial intelligence based-patterns students learn and are different than any pattern learned before [1]. Looking at Figure 1a and 1b, we see that, as compared to preceding projects, Sims is very dissimilar. Furthermore, the Sims game is the most self similar to itself, and when students program their own games in gamelet madness 1-4 and the Final project, they are markedly more similar to the Sims project than projects that precede the Sims assignment. Therefore, we can view Sims, in the context of the Educational Game Design class, as a ‘watershed moment’: once the hill climbing and diffusion patterns are taught to students, it is not only different from anything learned before, but students actively choose to implement these sophisticated patterns when it comes to designing their own games.

This visualization is a good initial attempt at analyzing the enormous amount of student-created data in the Educational Game Design Class. Further visualization research should look at visualizing individual agent-based similarity between two projects and figuring out a more detailed representational vector for each game.




| | Frogger | Sokoban | Centipede | Sims | GM 1 | GM 2 | GM 3 | GM 4 | Final Project |
|---|---|---|---|---|---|---|---|---|---|
| **Frogger** | .71 | .59 | .62 | .41 | .47 | .46 | .44 | .47 | .49 |
| **Sokoban** | .59 | .71 | .70 | .53 | .57 | .57 | .55 | .56 | .60 |
| **Centipede** | .62 | .70 | .80 | .54 | .63 | .60 | .59 | .62 | .64 |
| **Sims** | .41 | .53 | .54 | .82 | .62 | .63 | .66 | .57 | .66 |
| **GM 1** | .47 | .57 | .63 | .62 | .64 | .64 | .65 | .61 | .67 |
| **GM 2** | .46 | .57 | .60 | .63 | .64 | .65 | .66 | .61 | .68 |
| **GM 3** | .44 | .55 | .59 | .66 | .65 | .66 | .68 | .61 | .70 |
| **GM 4** | .47 | .56 | .62 | .57 | .61 | .61 | .61 | .60 | .64 |
| **Final Project** | .49 | .60 | .64 | .66 | .67 | .68 | .70 | .64 | .72 |

**Fig. 1.** White represents similarity. **A)** Top figure: each vertical and horizontal line is one student's game compared to every other student's game in the Educational Game Design Class. **B)** Bottom figure is the same as the Top but with each game-block similarity score averaged.

## References


1. Lewis, C., Repenning, A.: Creating Educational Gamelets. In: DiGiano, C., Goldman, S., Chorost, M. (eds.) Educating Learning Technology Designers: Guiding and Inspiring Creators of Innovative Educational Tools, pp. 203–229. Routledge, New York (2008)
2. Repenning, A., Basawapatna, A., Koh, K.H.: Making University Education More Like Middle School Computer Club: Facilitating the Flow of Inspiration. In: WCCCE 2009, Burnaby, Canada. ACM Press, New York (2009)
3. Lukashenko, R., Graudina, V., Grundspenkis, J.: Computer-Based Plagiarism Detection Methods and Tools: An Overview. In: CompSysTech 2007, Rousse, Bulgaria, June 14-15, pp. IIIA.18-1–IIIA.18-6 (2007)

# Are Pixel Graphs Are Better at Representing Information than Pie Graphs?

Jolie Bell and Jim Davies

Institute of Cognitive Science, Carleton University
1125 Colonel By Drive, Ottawa, ON, K1S 5B6 Canada
jbell8@connect.carleton.ca, jim@jimdavies.org

**Abstract.** This study investigates whether pixel graphs more accurately represent percentage based data than pie graphs or bar graphs. Participants were asked to answer 24 questions each pertaining to a different graphical stimulus. Results suggest that pixel graphs are significantly better at representing large quantities of percentage based data than pie graphs.

**Keywords:** Information Representation, Graph, Data.

## 1 Introduction

A pixel is a small block representing one percent out of one hundred percent. Pixel graphs are color-coded to enable legibility. In order to use a pixel graph to represent data (e.g., 30% of Canadian children prefer chocolate milk to hot chocolate during the winter months) each pixel must be given a specific color to reflect the data (e.g., 30 blue pixels, representing the chocolate milk preference and 70 red pixels representing the hot chocolate preference).

There is a recent trend in representing data to laypeople in the form of pixel graphs (1). Data collected by private blogs and advertising companies hope that the novel appearance of the pixel graph will capture consumer attention. Companies, such as Hewlett Packard, are using pixel graphs to represent percentage – based information instead of the more traditional pie graphs (1).

To date, pixel graphs are most commonly used to represent data in geographic maps displaying information about depth and temperature and in medical and neural-imaging studies. While it is clear that pixel graphs have a place in representing information to scientists, are pixel graphs better than pie graphs when displaying percentage - based information to laypeople?

One major concern is that the appeal of pixel graphs may be purely aesthetic and there ability to attract consumers may simply where off once the novelty of the pixel graph ends (1). Companies representing sales information to consumers using the unfamiliar pixel graphs instead of the traditional pie graphs may gain an advantage in the ability to deceive consumers by purposefully designing the pixel graph to be misleading (3).

Pie graphs have a long history of being designed to intentionally manipulate viewers by being vague, confusing or misleading (6). Most laypeople have a high

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 288–291, 2010.

level of familiarity with pie graphs and are capable of detecting misleading information and distinguishing between poorly designed pie graphs and well designed pie graphs (6). A final concern using pixel graphs to represent information to laypeople is that due to the unfamiliarity of the pixel graph, laypeople will not be able to read the graph. Unlike pie graph, there is no standard for the pixel graph; information is displayed in whatever shape, format or color the designers choose. This untraditional aspect makes the pixel graph very attractive in advertising, difficult to read for the novice layperson (6, 4).

Researchers detailing the visual mining of E-customer behavior at Hewlett Packard discovered that they needed a better way to present their data. The researchers wanted a graphical format that would enable them to combine all of their data, from several different types of graphs into one graph for a general comparison. The data from their study contained hundreds of data points that needed to be represented in an easy, quick read format. After trying many different graphical formats, the researchers found that pixel graphs could easily accommodate their requirements (2).

2 found that in order to display enormous datasets (e.g., over 100 points of data on a single graph) with accuracy in an easy quick read format they needed a graph that did not focus the reader's attention on numerical values (e.g., the range of a particular variable that is represented by an axis – degrees of temperature, speed in seconds, weight in pounds) but on the visual relationship between the different variables represented on the graph. The researchers found that the color-coded format of the pixel graph made it possible for them to represent the relationships between the data points in an easy quick read format (2).

Using pixel graphs to display large quantities of information is also common in neuroimaging. In a study of blood brain barrier disruption, neuroscientists utilized a new computer program that graphically represented their findings in pixel graphs. As part of their analysis, the researchers rated the new computer graphing system. The researchers found that by using pixel graphs, they were able to see an immediate and instantaneous comparison between within-participants measurements of blood brain barrier disruptions, and between-participants measurements as each participant was evaluated (5).

Literature suggests that unlike pie graphs, pixel graphs are capable of displaying an enormous amount of information in compact space, with clear and accurate labeling, the arbitrary nature of the shape and color-coding segments in pixel graphs are not confusing. Are pixel graphs more useful than pie graphs for today's on-line world of commerce? This study investigates the potential of pixel graphs as a useful way to represent information to laypeople. The bar graph is the most accurate graphical display of for representing percentage-based data; pie graph is the most popular way to represent percentage - based data (6). We investigate the accuracy of bar graphs compared to the pie and pixel graphs in this study.

One problem in comparing pie, bar and pixel graphs is the level of metric detail in the presentation of each graph. Metric detail is the level of numeric explanation (e.g. written percentages labeling each variable on the $x$ axis and a numeric scale assisting in comparisons between variables along the $y$ axis) and the amount of visual assistance given to discriminate one variable from another in the graph. The variables represented in pie and pixel graphs are displayed as one whole piece of visual information, distinguished from one another by color. In order to control differences in metric detail, the three types of graphs are presented in metric and non – metric condition.

# 2 Method

## 2.1 Participants

The participants in this experiment were 40 undergraduate college students enrolled in psychology courses at Carleton University. Each student received .05% extra credit toward their final grade for participation in this study. Twenty-five participants were female and 15 participants were male.

## 2.2 Design

Four independent variables, type of graph (pie, bar, pixel), amount of information represented, small (4 points of data) or large (12 points of data), level of question difficulty, easy or hard, and the level of metric information available on the graph (e.g. metric or non – metric).

## 2.3 Stimuli

Twenty - four trials of a graph representing either a small or large data set (decided based on the number of distinguishable colors available in the graphical design database). The Y axis of the metric graphs displayed ascending percent from 0 to 100, the X axis displayed the conditions being measured (e.g. favorite television shows among undergraduates, number of undergraduates is each major). One question was shown with each of the 24 trials. There were 12 easy questions and 12 hard questions in the experiment. All of the stimuli where created using Gimp photo manipulation software.

## 2.4 Procedure

In this 3x2x2 between subject experiment, participants completed 24 timed trials in random order, answering the 3 way multiple choice question displayed below the graph in each trial.

# 3 Results

All statistics in this study use an alpha level of .05. Particiants were significantly more accurate reading bar graphs than reading pie $F(13.67, 574.02) = 12.31$ or pixel graphs, $F(12.17, 511.81) = 11.47$. Pixel graphs are more accurate than pie graphs $F(13.16, 554.20) = 13.25$.

Pixel graphs are more accurate than pie graphs in the large data set condition $F(13.20, 554.20) = 13.25$, but pie graphs are more accurate than pixel graphs in the small data set condition, $F(6.50, 273.10) = 13.99$.

Metric pie graphs are significantly more accurate than the metric pixel graphs $F(2.98, 125.26) = 15.37$; Non metric pixel graphs are more accurate then the non metric pie graphs $F(6.77, 284.30) = 9.99$.

## 4 Discussion

This experiment confirmed that bar graphs are the most accurate graphical format for presenting percentage data (6). The most important finding of this experiment: pixel graphs are significantly more accurate in representing percentage-based information than pie graphs. However, the fact that there were no significant interactions between size of data and type of graph or the level of metric description suggests that the significance of the repeated measures ANOVAs are possibly the result of the questions being asked and not the accuracy of the graphs. Another possible reason for the lack of any significant interaction is that the methodology of the experiment combined too many independent variables: a series of experiments with the independent variables spread out over more trials is the next step.

It is important to consider that pixel graphs vary in quality: some may be easier to read than others. In the next study, pixel (pie & bar) graphs with varying design will be rated against one another (using personal preference, accuracy and reaction time to questions) by laypeople.

## References

1. Dash, A.: Are Pixels the New Pies?, http://www.dashes.com/anil/2007/07/pixels-are-the-new-pies.html (retrieved on December 16, 2008)
2. Hao, M., Ladisch, J., Dayal, U., Hsu, M., Krug, A.: Hewlett Packard Research Laboratories, Palo Alto, CA, http://www.hpl.hp.com/techreports/2001/HPL-2001-144.pdf (retrieved on December 16, 2008)
3. Healy, K.: Survey Article: Digital Technology and Cultural Goods. The Journal of Political Philosophy 10, 478–500
4. Simanek, D.: Psychophysics (2008), http://www.lhup.edu/~dsimanek/physics.htm#physdocs (retrieved on December 16, 2008)
5. Tomkins, O., Shelef, L., Kaizerman, L., Friedman, A.: Abstracts of the 16th annual meeting of the Israel society for neuroscience. Neural Plasticity 10, 168 (2007)
6. E. R. The Visual Display of Quantative Information. Graphics Press Llc, Cheshire, Connecticut (2000)

# Thinking with Words and Sketches – Analyzing Multi-modal Design Transcripts Along Verbal and Diagrammatic Data*

Martin Brösamle and Christoph Hölscher

University of Freiburg, Centre for Cognitive Science, Friedrichstr, 50, 79098 Freiburg, Germany
{martinb,hoelsch}@cognition.uni-freiburg.de

## 1 Introduction

Originally, our main research goal was to investigate human wayfinding in complex architectural spaces such as conference centres, airport terminals and the like. (Hölscher et al., 2006) At the same time, the profession of architecture, which largely forms these environments, came into play. Sketches and Diagrams play a central role in the process of architectural design – to assist thinking and as a device of communication. (Laseau, 1989) Furthermore, Goel (1995) argues that certain characteristics of sketches perfectly match the requirements in early phases of design problem solving. In an interview study with experienced architectural designers, our informers frequently produced sketches (Brösamle & Hölscher, 2008), and the verbal transcripts of these interview could only be understood with reference to these sketches. Unlike the transcript-centred approach in the earlier studies, we presently employ a diagram-based way of querying the multi-modal corpus. This exposes the diagrammatic characteristics of the design activity to the research process, and grounds the often idiosyncratic verbal comments in an architecture-typical medium, namely plans and sketches.

## 2 Methods

Participants were asked to place a waiting area in the main hall of a complex university building while taking care of navigability issues. In order to capture the spatial characteristics inherent in such a design task, references to elements in plans and sketches were video-recorded, such that all drawing and gesture activity could be annotated in the plan, and in the verbal transcripts. A diagram-based query mechanism was developed: For a chosen seed set of diagrammatic elements, all transcript passages are identified, in which a reference to a selected

* This research was funded by the German Research Foundation (DFG), SFB/Tr 8, Spatial Cognition. We would like to thank our interview partners for sharing their expertise with us. Many thanks to Julia Asbrand, Inka Hähnlein, Tabea Koll, David Kühner, and Antje Roniger for great support in data preparation and analysis.

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 292–294, 2010.

diagrammatic element occurs. In a second step, the identified verbal contexts are attached to the respective gesture or pencil stroke and integrated into the diagrammatic representation. There are two ways of using the diagrammatic query mechanism: (1) across participants, verbal contexts are retrieved for each diagrammatic element of a certain type or in a certain area. (2) Within one participant, contexts are retrieved and additional links are inserted to emphasize co-located references to other diagrammatic elements. As a result, all diagrammatic elements are linked together via the transcript episode referring to them.

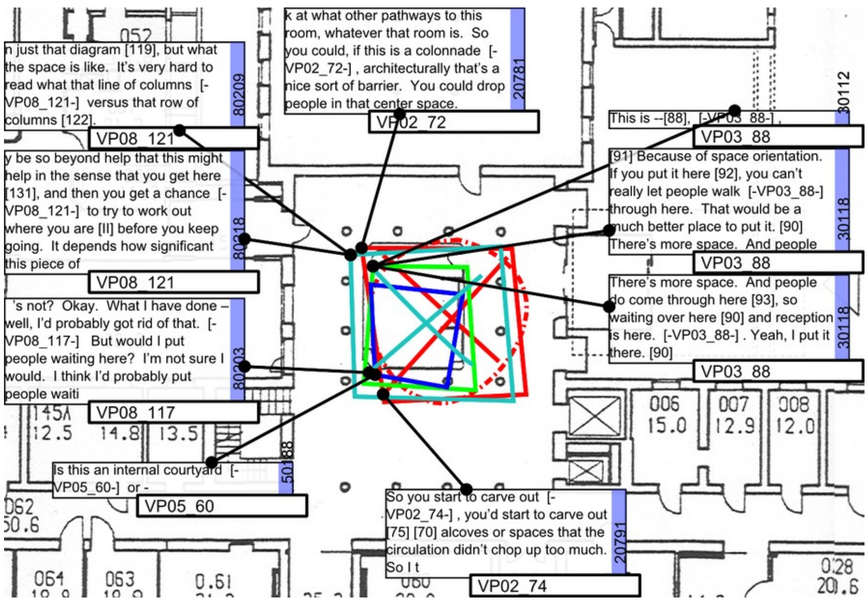



**Fig. 1.** Across participants query for verbal contexts referring to the architectural structure in the middle of the main hall, the "Central Block". Verbal material is placed in *context boxes* which indicates the *paragraph number* in the blue stripe right to the text. The element underlying the retrieval of that context is marked by its number enclosed by *[-* and *-]*. Paragraph numbers are five digit decimal numbers indicating the informer number by the first digit. Within each interview paragraph numbers are in ascending but not necessarily consecutive order. A continuous line connects each context box with the diagrammatic element, which underlies the retrival of that particular verbal context. Notations of the form *VPxx_yy* indicate participant *xx* and diagrammatic element *yy* underlying the query of that context box.

## 3 Results and Discussion

We employed the first method to compare spatial elements with different centrality regarding the waiting area placement task. The retrieved verbal material demonstrates the sensitivity of the method towards the properties of the seed set: topics and predominant aspects clearly change proceeding from the most central seeds to more peripheral ones. Secondly, the method allows for extracting key concepts in relation to concrete spatial structures represented in the plans and sketches. Figure 1 shows the verbal contexts of one "central block", which was located in a main hall, where participants were supposed to place the waiting area. The diagram illustrates, how the spatial query extracts phrases from the verbal transcripts, which – in a way – characterize the role of this architectural structure in relation to the design task. ("you can't really let people walk through here" [Par. 30118]; "architecturally, this is a nice sort of barrier" [Par. 20781]).

The second method identifies key areas in the diagrammatic material as well as key episodes in the transcript: The number of outgoing connections from a verbal context gives an indication of how many different architectural elements connected in the argument; the covered area reveals how global or local it is. Key locations, by contrast, are spatial areas with many connections to different text passages referring to them. Taken together, these novel approaches help detect connections and commonalities otherwise overlooked in verbal data. Future work will cross-validate this diagram-based method with other formal methods of design protocol analysis, like linkography. (Goldschmidt, 1990)

The results presented before are products of a first application of the presented method. Although preliminary, they underpin the scientific potential for investigating multi-aspectual domains such as architecture. (Bertel, Vrachliotis, & Freksa, 2006) The instrument allows for actually handling complexity instead of reducing it to the low-dimensionality of laboratory measurements.

## References

Bertel, S., Vrachliotis, G., Freksa, C.: Aspect-oriented building design: Towards computer-aided approaches to solving spatial constraints in architecture. Aspect-oriented building design: Towards computer-aided approaches to solving spatial constraints in architecture. In: Allen, G. (ed.) Applied Spatial Cognition: From Research to Cognitive Technology, Lawrence Erlbaum Associates, Mahwah (2006)

Brösamle, M., Hölscher, C.: The architects' understanding of human navigation. In: Haq, S., Hölscher, C., Torgrude, S. (eds.) Movement and Orientation in Built Environments: Evaluating Design Rationale and User Cognition, Bremen/Freiburg, Germany (2008)

Goel, V.: Sketches of thought. Massechusetts Institute of Technology (1995)

Goldschmidt, G.: Linkography: assessing design productivity. In: Trappl, R. (ed.) Cybernetics and Systems 1990 (1990)

Hölscher, C., Meilinger, T., Vrachliotis, G., Brösamle, M., Knauff, M.: Up the down staircase: Wayfinding strategies in multi-level buildings. Journal of Environmental Psychology 26, 284–299 (2006)

Laseau, P.: Graphic thinking for architects and designers, 2nd edn. Van Nostrand Reinhold, New York (1989)

# How Diagram Interaction Supports Learning: Evidence from Think Alouds during Intelligent Tutoring

Kirsten R. Butcher

University of Utah, Department of Educational Psychology, 1705 E Campus Center Drive, MBH 327, Salt Lake City, UT, 84112, USA
kirsten.butcher@utah.edu

**Abstract.** The goal of this research was to explore the impact of diagram interaction on students' cognitive processes during learning. Using a think-aloud protocol, students' processes were studied as they practiced problem solving with an intelligent tutoring system containing diagrams that were either static or interactive. Diagram interaction resulted in better transfer performance, prompted deeper cognitive processing, and reduced students' reliance on shallow problem-solving processes.

**Keywords:** Diagrams, Intelligent Tutoring, Think-Aloud Protocols.

## 1 Introduction

Visual representations have long been recognized to support learning, especially when relevant diagrams accompany text materials [e.g., 1]. Recently, research has begun to explore the potential benefits of interactive visual materials. Butcher and Aleven [2, 3] previously have demonstrated that diagram interaction can support student learning, even in an intelligent tutoring system that provides multiple supports for student learning by doing. These studies varied the location of student interactions (i.e., geometry diagrams vs. solution tables) during problem solving with the Geometry Cognitive Tutor. Results demonstrated that students who interacted with diagrams exhibited more robust learning of geometry principles, as evidenced by transfer task performance [2] and long-term retention of problem-solving skills [3].

One explanation of the benefits of diagram interaction may be that interaction creates an integrated visual-verbal representation that reduces a learner's extraneous cognitive load [4]. However, another explanation may be that diagram interactions focus learners' attention on concrete visual representations that students can process in meaningful ways. Even static diagrams have been found to increase student generation of self-explanations during text learning [5]. Interactive diagrams may be even more effective in directing attention to visual elements, making it more likely that students will engage in deep, active learning processes. To explore the effects of diagram interaction on students' cognitive processes during learning with an intelligent tutoring system, a think-aloud study was conducted.

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 295–297, 2010.

## 2 Think-Aloud Study

**Participants.** Ten high-school students (*M* age = 14) were recruited as participants via a newspaper advertisement. Due to a (parent-reported) learning disability that made thinking-aloud difficult, one student was dropped from study analyses.

**Materials and Procedure.** Participants were randomly assigned to one of the two intelligent tutoring conditions from Butcher and Aleven [2]: 1) an *interactive diagram* condition, where students clicked directly on individual elements of geometry diagrams to enter answers and receive feedback; or 2) a *static diagram* condition, where students saw static (non-interactive) geometry diagrams during problem solving but clicked in a separate solution table to enter answers and receive feedback. Assessment materials included three types of items: *practiced*, *transfer*, and *diagram reasoning*. Practiced items were geometry problems of similar format and difficulty to those solved during tutoring. Transfer items required students to troubleshoot unsolvable problems (a situation *not* encountered during tutor practice); students needed to identify a geometry theorem and the diagram elements that would be needed to solve for goal angles. Diagram reasoning items asked students to use geometry theorems to explain how a given angle could be used to find other angles.

Following consent/assent procedures, students completed a demographic questionnaire and pretest. They then thought aloud as they worked with their assigned condition of the tutoring system for one hour. Finally, students completed the posttest.

A research assistant coded all verbal data using cognitive process categories adapted from previous research [5, 6]. A second rater coded 20% of the data, showing good interrater reliability ($K_w$ = .73). Deep and shallow process categories were formed by grouping relevant processes, as seen in Table 1. Deep processes reflected conceptual reasoning and goal-oriented planning; shallow processes reflected thinking that was irrelevant to or isolated from domain concepts, or repeated provided content.

**Table 1.** Component cognitive processes and think-aloud examples for process categories

| *Category* | *Component Processes* | *Example Statement* |
|---|---|---|
| Deep | Self-Explanation | "[It is] same side interior because CB is the transversal of AB and DC which are parallel." |
| | Planning | "Actually, I think I'm going to try to solve ALY." |
| Shallow | Mathematic Operations | "So, I need to do 180-58." |
| | Reading/Paraphrasing | "They only give me… OY is parallel to TB." |
| | Shallow Reasoning | "Well, it's been 180 minus something for the last three problems." |

**Results.** A repeated measures MANOVA demonstrated that, compared to static diagrams, interactive diagrams led to higher scores on transfer items ($F_{(1,7)}$ = 9.1, $p$ = .02). Observed cognitive processes were analyzed using a MANCOVA, in which the total number of think-aloud utterances and pretest performance were used as covariates. Compared to students using static diagrams, students using interactive diagrams were more likely to engage in deep processes ($F_{(1,5)}$ = 8.0, $p$ = .04) and less likely to engage in shallow processes ($F_{(1,5)} = 7.6$, $p = .04$). Overall, learning outcomes were positively correlated with deep processes and negatively with shallow processes (see Table 2).

**Table 2.** Correlations between observed processes and learning outcomes

| *Process Type* | *Practiced Items* | *Transfer Items* | *Diagram Items* |
|---|---|---|---|
| Deep | .80** | .57† | .72* |
| Shallow | -.64† | -.63† | -.65† |

†p ≤ .10; *p ≤ .05; **p ≤ .01.

## 3 Discussion

Results demonstrated that interactive diagrams supported students' transfer performance and promoted the use of deep cognitive processes – such as self-explanation and planning – during learning. Interactive diagrams also reduced students' use of shallow processes. These results support the possibility that diagram interaction supports learning by encouraging deep, active processing of learning materials, even when students are working with a highly-interactive and successful intelligent tutoring system. Follow-up work is exploring the impact of interactive elements vs. attentional cues in promoting student learning during intelligent tutoring.

**Acknowledgments.** The author thanks Vincent Aleven, Octav Popescu, Michael Nugent, Thomas Bolster, and Nicki Turnidge-Halvorson for their contributions to this work. This project was supported in part by the Pittsburgh Science of Learning Center, through funding from the National Science Foundation (SBE-0836012).

## References

1. Dwyer, F.M.: The effectiveness of visual illustrations used to complement programmed instruction. Journal of Psychology 70, 157–162 (1968)
2. Butcher, K.R., Aleven, V.: Integrating visual and verbal knowledge during classroom learning with computer tutors. In: McNamara, D.S., Trafton, J.G. (eds.) Proceedings of the 29th Annual Cognitive Science Society, pp. 137–142. Cognitive Science Society, Austin (2007)
3. Butcher, K.R., Aleven, V.: Diagram interaction during intelligent tutoring in geometry: Support for knowledge retention and deep understanding. In: Love, B.C., McRae, K., Sloutsky, V.M. (eds.) Proceedings of the 30th Annual Conference of the Cognitive Science Society, pp. 1736–1741. Cognitive Science Society, Austin (2008)
4. Tarmizi, R.A., Sweller, J.: Guidance during mathematical problem solving. Journal of Educational Psychology 80, 424–436 (1988)
5. Butcher, K.R.: Learning From Text with Diagrams: Promoting Mental Model Development and Inference Generation. Journal of Educational Psychology 98, 182–197 (2006)
6. Schworm, S., Renkl, A.: Computer-supported example-based learning: When instructional explanations reduce self-explanations. Computers & Education 46, 426–445 (2006)

# Creating a Second Order Diagrammatic Logic

Peter Chapman and Gem Stapleton

Visual Modelling Group, University of Brighton, UK
{p.b.chapman,g.e.stapleton}@brighton.ac.uk

**Abstract.** Many of the formal diagrammatic logics that have been developed are limited to be first order (typically monadic). This means that such logics cannot define commonly occurring concepts and, thus, are not as widely applicable as we might like. Suitably increasing their expressiveness will allow both the formalization of second order concepts and the study of such concepts from a new perspective. Our aim is to produce a second order diagrammatic logic and we present the initial ideas towards the development of such a logic.

**Introduction.** Since the seminal work of Shin [5], methods for the design and development of diagrammatic logics have become better understood. The range of diagrammatic logics available is increasing, but those which are formalized are typically limited in expressiveness to at most that of first order logic. Examples include existential graphs [1], Euler diagrams [6], Venn-II, Euler/Venn diagrams [7], spider diagrams [3], and constraint diagrams [4]. In terms of expressiveness, Venn-II and Euler diagrams (as in [6]) are equivalent to Monadic First Order Logic (MFOL), spider diagrams are equivalent to MFOL with equality, and Euler/Venn diagrams have a level of expressiveness that is between Venn-II and spider diagrams. Recent work has considered spider diagrams and their expressiveness with respect to star-free regular languages, which led to the development of spider diagrams of order [2].

The limitation to first order precludes the formalization of many commonly occurring concepts in both mathematics and computer science, such as defining the language $(aa)^+$, or the property of being finite. In this paper, we propose second order spider diagrams of order, incorporating aspects of the (first order) constraint diagram notation [4] and the ability to quantify over sets, for instance.

**Second Order Spider Diagrams.** Here we demonstrate how to define second order concepts using an extension of spider diagrams of order. First, we note that the semantics of spider diagrams of order are model based: an *interpretation* consists of a universal set, $U$, a strict total order, $<$, on $U$ and a mapping from the curve labels to subsets of $U$. In addition, we include further syntax whose interpretation is determined by $<$, such as: two constants min and max, and a function symbol S, where min is interpreted as the least element of $<$, max is the greatest element and S is a successor function that respects $<$, following Thomas who studied the definability of regular languages using logic [8]. When extending spider diagrams to include second order facilities, this notion of an interpretation is sufficient.

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 298–300, 2010.

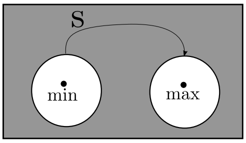



**Fig. 1.** $U$ is even

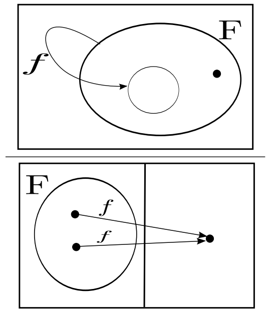



**Fig. 2.** $F$ is finite

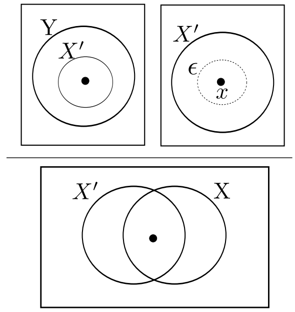



**Fig. 3.** $X$ is dense in $Y$

To take the level of expressiveness to second-order, we need to quantify over subsets of $U$, for instance. We make use of unlabelled curves to assert the existence of a subset and we introduce the notion of a free variable, which essentially acts as a universal quantifier. Free variables are represented using labels in italics, whereas syntax with a fixed interpretation has non-italicized labels. Thus, a curve with an italicized label essentially acts as universal quantification over sets. Unlabelled syntax is existentially quantified. We now present a series of examples to demonstrate the use of this new syntax.

**Example 1.** In terms of regular languages, $(aa)^+$ is not star free and, therefore, is not first order definable. Using logic, we can define $(aa)^+$ by asserting that $U$ has even cardinality, given a single letter alphabet. A second order spider diagram making this assertion can be seen in figure 1. Here, the two unlabelled curves assert the existence of two sets, $X_1$ and $X_2$, that partition $U$ (owing to their disjoint interiors, the shading, and the rectangle representing $U$). The presence of min in the lefthand set expresses that $\min \in X_1$ and, similarly, $\max \in X_2$. The use of the arrow tells us that all the successors of elements in $X_1$ are precisely the elements of $X_2$. This follows the interpretation of arrows in constraint diagrams, where the source of the arrow gives a domain restriction on S and the target represents the image of S under this restriction. The diagram as whole expresses

$$\exists X_1 \exists X_2 \big(X_1 \cap X_2 = \emptyset \wedge X_1 \cup X_2 = U \wedge \min \in X_1 \wedge \max \in X_2 \wedge im(\mathrm{S}|_{X_1}) = X_2\big).$$

From this it follows that $|U|$ is even, since every element, except max, has a successor. If we had instead located max in $X_1$ then $U$ would have odd cardinality.

**Example 2.** The diagram in figure 2 expresses that F is finite, which again is a second order concept. Informally, it states that any non-surjective function, $f\colon \mathrm{F} \to \mathrm{F}$, is not injective. The italicized label, $f$, represents a free second order function variable, whose image, $im(f)$, is a proper subset of F. The line between the diagrams tells us that if the statement made by the diagram above the line is true then the statement made by the diagram below the line is true; thus, we have represented implication and the presentation is of the style of natural deduction. The bottom diagram consists of two boxes, each one representing the universal set. We have two distinct elements, $x_1$ and $x_2$, in $F$ that both map to the same element, $y$, under $f$. This element, $y$, may or may not be equal to one of $x_1$ and $x_2$; since it is in a different box we have not made any assertion about distinctness. The diagram as a whole expresses

$$im(f) \subset F \Rightarrow \exists x_1 \exists x_2 (x_1 \neq x_2 \wedge x_1, x_2 \in F \wedge f(x_1) = f(x_1)).$$

which it equivalent to

$$\forall f \Big( im(f) \subset F \Rightarrow \exists x_1 \exists x_2 (x_1 \neq x_2 \wedge x_1, x_2 \in F \wedge f(x_1) = f(x_1)) \Big).$$

To justify that this captures the property of being finite, suppose that F is infinite. Then there would be an injective function from F to a proper subset, contrary to the diagram's assertion. Hence $F$ cannot be infinite.

**Example 3.** A second order concept that occurs in topology is that of a set being *dense*: X is **dense** in Y if and only if for all $X' \subseteq$ Y where $X'$ is non-empty and open, there exists $x \in X' \cap$X. We can represent this diagrammatically as shown in figure 3. As before, Y and X have fixed interpretations as subsets of $U$, since they are not in italics, but $X'$ is free, and ranges over all subsets of Y. Here we have included new syntax, $\epsilon$ and the dashed curve, to assert that the $\epsilon$ neighbourhood of the element represented by the free variable $x$ is a subset of $X'$. Such notation is useful for applications in an analysis setting.

**Summary.** We have begun the development of a second order diagrammatic notation that is more expressive than the (first order) diagrammatic logics defined to date. Our next step is to more fully explore the design, paying particular attention to how we might define other second order concepts. We want to to ensure that the notation is well-matched to semantics and includes free-rides where possible.

**Acknowledgements.** This research is supported by EPSRC grants EP/ H012311/1 and EP/H048480/1.

## References

1. Dau, F., Ekland, P.: A diagrammatic reasoning system for the description logic $\mathcal{ACL}$. Journal of Visual Languages and Computing 19(5), 539–573 (2008)
2. Delaney, A., Taylor, J., Thompson, S.: Spider diagrams of order and a hierarchy of star-free regular languages. In: Stapleton, G., Howse, J., Lee, J. (eds.) Diagrams 2008. LNCS (LNAI), vol. 5223, pp. 172–187. Springer, Heidelberg (2008)
3. Howse, J., Stapleton, G., Taylor, J.: Spider diagrams. LMS Journal of Computation and Mathematics 8, 145–194 (2005)
4. Kent, S.: Constraint diagrams: Visualizing invariants in object oriented modelling. In: Proceedings of OOPSLA 1997, pp. 327–341. ACM Press, New York (1997)
5. Shin, S.-J.: The Logical Status of Diagrams. Cambridge University Press, Cambridge (1994)
6. Stapleton, G., Masthoff, J.: Incorporating negation into visual logics: A case study using Euler diagrams. In: Visual Languages and Computing 2007, pp. 187–194. Knowledge Systems Institute (2007)
7. Swoboda, N., Allwein, G.: Using DAG transformations to verify Euler/Venn homogeneous and Euler/Venn FOL heterogeneous rules of inference. Journal on Software and System Modeling 3(2), 136–149 (2004)
8. Thomas, W.: Classifying regular events in symbolic logic. Journal of Computer and System Sciences 25, 360–376 (1982)

# An Attention Based Theory to Explore the Cognitive Affordances of Diagrams Relative to Text

Peter Coppin

Faculty of Information, University of Toronto, 140 St. George St.,
Toronto, Ontario, Canada, M5S 3G6
peter.coppin@utoronto.ca

**Abstract.** *Why proofs are (usually) text based* [1] *if diagrams reduce inferential load [1]?* The purpose of this poster is to explore ergonomic affordances of text that may encourage its use in proofs by building on prior work in attention [2,3]. We claim that textual notations may focus a reasoner's "spotlight" of attention through serialized sequential chunks, whereas many diagrams may "diffuse" attention (enabling a reasoner to discern how elements fit together holistically, but with less focus on each element or reifying and explicitly representing connections/relationships between them).

**Keywords:** attention, visual thinking, proof, logic, geometry.

## 1 Introduction

*Why are most geometric proofs usually "non-visual" (e.g., textually based)?* Exploring this question exposes issues that could inform the design of notation systsems to increase abilities to conceptualize, comprehend, and communicate ideas in education, public policy, and beyond. For example, Shimojima and Katagiri [4] demonstrated that diagrams reduce "inferential load" during reasoning by scaffolding visual-spatial aspects of memory and attention (cf. [5,6]). If diagrams reduce inferential load, then it would seem to follow that a diagrammatic notation system would emerge as the dominant ergonomic paradigm for proofs. However, as [1] described: *"[The diagram] is only an heuristic to prompt certain trains of inference…it has no proper place in a proof…the proof is a syntactic object consisting only of sentences arranged in a finite and inspectable array."* (as quoted in [7]). Thus, we sought to explore ergonomic affordances of text (relative to diagrams) that may encourage their use in proofs by examining the role of text and diagrams in Proposition 35 of Euclid's elements. A better understanding of text relative to diagrams could enable more effective communication in math, logic, science, education, engineering, public policy, and beyond.

By building on prior work in attention [2,3], we claimed that textual notations may focus a reasoner's "spotlight" of attention through serialized sequential chunks, enabling the methodical presentation of a rational argument in a way that is not

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 301–303, 2010.

possible (or is very difficult given current understandings and practices) when using a diagrammatic notation that may "diffuse" attention; such notations may enable a reasoner to discern how elements fit together holistically, but place less focus on individual steps in the argument, or on reifying and explicitly representing connections/relationships between steps.

## 2 Prior Work: Language May Guide Attention through Visual-Spatial Structures

Larkin and Simon suggested in [8] that a cognitive dimension of sentences is their list-like structure; each item on the list is only adjacent to the item before or after it on the list. In contrast, items in a diagram are adjacent to many items on a list. This is synergistic with Barwise and Etchemendy [8]. We extrapolate that a list-like structure (as suggested by Larkin and Simon) can be inferred or induced from a diagram. Each sentence inferred from a diagram is like a path that guides attention through visual-spatial relationships in a diagram. This is demonstrated by Shimojima and Katigiri's eye tracking study in [4] that showed how reasoners mentally guide their attention through a "non-physical drawing." They suggest that actual drawings support non-physical (mental) drawings, reducing inferential load. Thus, it appears that sentences guide attentional paths through both physical and non-physical (mental) visual-spatial structures and that rational language/propositional logic guides attention and motor movements (through eye fixations) through non-physical visual-spatial representations. ***What is it about sentences guiding attentiona paths through both physical and non-physical (mental) visual-spatial structures that might enable (an experience of) rigor?*** Clues from the cognitive psychology of attention follow.

### 2.1 Research in Attention

Using a spotlight metaphor, Treisman [3] described how attention can either be narrowed to focus on a single feature (focal attention [FA]), when we need to see what other features are present and form an object, or diffused over a whole group of items which share a relevant feature (diffused attention [DA]) [3]. Text may guide focused attention through such structures to build precisely focused but less holistic ideas.

## 3 Theory: Text Affords Rigor by Directing Focal Attention

***1. Why proofs are often sequential:*** FA may "walk" through different parts of a conceptual visual-spatial structure due to "narrower," but more intense focus (rather than experiencing / attending to the whole structure at once). ***2. Why a diagram usually cannot constitute a convincing "holistic" proof:*** The need for symbols to fall within the narrow spotlight of FA means that diagrams, and the spatial relationships they embody, usually cannot be taken in (i.e. attended to) all at once. So they should instead be processed in a way that allows linkages between earlier perceptual memories and later percepts. ***3. Why text is effective for proofs:*** External symbolic representations such as text are designed such that each symbol can sequentially fall within the narrow focus of FA.[1] ***4.Why propositional logic is used for proofs:*** Propositional structures in proofs may provide symbolic "short-cuts" as stand-ins for visual-spatial relationships that cannot all simultaneously fall in FA.

## 4 Implications for Notation Design

Focal notations (e.g. text) may afford: 1) atoms of meaning sparse enough to be perceived using FA (relative to DA), 2) sequential symbols, and 3) a hierarchical structure whereby complex relationships can be explored by referencing previous perceptual symbols in the sequence. A better understanding may the design heterogeneous proof systems. E.g, a visual serialized chunking system for FA could be rigorous. A next step is to empirically "validate" these design principles.

**Acknowledgements.** P. Coppin is supported by an NSF GRF. Thanks to S. Hockema, J. Burton, and reviewers who contributed to these ideas.

## References

[1] Tennant, N.: The Withering Away of Formal Semantics? Mind & Language 1, 302–318 (1986)
[2] Scholl, B.J.: Objects and attention: the state of the art. Cognition 80, 46 (2001)
[3] Treisman, A.M., Gelade, G.: A feature-integration theory of attention. Cognitive Psychology 12, 97–136 (1980)
[4] Shimojima, A., Katagiri, Y.: An Eye-Tracking Study of Exploitations of Spatial Constraints in Diagrammatic Reasoning. Diagrammatic Representation and Inference, 74–88 (2008)
[5] Ballard, D.H., Hayhoe, M.M., Pook, P.K., Rao, R.P.N.: Deictic Codes for the Embodiment of Cognition. Behavioral and Brain Sciences 20, 723–767 (1995)
[6] Spivey, M.J., Geng, J.J.: Oculomotor mechanisms activated by imagery and memory: Eye movements to absent objects. Psychological Research 65, 235–241 (2001)
[7] Barwise, J., Etchemendy, J.: Visual information and valid reasoning. In: Visualization in Teaching and Learning Mathematics, pp. 9–24. Mathematical Association of America (1991)
[8] Larkin, J.H., Simon, H.A.: Why a diagram is (sometimes) worth ten thousand words. Cognitive Science 11, 65–100 (1987)

[1] A weakness is that the model ignores expertise in enabling chunking of less-sequential structures. This issue is a topic of future research.

# How Does Text Affect the Processing of Diagrams in Multimedia Learning?

Krista E. DeLeeuw[1], Richard E. Mayer[2], and Barry Giesbrecht[2]

[1] Knowledge Media Research Center, Konrad-Adenauer-Strasse 40,
72072 Tuebingen, Germany
k.deleeuw@kmrc-iwm.de

[2] Department of Psychology, University of California, Santa Barbara,
Santa Barbara, California 93106-9660
{mayer,giesbrecht}@psych.ucsb.edu

**Abstract.** In multimedia learning, adding written text to a diagram tends to hurt learning when the verbal information could instead be presented as narration. Explanations of this effect have been based on models of working memory. The current study employs eye tracking to explore an alternative explanation: that this effect stems from differences in perceptual processing instead. The results of this study suggest that accompanying diagrams with written text leads to perceptual (visual) overload. This finding represents an important distinction for explanations of multimedia learning effects.

**Keywords:** Multimedia learning, working memory, perceptual processing, text modality, cognitive load theory.

## 1 Introduction

When people learn from an instructional lesson containing diagrams and text, there are many demands on their limited cognitive resources. Together, these processing demands may overload the learner's cognitive resources. Theories of multimedia learning – such as cognitive load theory (CLT, [1]) and the cognitive theory of multimedia learning (CTML, [2]) – have described effects of overloading processing capacity during learning and have prescribed design principles that are aimed at avoiding this overload. For example, several studies (see [3]) have shown that people learn better from diagrams accompanied by narration than from diagrams accompanied by written text. What has not yet been empirically explained, however, is the underlying processing that leads to this type of effect.

In general, CLT and CTML tend to rely on theories of working memory (WM) [4] to explain these types of effects. For meaningful learning to occur, diagrams and words must be actively processed in WM. Additionally, any processing required only by the instructional design (referred to as *extraneous load*) is also thought to occur in WM. The most common type of evidence for this argument comes from studies that show decreased scores of learning outcome when the instructional materials require extraneous processing above the capacity of one's WM [5].

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 304–306, 2010.

However, differences in the required perceptual processing are also likely because of physical differences between multimedia designs. The aim of the current study is to empirically test which type of processing is affected by adding written text to a diagram. In other words, how does the addition of written text (as opposed to narration) affect how a learner processes a diagram? Lavie and colleagues [e.g. 6] have established a methodology that allows for a distinction between perceptual load and WM load. In these studies, the typical task involves a visual search task containing task-irrelevant distractors. The influence of the distractors decreases when the perceptual load of the task is high, but increases when the WM load of the task is high. In a high perceptual load situation, perceptual resources are unavailable to additionally process distractors. In a high WM load situation, WM resources unavailable to inhibit responses to distractors.

The current study adapted this task by presenting visual task-irrelevant distractors during multimedia lessons containing diagrams and either narration (diagrams+narration or D+N), onscreen text (diagrams+text or D+T), or both (diagram+text+narration or D+T+N), which are thought to vary in extraneous load, from low to high, respectively. The primary dependent measure of this study was the number of times learners looked at these distractors during learning, with the assumption that learners look at distractors because they have noticed them [7], when they would instead benefit from ignoring them.

According to the *working memory hypothesis*, when a lesson requires extraneous processing for reading text, learners should have fewer free WM resources available to inhibit responses to distractors and should therefore make more eye fixations on the distractors. According to the *perceptual hypothesis*, when the lesson requires extraneous processing for reading text, learners will have fewer perceptual (visual) resources available to process the distractors, leading to the learners making fewer eye fixations on the distractors.

## 2 Method and Results

Thirty-six college students learned about brakes [8], toilet tanks [9], and lightning [10] in a within-subjects design, consisting of three levels of extraneous load (D+T, D+N, D+T+N) counterbalanced across the three lessons, each of which contained several static diagrams depicting functional steps in the processes described in the accompanying narration and/or text. During each lesson, distractors randomly appeared for 400 msec on the screen at one of four locations – to the left and right of the diagram and to the left and right of the text (or text placeholder). Each lesson was between 1.5 minutes and 3 minutes long, allowing for between 96 and 146 presentations of the distractor stimuli.

A repeated-measures ANOVA showed that there was a significant difference in the proportion of fixations on distractors among the conditions, $F(2,70)=7.56$, $p=.001$. Post-hoc comparisons showed that participants made a significantly higher proportion of fixations on distractors during the D+N condition ($M=.025$, $SD=.028$) compared to the D+T ($M=.005$, $SD=.008$; $p<.001$) and D+N+T ($M=.010$, $SD=.028$; $p=.02$) conditions, which did not differ significantly from each other ($p=.30$). In other words, participants made a significantly higher proportion of fixations on the distractors for the low-load condition (i.e. when there was no text on the screen) compared to the higher-load conditions (i.e. when there was text on the screen). This result supports the perceptual hypothesis: learners were more distracted (by the onscreen distractors) when extraneous load was low compared to when extraneous load was high.

## 4 Summary and Discussion

The main finding of this study is that learners attended to distractors less when extraneous load was increased. The proportion of eye fixations on distractors was less for high-load conditions (i.e. lessons containing onscreen text) and more for the low-load condition (i.e. lessons in which verbal information was off-loaded from the visual channel by using narration). In other words, learners looked at distractors less when there was text on the screen. This finding does not appear to be merely an effect of a smaller functional field of view during reading in the higher-load conditions because this effect held even if only fixations on the diagram and on the distractors (and not on the text) were taken into account. These results suggest that visual information contained in displays with diagrams and onscreen text required all of the learner's perceptual processing resources, leaving no resources available for processing the distractors, thus supporting the perceptual hypothesis. This indicates that at least this source of extraneous load tends to act as perceptual load rather than working memory load. This finding suggests an important clarification of CLT and the CTML that has not previously been specified or tested. In addition to this finding specific to replacing narration with onscreen text, further studies are currently investigating the perceptual load of other types of multimedia instructional design.

## References

1. Sweller, J.: Cognitive load theory, learning difficulty, and instructional design. Learn. Instr. 4, 295–312 (1994)
2. Mayer, R.E.: Multimedia Learning, 2nd edn. Cambridge University Press, New York (2009)
3. Ginns, P.: Meta-analysis of the modality effect. Learn. Instr. 15, 313–331 (2005)
4. Baddeley, A.D.: Human memory. Allyn & Bacon, Boston (1999)
5. Moreno, R., Mayer, R.E.: Cognitive principles of multimedia learning: The role of modality and contiguity. J. of Educ. Psychol. 91, 358–368 (1999)
6. Lavie, N., Hirst, A., de Fockert, J.W., Viding, E.: Load theory of selective attention and cognitive control. J. of Exper. Psychol. 133, 339–354 (2004)
7. Just, M.A., Carpenter, P.A.: A theory of reading: from eye fixation to comprehension. Psychol. Rev. 87, 329–354 (1980)
8. Mayer, R.E., DeLeeuw, K.E., Ayres, P.: Creating retroactive and proactive interference in multimedia learning. Appl. Cog. Psychol. 21, 795–809 (2007)
9. Hegarty, M., Kriz, S., Cate, C.: The roles of mental animations and external animations in understanding mechanical systems. Cog. and Instr. 21, 325–360 (2003)
10. Mayer, R.E., Steinhoff, K., Bower, G., Mars, R.: A generative theory of textbook design: Using annotated illustrations to foster meaningful learning of science text. Educ. Tech. Res. and Dev. 43, 31–43 (1995)

# An Experiment to Evaluate Constraint Diagrams with Novice Users

Noora Fetais and Peter C.-H. Cheng

Representation and Cognition Research Group, Department of Informatics,
University of Sussex, Falmer, Brighton, BN1 9QH, UK
{N.Al-Marri,P.C.H.Cheng}@sussex.ac.uk

**Abstract.** Constraint diagrams [1, 2] constitute a graphical notation for program specification. This paper presents an experiment that compares the efficacy of constraint diagrams (CD) with natural language (NL) for understanding program specification statements. In a web-based competition 33 participants were given training either on CD notation or equivalent NL expression, and answered questions about specification statements. It was predicted that the CD participants would find learning and answering questions harder than those in the NL group, because they had no prior experience of CD notation. Although the CD group spent more time on the training examples and were less confident about their answers, they spent less time answering the questions and achieved approximately the same proportion of correct answers as the NL group.

## 1 Introduction

Constraint diagrams (CD) are a notation created by Kent [1] that may be simpler and more effective than other systems for specifying programs [2] because of their visual nature. However, the opposing argument, which is only supported with examples, states that constraint diagrams are not always well matched to the intended meaning [3]. Additionally, although there are some studies on CD, they were either an experiment with no direct comparison with other methods [4], or a comparison of CD with Visual OCL from a theoretical perspective without empirical evaluation [5]. We are interested in whether CD can be used for program specification by individuals with little formal knowledge of programming; hence an experiment was conducted to contrast CD with program specification statements in natural language (NL). Can CD be a suitable language for communication between clients who are experts in their own domain but novices at programming, and software engineers?

## 2 Experiment

Some may question the legitimacy of using NL as the comparator in this study instead of formal languages. However, NL is a notation that is neither demanding nor beyond novice users' capabilities, which are needed for communicating requirements between the client and the software engineer.

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 307–309, 2010.

There were five measures: time spent on the training examples; time to answer the questions; percentage of correct answers; confidence rate; and number of returns to the examples. There were two domains employed in the material: "Video Rental Service" which was used in the examples, and "Patient Record System" which was used in the questions. There were eight concepts: sets and types; members of sets; sets' relations; relationships; spiders; spiders' relationships; invariant; and event specification. The number of questions, as determined by a pilot, was 24. An example had (I) a title of the introduced concept, (II) a diagram in the case of the CD version or a statement in the case of the NL version, (III) a description, and (IV) a definition of important parts in the concept. Each concept/example was followed by three questions. A question had (i) a title including the question number, (ii) a diagram in the CD version or a statement in the NL version, (iii) a question, (iv) three answering options: "Yes", "No" and "Not Specified", (v) a judgment of the level of difficulty of the question on a five-level scale, and (vi) feedback on the answer.

A between-participants design was adopted with two separate groups and two different representations, CD and NL, of the same material, comprising questions and examples which were informationally equivalent [6]. The university students, entering the competition for prizes, were randomly assigned, using training web-based software, to either the CD version or the NL version. Training was required to mask individual differences in cognitive styles [7].

Although 53 participants began this experiment, we analyzed data of 33 participants who completed more than half of the questions. There was a general trend in the proportion of correct answers and confidence rate, each decreasing from the earlier to the later questions, but the general trend increased for both time spent on the questions and on the training examples from the earlier to the later. The question-by-question variability increased substantially in the second half of the questions for both correct answers and question time, and increased substantially in some parts of the questions for confidence rate. The example-by-example variability however, did not follow the general trend. It decreased substantially in some parts for the time spent on the training examples.

By using a chi-square test on correct answers and t tests on question time, confidence rate and example time, it was found that there were no significant differences between the two groups. Comparing the CD group with the NL group, the first group achieved 12 correct answers, 23 questions were answered faster, 8 questions were answered more confidently and 1 example was studied faster. By statistically measuring the time spent per question, the CD group was better than the NL group. However, when measuring the confidence rate and when measuring the time spent per example, the CD group was worse than the NL group. In addition, the CD group was no worse than the NL group when measuring the proportion of correct answers. As the level of chance performance in answering a question was 0.33, a chi-square test comparing the numbers for the two groups was not significant.

Pearson product moment correlation showed that the performance of the two groups was not substantially different because both groups found the same questions/examples easy or difficult. It showed that between variables, such as (a) question time and correct answer averages, (b) confidence rate and correct answer averages, and (c) question time and confidence rate averages, the values were not statistically significant. Although it appeared that a question that took longer was less accurate and there was less confidence involved, there was only a weak indication that a longer time spent on the question meant more accurate and confident results. Furthermore, a higher confidence rate led to less accurate answers, but there was only a weak indication that more confidence in answering meant less accurate results. While the NL group tended to be less confident about answering difficult questions, the CD group's confidence rate for easy or difficult questions was approximately the same.

## 3 Discussion

Unpredictably, learning CD notation was not difficult. The results showed that the CD group had the ability to achieve the same proportion of correct answers as the NL group and in less time, but they spent more time on learning and were less confident about answering, as predicted. Although CD notation had the benefit of being formal, it might be as good as using NL for comprehension of program specification statements. The results of our study provided practical indications for enabling non-technical users to use CD. More specifically, our study showed that users without prior experience in CD benefited from training. In particular, using CD decreased the interpretation time required to understand the program specification. There were no obvious classes of questions/concepts that one of the representations dealt with better with than the others [3]. Notably, the results of this study showed that participants did not find the same questions confusing, and thus there were no systematic differences between the two groups. Despite the fact that the number of correct answers and the level of confidence decreased across time, item analysis showed that there was no particular question that was easier in one representation than in the other.

**Acknowledgments.** The authors wish to thank John Howse for his comments on the experiment material.

## References

1. Kent, S.: Constraint diagrams: Visualizing invariants in object oriented models. In: Proceedings of OOPSLA 1997, pp. 327–341. ACM Press, New York (1997)
2. Howse, J., Schuman, S.: Precise visual modelling. In: SoSym (2005) (to appear)
3. Stapleton, G., Delaney, A.: Evaluating and generalizing constraint diagrams. Journal of Visual Languages and Computing 19, 499–521 (2008)
4. Fish, A., Masthoff, J.: An experimental study into the default reading of constraint diagrams. In: VLHCC 2005, Proceedings of Visual Languages and Human Centric Computing, Dallas, pp. 287–289 (2005)
5. Fish, A., Howse, J., Taentzer, G., Winkelmann, J.: Two visualizations of OCL: A comparison. Technical Report VMG.05.1, University of Brighton (2005)
6. Larkin, J.H., Simon, H.A.: Why a diagram is (sometimes) worth 10000 words. Cognitive Science 11, 65–99 (1987)
7. Frandsen, A.N., Holder, J.R.: Spatial visualization in solving complex verbal problems. Journal of Psychology 73 (1969)

# “Graph-as-Picture” Misconceptions in Young Students

Grecia Garcia Garcia and Richard Cox

Representation and Cognition Lab, Sussex University, Falmer, BN19RH, UK

**Abstract.** This paper presents work on the relatively under-researched topic of children’s graphical knowledge. It extends Janvier’s [1] research on the graph-as-picture misconception (GAPm, e.g. interpreting a line graph as the picture of a mountain) by identifying the prevalence of GAPm in a younger population and by examining gender differences. We used an established method for identifying GAPm and non-GAPm individuals and we also investigated the behaviour of these two groups of students on a diagram/picture decision task designed to provide insights into how students mentally represent their graphical knowledge. One in four of the students studied showed evidence of possessing a GAPm, with a higher prevalence in boys than in girls.

## 1 Introduction

Janvier[1] studied a phenomenon he described as “the simultaneous use of the graph at the symbolic and pictorial levels” (p.118, [1]) in a small sample of secondary school students. A graph-as-picture misconception (GAPm) is deemed to be present when a person interprets an abstract diagrammatic representation as the picture of an object or scene (e.g. interpreting a line graph of a polynomial function as a pictorial representation of a mountain). However, Janvier [1] did not report how *prevalent* GAPm was in the sample he studied. Hence, one aim of the research reported in this paper was to establish the prevalence of GAPm in a sample of young (elementary school) students as a starting point for thinking about GAPm prevention and remediation. Our further aims were to study GAPm across a range of graphs and diagrams since Janvier used only line graphs as stimuli in his study.

We adapted Janvier’s [1] paper-and-pencil method of assessing the GAPm by implementing an interactive version on a touchscreen computer system. This was because results from using the paper-and-pencil graph-comprehension test does not provide insights into a student’s mental representation and cognitive processing. We used the paper-based test but also added an additional activity consisting of a graphical decision task. The graphical decision task was designed from a cognitive information-processing perspective similar to that employed in object-recognition studies (e.g. [2]). The graphical decision task used a heterogeneous corpus of diagrammatic representations and pictures and allows a broader range of GAPm’s to be studied. Using the additional task we were able

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 310–312, 2010.

to study *X-as-picture misconceptions* where "X" is one of six representational types (Section 2). The new task also allowed us to look at *bidirectional* misclassification patterns, that is, errors characterizing a diagram as a picture *and* errors characterizing a picture as a diagram.

## 2 The Study

The participants were 48 elementary school students (23 girls and 25 boys): 13 third-grade (mean age in months: 94.3), 14 fourth-grade (106.2), 12 fifth-grade (116.3), and 9 sixth-grade (129.4). Thirty-seven attended a public (state) elementary school and 11 were children of university staff. The children participated in two activities: (1) a graphical discrimination task and (2) a line graph comprehension test. The line graph comprehension test was used to identify students with and without the GAPm, to establish its prevalence in our sample and to compare the behaviour of students in the GAPm and no-GAPm groups in terms of their performance on the graphical decision task.

The **Graph Comprehension Questionnaire (GCQ)** consisted of a multiple choice questionnaire based on Curcio and Smith-Burke [3] and Janvier [1]. A gender "neutral" cover story topic for the scenario was chosen (a child taking a walk). The questionnaire presented four line graph comprehension questions: the first two asked the child to read one of the axes (first the x-axis, then the y-axis); the next question required an operation over two values of the graph; and the final item required an interpretation of the line graph. The interpretation question included the following response options: "He walked slow in the first 3 hours and then he went fast"; "He walked fast and then he went across a bridge"; "He walks fast in the first 3 hours and then he walks slow"; "He walks up a hill and then goes down the other side"; "Other answer".

In the **Graphical Decision Task (GDT)** a total of 144 items were used. Each item consisted of a monochrome JPEG image of a diagram or a picture drawn by the first author. Care was taken to equate the diagrams (graphs) and pictures in terms of visual complexity and visuospatial features. The diagrams were based on published primary school resources (e.g. UK National Curriculum material, teaching resource books, etc.). There were six forms of graphical (diagrammatic) representation: tables, bar charts, line graphs, pie charts, hierarchies/network diagrams, and set diagrams. Representations were presented in 72 unique pairs (picture-graph, graph-graph, picture-picture). One pair was presented per trial. Stimuli were presented in the same order to all participants. Each child was instructed to sort each card by dragging it using her finger on the touchscreen into one of 2 areas which were labelled "more like diagrams, charts, graphs" and "more like pictures". The task was first demonstrated by the experimenter who showed that on some trials the images should be sorted into different areas and sometimes both stimuli in the pair were sorted into the same area. The student then performed the task and continued until 5 minutes elapsed or she had finished the 72 trials.

## 3 Results and Discussion

Students were divided into three different groups on the basis of their responses to the interpretation question. The first group consisted of those who chose the "hill" response option (GAPm group); a second group was comprised of children who responded with the "bridge" option (GABridge group); and the third group consisted of those students who did not choose either the "hill" or "bridge" response options (non-GAPm). Of the 45 children, 33 (73.3%) were classified as not having the GAPm (non-GAPm group). Sixteen students in the non-GAPm group responded correctly to the interpretation question. Twelve students (26.7%) were either in the GAPm or GABridge groups (9 and 3 children, respectively). Therefore approximately one in four students manifested a graph-as-*another-representation* misconception. As mentioned earlier, the problem scenario we chose for the GCQ was designed to be gender-neutral. GAPm was found to be more prevalent in boys than in girls, contrary to Janvier's finding [1]. One in five girls showed a GAPm compared to one in three boys.

The GDT allowed us to look at representational misclassification error patterns bidirectionally, i.e. diagrams to picture and also picture to diagrams. Our results show that the GAPm group tended to classify diagrams as pictures across *all* the representational forms. The proportions of *X-as-picture* errors per group were: 70.6% (GAPm group) vs. 0.0% (non-GAPm group) in hierarchies/network diagrams; 60.0% vs. 19.7% in set diagrams; 40.0% vs. 12.0% in pie charts; 27.3% vs. 5.0% in line graphs; 21.4% vs. 7.8% in bar charts; and 21.1% vs. 5.4% in tables. Hence, *X-as-picture* misconceptions were more common in GAPm students. Results of the GDT suggest that the "graph-as-picture misconception" framework requires some extension and refinement. We suggest that the term *X-as-picture* or *graph-as-another-representation* misconception better captures our findings than the term "graph-as-picture misconception". The results also seem to be consistent with a "knowledge in pieces" perspective on misconceptions [4]. What has hitherto been referred to as a graph-as-picture misconception might better be described as a collection of semi-misconceptions that vary widely across individuals. Knowledge and misconceptions are probably fragmented [4] and are not elicited equally by all forms of representation.

## References

1. Janvier, C.: Use of Situations in Mathematics Education. Educational Studies in Mathematics 12(1), 113–122 (1981)
2. Humphreys, G.W., Riddoch, M.J.: Picture naming. In: Humphreys, G.W., Riddoch, M.J. (eds.) Visual Object Processing: A Cognitive Neuropsychological Approach, pp. 107–143. Lawrence Erlbaum Associates, Hillsdale (1987)
3. Curcio, F.R., Smith-Burke, M.T.: Processing information in graphical form. American Educational Research Association, (ERIC Document Reproduction Service No. ED215874), New York (1982)
4. Smith, J.P., di Sessa, A.A., Roschelle, J.: Misconceptions Reconceived: A Constructivist Analysis of Knowledge in Transition. Journal of the Learning Sciences 3(2), 115–163 (1993/1994)

# What Students Include in Hand-Drawn Diagrams to Explain Seasonal Temperature Variation

Victor R. Lee

Department of Instructional Technology and Learning Sciences, Utah State University, Logan, UT, USA
victor.lee@usu.edu

**Abstract.** This study seeks to understand how hand-drawn diagrams are made to represent ideas related to the cause of the seasons. It draws from a set of diagrams produced by a sample of high school students (N=652) in the United States who were asked to describe their naive understandings regarding the cause of the seasons. Using codes applied to students' short, written explanations of the cause of the seasons and codes applied to drawings that were produced, this paper describes patterns seen across diagrams and identifies a few depiction trends associated with specific categories of explanations.

**Keywords:** astronomy, seasons, K-12 education, schematic diagrams, drawings.

## 1 Introduction and Study Overview

This study seeks to understand how drawn diagrams are typically used to represent ideas related to the cause of the seasons. Drawings and written responses were gathered using an open-ended assessment instrument administered to 652 ninth-grade students in the United States at the beginning of their academic year. In the portion of the assessment instrument considered for this study, the students were asked to first explain in writing why there was temperature variation between summer and winter. Immediately following that written response, each student was encouraged to provide a diagram to support his or her written explanation. While the provision of a student-authored diagram was optional, 496 students (76%) provided their own diagrams along with their written explanation for the seasons.

Written student responses were reliably coded ($\kappa = 0.80$) based on the content of the explanation that was provided, using the following categorization scheme:

1. *Closer-farther explanations*: In this incorrect, albeit common [1, 3] explanation for the seasons, students explain the cold temperatures of winter coming about as a result of increased overall distance from the sun.
2. *Side-based explanations*: This is another incorrect category of explanation for the seasons, in which either the eastern or western hemisphere is oriented toward, or facing the sun, and therefore will be warmer.

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 313–315, 2010.

3. *Tilt-based explanations*: In this category, axial tilt plays prominently. Tilt-based explanations include the scientifically accepted explanation, in which the Earth's axial tilt makes the incoming solar energy more or less direct for different parts of the earth, and the less accurate explanation in which the tilt makes either the northern or southern hemisphere be closer to the sun.
4. *Climate-base explanations*: This explanation can to be tautological in structure: "It's cold in winter because that's when it's cold". Presence of snow, ice, and cloudy skies in the winter is described as the cause of the cold months, while the warmer sun and less cloud-cover is described as the cause of warmer months.

A second set of codes was applied to the drawings independent of what was written in the explanations. These codes specified primarily the graphical elements in each of the drawings. These codes included such as "Sun represented as a partial circle", "Orbit depicted as an oval shape", "Axis line drawn on Earth shape", etc. In total, there were 71 possible codes in this second coding scheme.

## 2 Initial Findings

In total, there were 152 closer-farther responses (23.3%), 101 side-based (15.5%), 294 tilt-based (45.1%) and 46 climate-based (7.1%). For each category, there were 115, 83, 232, and 31 drawings, respectively. Twenty-three more belonged to no category.

Nearly all drawings included the sun and the Earth, indicating that some elementary basis for conceptualizing the seasons as an astronomical phenomenon was present. There was a split in terms of how the sun was rendered. Distance-based explanations tended to more heavily involve full-circles for suns (92.2% of the time) and climate-based ones tended to use partial circles (32.3% of the time).

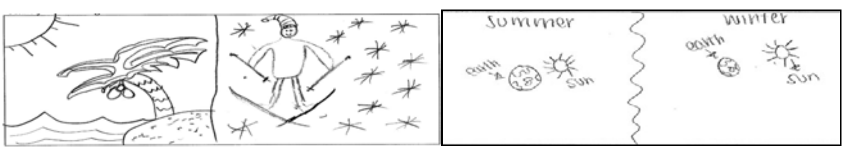



**Fig. 1.** Typical climate-based (left) and distance-based (right) diagrams

As shown in Table 1, there were some slight differences across how the drawn Earth and sun sizes compared in the three most frequently given explanations (distance-based, side-based, and tilt-based ) ($\chi^2$ = 11.70, 4df, $p < 0.05$). More students with tilt-based explanations tended to draw the Earth and sun as equal in size than other groups. Still, regardless of explanation, most students in each group depicted the sun as larger than the earth.

Orbital paths appeared in diagrams relatively infrequently. Surprisingly, this held true for those students who gave distance-based explanations. It is often expected that distance-based explanations will involve students conceptualizing a highly elliptical or eccentric orbital path [2, 4]. In this study, most distance-based students simply drew two earths, with one closer to the sun and another than was farther from it (Figure 1). When orbits were included, there were only a small number that conformed to the elongated orbit representation (17/107). Still fewer instead used a roughly circular shape (11/107), but positioned it so that the sun was located away from the center. Also, despite the importance of travelling light to explaining the seasons, most of the students' diagrams lacked any sort of marker for light rays or solar energy moving from the sun to the Earth. Only 16.1% of all diagrams in the corpus with suns and Earths (78/484) had some graphic element between the earth and sun indicating sunlight or energy.

**Table 1.** Size of drawn Earth relative to drawn sun across three seasons explanations. Note that multiple suns or Earths may have been drawn, and were therefore multiply coded.

| Earth (E) size relative to Sun (S) | Distance-Based explanation | Side-based explanation | Tilt-based explanation |
|---|---|---|---|
| E < S | 50.0% (53/106) | 39.2% (29/74) | 42.9% (88/205) |
| E = S | 23.6% (25/106) | 27.0% (20/74) | 38.5% (79/205) |
| E > S | 33.0% (35/106) | 37.8% (28/74) | 23.4% (48/205) |

Finally, when drawing the Earth, the graphic elements students used were very simple, typically involving a circle that was only sometimes crossed by an axis line (126/484). More axes lines appeared in drawings of tilt-based explanations (45.2%) compared to the other explanations (12.6%). While it is nearly half, that frequency is still a bit lower than expected considering the importance of axial tilt to a tilt-based explanation.

## 3 Summary

When it comes to students' diagrams explaining why we have seasons, there is a great deal of variation but still some common trends. Students' diagrams often do not include important ideas relevant to the seasons. Specifically, very few students show the tilt of the Earth or the movement of solar energy from the sun to the Earth. Also, while many students knew to represent the sun as larger than the Earth, a majority of students still did not. That may suggest that fragmented or incomplete models are in place mentally when students are tasked with visually representing the seasons.

## References

1. Atwood, R.K., Atwood, V.A.: Preservice elementary teachers' conceptions of the causes of seasons. Journal of Research in Science Teaching 33, 553–563 (1996)
2. Duschl, R.A., Schweingruber, H.A., Shouse, A.W.: Taking science to school: Learning and teaching science in grades K-8. The National Academies Press, Washington (2007)
3. Newman, D., Morrison, D.: The conflict between teaching and scientific sense-making: The case of a curriculum on seasonal change. Interactive Learning Environments 3, 1–16 (1993)
4. Schnepps, P., Sadler, P.: A private universe: An insightful lesson on how we learn. Pyramid Film & Video, Santa Monica (1988)

# Diagrammatic Specification of Mobile Real-Time Systems

Sven Linker*

Carl von Ossietzky University of Oldenburg
sven.linker@informatik.uni-oldenburg.de

**Abstract.** Behavior of spatio-temporal systems depends on real-time as well as spatial aspects. More and more safety-critical systems fall into this domain and thus raise the urge for formal specification and verification methods for this type of systems. For this purpose, we develop a diagrammatic language of *Shape Diagrams* that concentrates on the critical concepts and is usable by both engineers and scientists. We present two syntaxes, an abstract one based on hypergraphs and graph transformation systems that constitutes the abstract structure, and a concrete one given in terms of conventions for drawing diagrammatic pictures.

**Keywords:** diagrammatic specification, spatio-temporal systems, formal reasoning, graph transformation.

In the last decades, the use of *real-time systems*, i.e., systems which are required to react to given inputs within a certain time bound, has dramatically spread, especially in safety-critical areas. *Mobile* real-time systems additionally have to respect spatial constraints and relations to ensure safe behaviour. Examples of such systems would be cars organizing themselves automatically as platoons, aircraft controlling devices, or, on a lesser scale of criticality, automated and autonomous vacuum cleaners. Due to the complexity of these systems, methods for formally verifying their correct behaviour are highly desired. However, mathematical formalisms allowing for proofs of the correctness of mobile real-time systems with respect to time bounds as well as both qualitative and quantitative spatial constraints are sparse. To the best of our knowledge, the only formalism capturing spatial and temporal aspects in a uniform manner is *Shape Calculus* [1], a multi-dimensional logic interpreted on models based on polyhedra.

Diagrams are an often used engineering method to enhance communication between engineers during development. Hence a diagrammatic language suited for the specification of mobile real-time systems is desired. To bridge the gap between engineering tasks and the formal verification of correct behaviour, such a language should be equipped with formal semantics.

For this purpose, we propose the language of *Shape Diagrams* (see Fig. 1). In the following, we briefly describe a concrete and hint at the definition of

* This work is supported by the German Research Foundation (DFG), grant GRK 1076/1.

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 316–318, 2010.

an abstract syntax for this language. We establish the *concrete syntax* informally by conventions. Shape Diagrams consist of a stack of *layers*, which denote successive points in time. Layers are depicted by rectangles parallely projected onto the drawing plane. The interior of a layer describes a spatial situation, where objects are abstracted to labelled rectangles called *shapes*. To reduce the complexity of Shape Diagrams, we do not allow for arbitrarily shaped objects. For non-rectangular objects, safe bounding boxes have to be used. To represent restrictions on the durations between layers as well as the distances between objects and the borders of layers, arrows annotated with real-valued intervals are employed. Note that arrows constraining distances may connect shapes across layers. In such a case, auxillary lines have to be drawn to resolve ambiguities. We support an *assumption/commitment-style* reasoning, i.e., to express that under certain assumptions, a system is required to fulfill the commitments, we employ shading. E.g. the diagram in Fig. 1 asserts, that if two cars drive one after the other at a distance of 60m, they are required to build a platoon within 10 to 30s, i.e., the rear car follows the car in front at a distance between 2 and 3m.

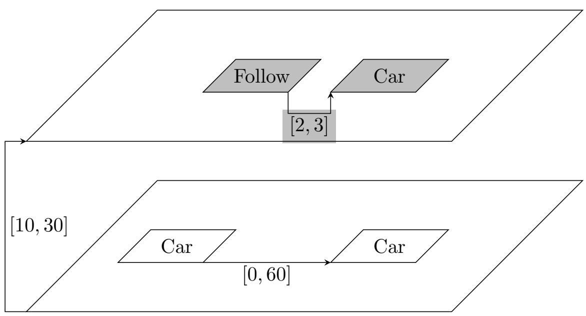



**Fig. 1.** A Shape Diagram

The *abstract syntax* of Shape Diagrams has to represent the following entities and properties: the diagrammatic elements (layers, shapes and arrows), the attachment relations of arrows, the positions of shapes in relation to each other and the order of layers. To reflect these different aspects in the abstract syntax, we use the notion of *hypergraphs*, i.e. graphs where an edge may connect an arbitrary number of nodes via its so-called *tentacles*. We employ typed hyperedges, i.e. the type of an edge determines the number and names of its tentacles.

The part of an abstract syntax graph representing the lower layer of Fig. 1 is depicted in Fig. 2. Following Minas [2], each diagrammatic element is represented by a hyperedge. The different elements are distinguished by hyperedge types, e.g. we employ the types shape and tarrow for shapes and constraints on durations, respectively. The vertices of the graphs denote *attachment areas* of the elements, e.g. each tarrow edge visits two vertices with the tentacles $s$ and $t$. The layer edges visiting these nodes are representing the source and target of the arrow. The description of the relative position of a shape is more difficult. We use an approach developed by Guesgen [3] and Nabil et al. [4] generalizing interval relationships to more than one dimension. The idea is based on projecting the objects onto the axes, thus obtaining an interval on each axis for each shape.

Then the relations of two intervals in each dimension can be stored indepedently, by hyperedges representing the interval relations visiting attachment nodes of shape edges. In Fig. 2, the edges labelled = and < denote the relative positions of the shapes in Fig. 1.

For a formal definition of the abstract syntax, we employ a *graph transformation system*. Such a system consists of an axiom, i.e. a hypergraph, and transformation rules to obtain new graphs by repeatedly replacing parts of the axiom resp. the resulting graphs, similar to textual grammars. The rules differ strongly in their complexity. For example, the rule for the creation of layers is context-free, i.e. it only replaces a single hyperedge without taking other incident edges into account. For the creation of the spatial arrows, i.e. hyperedges of type sarrow, the existence of at least one shape or layer is necessary. Hence these rules are only applicable in a certain embedding context. The most complex rules create the relative spatial positions. They make use of both embedding contexts and application conditions [5], because on the one hand at least two shapes have to be present for edges defining relative positions, but on the other hand exactly one relation between two shapes has to be created for a complete syntax graph.

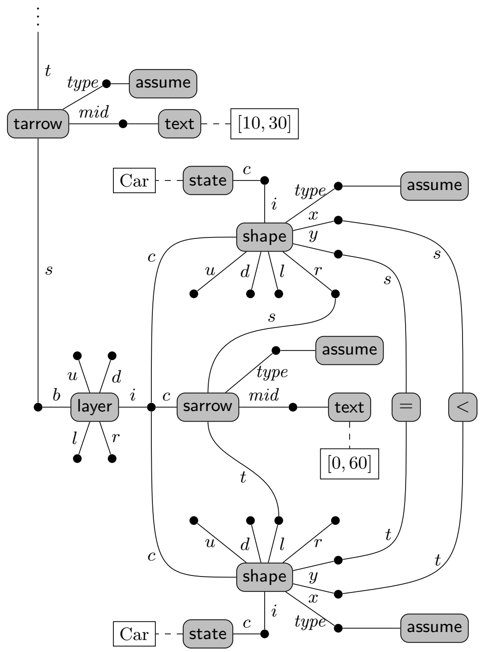



**Fig. 2.** Part of the Abstract Syntax Graph of Fig. 1

## References

1. Schäfer, A.: Specification and Verification of Mobile Real-Time Systems. PhD thesis, University of Oldenburg (2006)
2. Minas, M.: Hypergraphs as a uniform diagram representation model. In: Ehrig, H., Engels, G., Kreowski, H.-J., Rozenberg, G. (eds.) TAGT 1998. LNCS, vol. 1764, pp. 281–295. Springer, Heidelberg (2000)
3. Guesgen, H.W.: Spatial reasoning based on Allen's temporal logic. Technical Report TR-89-049, Int'l. Comp. Science Inst., Berkeley (1989)
4. Nabil, M., Shepherd, J., Ngu, A.H.H.: 2D projection interval relationships: A symbolic representation of spatial relationships. In: Egenhofer, M.J., Herring, J.R. (eds.) SSD 1995. LNCS, vol. 951, pp. 292–309. Springer, Heidelberg (1995)
5. Habel, A., Pennemann, K.H.: Correctness of high-level transformation systems relative to nested conditions. MSCS 19, 245–296 (2009)

# Manipulatable Models for Investigating Processing of Dynamic Diagrams

Richard Lowe[1] and Jean-Michel Boucheix[2]

[1] Curtin University, Australia
r.k.lowe@curtin.edu.au
[2] University of Burgundy, France, LEAD-CNRS
Jean-Michel.Boucheix@u-bourgogne.fr

**Abstract.** Existing approaches for collecting process data on human diagram comprehension have limited effectiveness. Analogue models that allow participants to manipulate diagram components offer powerful ways to capture the non-verbal and dynamic aspects of processing that are not available with some other approaches. Examples drawn from a variety of different domains illustrate the utility of model manipulation for revealing otherwise inaccessible aspects of how people process animated diagrams of complex content.

**Keywords:** Dynamic diagrams, processing, manipulatable models.

## 1 Introduction

Collecting useful research data about human processing of diagrams is a challenging task. However, such data are essential prerequisites for increasing the effectiveness of diagrams as tools for thinking, communication, and learning. One barrier to research in this area is the dearth of established techniques for capturing evidence about how diagrams are processed. In this paper, we discuss techniques that can provide deeper insights into this processing through data produced when people manipulate models. These approaches appear particularly useful for understanding how the dynamic aspects of animated diagrams are processed. The popularity of animations over static diagrams is likely due to their capacity to represent the subject matter dynamics directly rather than indirectly. However, the very dynamics that allow them to depict their referents in a behaviourally realistic manner can also pose processing challenges for viewers. To minimize these challenges, we need a better understanding of how people process the information that animated diagrams present.

Educational researchers investigate the effects of diagrams on learning. In one widely used approach, participants study a diagram (typically accompanied by written or spoken text) then produce written answers that indicate the outcomes of their learning experience. Although this approach has produced useful findings about issues such as how to improve multimedia learning outcomes, it is not suited to capturing fine-grain detail of the diagram processing by which these outcomes are generated. Further, the inherent limitations of textual output as a medium for representing dynamic graphic information mean that the data collected provide a somewhat indirect

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 319–321, 2010.

and impoverished indication of what has occurred. Eye-tracking is increasingly used to investigate how people process diagrams. Despite having proven very effective with static diagrams, it is more difficult to use with animations. In addition, eye tracking provides only a very limited sample of the processes involved during the comprehension of graphics. It captures the distribution of a viewer's foveal fixations across space and time but does not reveal the reasons for that distribution. Although eye tracking indicates what information is perceived, it gives no direct indication of the cognitive processes associated with those perceptions.

## 2 Collecting Better Process Data

The search for better ways to understand how people process animated diagrams needs to be based on a clear analysis of the representational format being studied and the perceptual/cognitive characteristics of the human viewer. To avoid possible distorting effects of inter-representational mediation, there should be a reasonably direct one-to-one mapping between the diagram being studied and the representation used to capture processing data. This is the case with eye tracking – fixation patterns map directly onto the stimulus diagram used. However, it is clearly not so with verbal measures. Words are ill suited to capturing the visuospatial and temporal detail necessary to characterize the subtleties present in even relatively simple animations.

If we assume that human information processing operates in a parsimonious fashion, it is likely that much of the information viewers internalize from an animated diagram is not represented verbally, initially at least [1]. Rather, the visuospatial and temporal details are probably represented internally via a mental model with structural characteristics analogous to those of the diagram. In this case, the form of external representation we use to probe the processes by which internal representations are developed should ideally have a similar character. Such representations should allow research participants to reveal information about their perceptual and cognitive processing of the diagram with a minimum of mediation.

## 3 Evidence from Manipulation

In studies of diagram processing involving different types of depicted dynamic content, we have used manipulatable models with attributes as described above. These models vary in the extent to which their visuospatial characteristics parallel those portrayed in the animated diagrams used as stimulus materials. Manipulation of a model involves generative processing based on the available information and can reveal aspects of processing that are not tapped by textual output or eye tracking data. With some types of content, such as a traditional piano mechanism, the model has a close visual resemblance to the diagram [2]. This physical model is constructed from acrylic sheet and incorporates pivots that allow the experimental participant to move its component parts in a manner similar to the depicted movements. Data collected using this approach reflect how participants mentally represent the movements of piano parts (sequence, extent, speed, etc.). From these data, we are able to infer possible causal relationships that underpin their developing interpretation of the depicted mechanism. With other types of content, the model omits aspects that are shown in the animated diagram so as to simplify manipulation and/or remove unwanted dynamic contingencies. An example of this type of model was used in a study that investigated viewer extraction of high and low perceptibility information from an animated five-ball Newton's Cradle diagram [3]. Here, the model consisted of five small coins that the participant manipulated with both hands to demonstrate different aspects of the animation's dynamics. This model deliberately omitted the supporting strings by which the balls were suspended in the animation. Participant demonstrations of Newton's Cradle dynamics using this model were especially useful for collecting data about their understanding of how subtle changes in spacings of the balls developed over the time course of the phenomenon.

Both the piano and Newton's Cradle diagrams depict mechanisms made from rigid components. However animated diagrams are also used to depict non-rigid systems, such as those found with biological content. In a study of learning from an animated kangaroo locomotion diagram, participants demonstrated aspects of the movement using a simplified plastic model [4]. In the animation, the shape of the kangaroo's body parts altered as their configuration changed during hopping. However, the model used was made from a set of discrete, rigid parts connected by pivots. Despite this simplification, participants' manipulations of the model produced much useful data such as how the angles between specific body parts changed during the locomotion process. A further biologically-oriented study investigated the learning of locomotion patterns from an animated depiction of a swimming fish [5]. One of the measures for this study involved participants using a track ball to simulate the movement of a missing fish body part (the head, middle or tail section). This manipulation produced useful quantitative data regarding the inferred frequency and amplitude of movements associated with the missing body parts. In this case, the model being manipulated was even further removed in a physical sense from the situation depicted in the animation. However, it still provided participants with a means of demonstrating the spatial and temporal changes in a way that paralleled those of the referent subject matter.

## References

1. Schnotz, W.: An Integrated Model of Text and Picture Comprehension. In: Mayer, R.R. (ed.) The Cambridge Handbook of Multimedia Learning, pp. 49–69. Cambridge University Press, New York (2005)
2. Lowe, R.K., Boucheix, J.-M.: Complex Animations: Cues Foster Better Knowledge Structures? In: 13th EARLI Conference, Research Paper. Amsterdam, Netherlands (2009)
3. Lowe, R.K., Schnotz, W.: Animations and Temporal Manipulation: Supporting Comprehension of Complex Dynamic Information. In: 12th EARLI Conference, Research Paper. Budapest, Hungary (2007)
4. Smalberger, C., Lowe, R.K.: Animation Display Speed and the Learning of Complex Scientific Content. In: World Conference on Science and Technology Education, Research Paper. Perth, Australia (2007)
5. Boucheix, J.-M., Lowe, R.K.: When Less is More: Eye Tracking Comparison of Dynamic Inference During Viewing and Mental Simulation. In: 13th EARLI Conference, Research Paper. Amsterdam, Netherlands (2009)

# Can Text Content Influence the Effectiveness of Diagrams?

Anne Schüler, Katharina Scheiter, and Peter Gerjets

Knowledge Media Research Center,
Konrad-Adenauer-Str. 40, 72072 Tübingen, Germany
{a.schueler,k.scheiter,p.gerjets}@iwm-kmrc.de

**Abstract.** Verbal instructional explanations are often accompanied by diagrams to enhance learning. Recent working memory research suggests that a high degree of spatial information conveyed through the verbal explanations may impair the instructional effectiveness of the diagrams. It is explained by interference between the processing of the diagrams, the processing of text containing spatial information and the execution of eye movements, associated with diagram inspection. In the current paper two experiments are summarized which showed the expected performance impairment when presenting diagrams together with text containing spatial information. Implications of these results for the presentations of diagrams are discussed.

**Keywords:** working memory; spatial information; multimedia learning.

## 1 Introduction

In the last decades, advances in technology have enabled the frequent use of diagrams, which often accompany verbal instructional explanations to visualize the information described in the text. The Cognitive Theory of Multimedia Learning [CTML; 1] provides a theoretical framework for learning with text and diagrams. Based on Baddeley's working memory model [2] the CTML assumes that texts are processed in the phonological loop, whereas diagrams are processed in the visuo-spatial sketchpad (VSSP). Because the two systems are limited in the amount of information that can be processed in parallel, text-diagram presentations should be designed in a way that interference within the same system can be avoided.

Since the introduction of the working memory model the structure of the VSSP has been further specified; however, these findings have not yet been considered in text-diagram research. Accordingly, the VSSP consists of a visual part which mostly deals with processing visual characteristics of objects (e.g., shape or color), and a spatial part which is responsible for handling relational or spatial information [3]. Furthermore, under very specific conditions not only diagrams but also text will be processed in the VSSP, namely, if the text contains information about visual (e.g., colors or shapes) or spatial aspects (e.g., locations of objects or spatial relations) [4, 5]. Finally, the spatial part of the VSSP is not only responsible for the processing of spatial information but also for the control of movements, especially eye movements [6]. These

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 322–324, 2010.

findings have implications for the effectiveness of different text-diagrams combinations as will be shown in the following.

## 2 Hypotheses and Empirical Evidence

As mentioned, one feature of text that can influence the processing demands of the VSSP is the text content. Whereas diagrams are assumed to be processed in the visual and spatial part of the VSSP, because they normally contain visual as well as spatial information, texts load the visual or spatial VSSP as a function of their content. Texts containing no visuo-spatial information at all do neither load the visual nor the spatial part of the VSSP. However, when presenting texts containing visual information the visual part of the VSSP becomes loaded, whereas when presenting texts containing spatial information the spatial part of the VSSP becomes loaded. Furthermore, as the spatial component controls the execution of eye movements looking at diagrams will result in an additional load of the spatial component. Based on this analysis, we assume that text containing spatial information interferes with the processing of diagrams in the spatial VSSP, because the processing of spatial diagrammatic contents, text containing spatial information, and the control of eye movements take place there. When presenting diagrams together with text that does not contain any spatial information, one would expect less interference because the load is distributed more equally across the different VSSP parts. Accordingly, diagrams presented together with text containing spatial information should result in worse learning outcomes than diagrams presented together with text not containing spatial information, that is, non-visuo-spatial or visual information.

We conducted two studies to test this hypothesis. Learners were presented with six static diagrams and accompanying verbal explanations about fictitious fish species (study 1) or fictitious plants (study 2). Among other things, we varied between subjects whether the texts contained information about visual features of the depicted learning object, that is, the color or form of specific parts (e.g., "The pectoral fin has the same light brown color as the dorsal fins") or whether it contained information about spatial features of the depicted object, that is, the location of a part or its spatial relation to other parts (e.g., "The pectoral fin lies between the two dorsal fins"). Note that the length and the grammatical structure of the texts containing visual or spatial information were equivalent and that both text versions referred always to the same features of the learning object depicted in the diagram. Furthermore, both texts contained identical non-visuo-spatial information on biological concepts and facts (e.g., "The fins are used for defense"). After the learning phase, learners had to recall the information presented in the text as well as the information presented in the diagram. The results showed that learners with text containing spatial information performed worse when the test items required them to recall the text information (studies 1 & 2), the visual diagrammatic information (study 1), the spatial diagrammatic information (study 1), or both, text and diagrammatic information (study 2). The only exception was observed in study 2 regarding the recall of spatial diagrammatic information, where learners with text containing spatial information outperformed learners with text containing visual information. With regard to the recall of non-visuo-spatial information, no differences between groups were observed. This may be seen as an indicator that text containing no visuo-spatial information does not interfere with diagram processing in the VSSP. In sum, learners who received diagrams together with text containing spatial information showed overall worse performance in recalling text-based and diagram-based information than learners who received diagrams together with text containing visual information. Importantly, learners with text containing spatial information did not only show worse performance when having to recall the text information, but also when having to recall the diagrammatic information, which was the same in all conditions. This result supports the assumption of interference between the processing of spatial text information and the processing of diagrams.

## 3 Conclusion

Our results confirm the assumed interference between text containing spatial information and the processing of diagrams. From a theoretical view, these results show that findings concerning working memory have also implications for more applied text-diagram research. Different information sources can compete for the same cognitive resources in working memory, which in turn leads to poorer performance. From a practical view, the presentation of text containing spatial information together with diagrams should be avoided. To provide spatial information it seems better to present diagrams because they provide direct and parsimonious access to visuo-spatial information and are more computationally efficient for accomplishing tasks that require the processing of visuo-spatial properties [7]. The reported studies are only a first step in applying basic cognitive research to more applied scenarios which is needed to develop more precise theoretical frameworks for explaining how learning with text and diagrams works. In a next step it should be analyzed whether the found patterns are also valid when no artificial but ecologically valid materials are used, where hence learners may vary with respect to the degree of prior knowledge that they possess.

## References

1. Mayer, R.E.: Multimedia Learning. Cambridge University Press, Cambridge (2009)
2. Baddeley, A.D.: Working Memory. Science 225, 556–559 (1992)
3. Logie, R.H.: Visuo-spatial Working Memory. Lawrence Erlbaum, Aberdeen (1995)
4. De Beni, R., Pazzaglia, F., Gyselinck, V., Meneghetti, C.: Visuospatial Working Memory and Mental Representation of Spatial Description. Eur. J. Cogn. Psychol. 17, 77–95 (2005)
5. Deyzack, E., Logie, R.H., Denis, M.: Visuospatial Working Memory and the Processing of Spatial Description. Brit. J. Psychol. 97, 217–243 (2006)
6. Postle, B.R., Idzikowski, Z., Della Sala, S., Logie, R.H., Baddeley, A.: The Selective Disruption of Spatial Working Memory by Eye Movements. Q. J. Exp. Psychol. 59, 100–120 (2006)
7. Larkin, J.H., Simon, H.A.: Why a diagram is (sometimes) worth ten thousand words. Cog. Scien. 11, 65–99 (1987)

# Attending to and Maintaining Hierarchical Objects in Graphics Comprehension


Atsushi Shimojima[1], Yasuhiro Katagiri[2], and Kozue Enseki[1]

[1] Doshisha University, 1-3 Tatara-Miyakodani, Kyotanabe, Kyoto, Japan
[2] Future University Hakodate, 116-2 Kamedanakano, Hakodate, Hokkaido, Japan



**Abstract.** Information graphics convey different levels of information, depending on how their elements are grouped into different units of objects. How people set boundaries to graphical objects to be interpreted and how they maintain the object boundaries during the given task are two important problems in understanding the way people comprehend information graphics. Table comprehension process was experimentally investigated in terms of eye gaze control behaviors when people were required to read off information distributed over large-scale objects, e.g., a row or a column, of an alphanumerical table. Our data suggest that attention can spread over large-scale objects and that some eye-movements occur to maintain the attended objects. We provide an explication of these observations applying recent findings about object-based attention and visual indexing.



**Keywords:** graphics comprehension, object-based attention, visual index, eye-tracking, embodied cognition.


**Approach.** Researchers and designers have noted that information graphics can express "higher-level" information as well as "lower-level" information. The former roughly indicates more abstract information carried by overall patterns formed by multiple graphical elements, while the latter more concrete information carried by individual graphical elements. For example, the locations of individual dots in a scatter plot indicate the existence of individual data points with specific values. In addition to this lower-level information, a scatter plot can express higher level-information by the shape and the density of a large-scale pattern formed by these dots. While the lower-level information is concerned with the values taken by individual samples in the data, the higher-level information is concerned with the overall distribution of the data, such as the strength of correlation between the two variables.

The present study is concerned with the fundamental operations necessary to extract different levels of information from the given graphics. Our approach is defined by the following leading questions:

1. how people set boundaries to graphical objects to be interpreted.
2. (when the reading task is complex enough) how they maintain object boundaries during the task.

**Preliminary Experiment.** In order to explore the attentional process to different levels of graphical objects and the maintenance process of attended objects, we conducted a preliminary eye-tracking study on 23 college students who were engaged in table


A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 325–327, 2010.

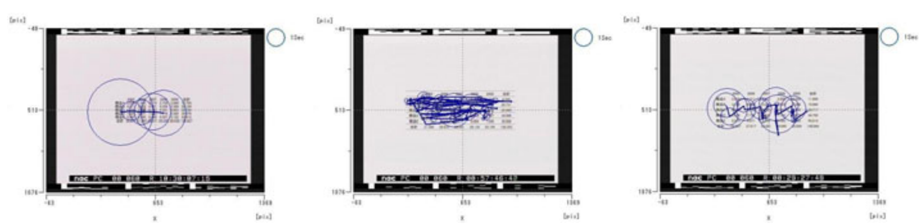
comprehension tasks. The stimuli are $6 \times 6$ alphanumerical tables, showing annual sales figures of different products. Four-digit numbers were used as sales figures for all the cells except for those showing total figures. The stimulus tables came in two different inter-cell spacings, *dense* and *sparse*. They also came in three patterns of grid lines, *vertical only*, *horizontal only*, and *none*.

The task was designed so that participants need to attend to objects in a table at two different hierarchical levels: cells at the small object level (*cell* task), and columns and rows at the large object level (*column* task and *row* task). It is expected that attention to different levels of objects should reveal itself as difference in eye movement patterns.

**Observation 1: Spreading Attention over Large-Scale Objects.** A particularly interesting phenomenon was found in the dense-table, column-task conditions. The average number of vertical saccades on dense tables was 5.6 for the column task. Considering each table contains the total of 5 columns of figures to be processed, this means that, on average, less than 1.2 saccades were made per column. Thus, under this condition, eyes move from columns to columns consecutively, often placing only one fixation on each column. Figure 1 (a) shows the typical fixation pattern on dense tables during the column task. Fairly long fixations were placed on individual columns, but eyes tended to leave a column after no or few vertical movements within the column. On the other hand, significantly more vertical saccades were observed in the sparse -table, column-task condition and in the row-task condition in general.

This suggests an interesting contrast in the way consecutive figures are integrated into an object under different conditions. Under the latter conditions, multiple fixations were placed on individual columns or rows of figures. This indicates that initial attentions were oriented to consecutive sub-regions of the column or row, each consisting of one or a few figures. Due to the task demand, these initially attended sub-regions were then integrated into a column or row of five figures, to which a task-relevant judgment was attached and kept attached during the trial. The type of *subsequent integration* seems to be necessary in these conditions, due to the large size and the wide inter-cell spacing of the objects to be integrated.

In contrast, often under the dense-table, column-task condition, only one fixation was placed on an entire column of five figures. This suggests that a single attention spread over the entire column, integrating five figures into an object already at the time of initial attention. Subsequent integration was not necessary in such a case, and task-relavent judgment could be directly attached to the initially attended object. This could



(a) (b) (c)

**Fig. 1.** Eye movements under column-tasks on (a) dense table without horizontal grid lines, (b) sparse table with horizontal grid lines, and (c) sparse table without horizontal grid lines. Circles indicate fixations, with the diameters roughly showing the fixation durations.

be enormous simplification of the relevant comprehension task, and it is consistent with a significantly shorter response time observed in the dense-table condition during the column task.

This process of spreading attention corresponds to “bounded activation,” or “coloring,” in S. Ullman’s theory of visual routines, whose function is to define coherent units of regions in the unarticulated visual scene so that further operations can be applied selectively to the activated regions. P. R. Roelfsema and his colleagues made this idea more exact by proposing computational and neurological models of the operation, and provided neuro-physiological evidence to its functioning in macaque monkey. “Object-based attention” actively investigated by J. Duncan and his successors largely overlaps with the operation of bounded activation. Strong empirical evidence for the operation has been accumulated in this tradition too.

**Observation 2: Eye Movements for Maintaining Large-Scale Objects.** Another consistent pattern in our data was concerned with the effect of orthogonal grid lines: vertical saccades during the row task were significantly more frequent on tables with vertical grid lines, and horizontal saccades during the column task were significantly more frequent on tables with horizontal grid lines. Figure 1(b) and 1(c) show typical fixations patterns during the column task, on tables with or without horizontal grid lines.

One plausible explanation for this pattern refers to the cost of maintaing large objects previously integrated. On this view, for example, horizontal saccades in the column task increased in the presence of horizontal grid lines because the horizontal grid lines divided vertical sequences of figures rather effectively, making it more difficult to keep them integrated as coherent objects. Thus, extra horizontal saccades were necessary in order to return attention to their locations, reintegrate them if necessary, and check the task-relevant judgment attached to them. Without such additional operations, the attached judgments would have been lost together with the objects to which they had been assigned.

This sort of maintenance processes is a realistic possibility given the visual indexing mechanism investigated by Z. Pylyshyn and his colleagues. According to the visual indexing theory, we can assign “indices” to several objects or locations in the visual scene. With these indices, we can quickly return attention to the locations of indexed objects without searching for them. We hypothesize that such an index was attached to a row or column of figures when it was first integrated. The present case is unique in that object groups, rather than individual objects or locations, were indexed and revisited for their tags.

**Conclusions.** Thus, the data from our preliminary experiment support the view that in order to fully understand how people extract higher-level and lower-level information, graphics comprehension research should explicitly consider the issues of how they set the boundaries of graphical objects to be interpreted, and how the object boundaries are maintained during the task. Our exploratory study with table-reading tasks illustrates the utility of this approach to graphics comprehension processes, especially when it is equipped with the supporting theories of object-based attention and visual indexing.

# Modelling English Spatial Preposition Detectors

Connor Smith, Allen Cybulskie, Nic Di Noia, Janine Fitzpatrick, Jobina Li, Korey MacDougall, Xander Miller, Jeanne-Marie Musca, Jennifer Nutall, Kathy Van Bentham, and Jim Davies

Institute of Cognitive Science, Carleton University,
1125 Colonel By Drive, Ottawa, ON, Canada K1S 5B6
jim@jimdavies.org

**Abstract.** In this paper we present five algorithms for the detection of spatial relationships within an image: above/below, adjacent to, occlusion, between, and close to.

**Keywords:** natural language, spatial relationships, cognitive modeling

The use of spatial relations has importance in understanding what people take away from diagrams, and the generation of scene descriptions.

We present five computational spatial-relation detectors intended to functionally model human perceptions corresponding to English words: above/below, adjacent_to, occlusion, between, and close_to. We model them after high-level representations found in language.

According to [1], spatial terms parse the space around some reference object(s) into regions, with some regions being more prototypical characterizations than others. Fuzzy logic captures this vagueness, describing the truth value of a spatial proposition with a real number between 0 and 1.

We use image data from an online game called Peekaboom [2], which consists of labels associated with point clouds in images. That is, we know "where" the label is in each image. For every pair of labels our system uses the detectors to generate spatial propositions with corresponding fuzzy values. For example, if a cup is above a table, our system might output a fact such as (`cup above-below table .9`).

**Above/Below.** The computations we use are based on Bloch's [3] fuzzy approach.

The target object is assigned a status of 'above' or 'below' the reference object based on whether its centroid's y-axis has a lesser or greater value than that of the reference object's y-axis value. A slope is then found for a line connecting the target and reference object's centroids using the rise and run. If the run equals 0 then the line is perfectly vertical and the relationship receives a belief score of 1.0. Otherwise, an angle of deviation is calculated by taking the arctangent of the line slope; this represents the angle between the line and the horizon. We calculate the truth value by applying: the vertical trigometric, sin^2, function of the arctangent (sin^2(angle)).

**Adjacenct_to.** People intuitively see two objects as adjacent if some edge of object A is near, or touches, some edge of object B [4]. Additionally, we hypothesize another

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 328–330, 2010.

relevant factor: that object area in relation to the shortest straight line that can be drawn between the facing edges of the objects. Intuitively we may judge two very small objects as not adjacent if the distance between them is much larger than the area of one or both of the objects.

If the objects overlap, or if there is an object between the target and reference, the truth value returned is 0.0. If the shortest distance between the objects is <= 1 (pixels), a truth value of 1.0. A percentage is then derived from a ratio of minimum distance between the objects over the size of the target object. A truth value is then returned as such (ratio to truth value): <2.0 to 1.0, <6.0 to 0.8, <12.0 to 0.5, <18.0 to 0.2, and >20.0 to a value of 0.0.

**Occlusion.** A contour junction in occlusion is where two contours meet and one of them appears to abruptly end. Most contour junctions are called T-junctions, where the stem is the occluded contour and the crossbar is the occluding contour [5]. Closure is when the continuation of a stem from one junction closes off a region by connecting to the stem of another junction; non-closure is when the stems do not close off a region [6].

There are three cases of occlusion to account for: A) where a smaller object occludes a larger object, and the smaller object is completely contained within the larger object's convex hull, resulting in no contour junctions. If one of the objects has all of its coordinates located within the convex hull of the other, it is returned as the occluding object. B) is when occlusion occurs but neither object is completely contained within the other. Lines of the convex hulls are checked to determine whether they partially share line functions – if so, the algorithm looks to find a line that forms a junction with the shared line. The object with the line that forms a junction with the shared line is returned as the occluded and the other as the occluding. C) is somewhat similar to Case B, however it deals with instances of closure, where the occluded lines of an object are perceived to (or do) meet at a point.

Truth values are then output based on the percentage of the occluded object's area that the occluding object occupies. Example: 90% = 0.9, 80% = 0.8, etc.

**Between.** The algorithm finds the centroid and area for all three objects. If the target hull overlaps either reference object by 50% the algorithm returns a truth value of 0.0. Otherwise, the convex hull of betweenness ($\beta_{CH}$) is generated - defined as per [7]. This joins the reference objects' hulls together and then returns a hull representing the space between them. If the target object has no overlap with either reference object and its centroid falls within $\beta_{CH}$, a truth value of 1.0 is returned.

If the overlap of the target convex hull with a reference object is between 0-50%, the detector returns a value calculated by the formula: 1 - (greatest overlap ratio/2). The greatest overlap refers to the greatest ratio of a reference object's convex hull which is overlapped by that of the target object.

If the target centroid does not fall within $\beta_{CH}$, the algorithm: (i) calculates the proportion of the target hull that overlaps with $\beta_{CH}$ to obtain $overlap_{ConvexBetween}$, and (ii) draws a circular field of betweenness, $\beta_{Circ}$, between the reference objects, and determines the proportion of the target hull that overlaps with $\beta_{Circ}$ to obtain $overlap_{CircBetween}$. The diameter of $\beta_{Circ}$ is the shortest distance, x, between the reference objects, and its centroid is the midpoint of x. The truth value is the greater of $overlap_{ConvexBetween}$ and $overlap_{CircBetween}$.

**Close_to.** We propose two hypotheses for ‘close to’ detection: The ad hoc unit scaling hypothesis and the framed space comparison hypothesis.

The ad hoc unit scaling hypothesis states that the distance between object A and object B is judged by estimating how many copies of A fit in the space between the objects (Robert Thomson, personal communication, February 13, 2009). This predicts that the order of presentation of the objects will affect the distance judgment. This is important when the two objects vary largely in size.

The framed space comparison hypothesis states that while objects remain the same distance apart in absolute terms, the larger the image frame, the closer the objects appear to each other.

First, an ad hoc unit scale judgement is made by finding the closest points between the target and reference. The line between these points is then extended through the target object to find the ad hoc unit. This unit is compared to the distance between the objects by finding a distance to ad hoc unit ratio. A belief value is then assigned from the ratio. Example (ratio to truth value): <=1.0 to 1.0, <2.0 to 0.9, <2.9 to 0.8, <3.1 to 0.5, <4.1 to 0.2, and otherwise a 0.0 truth value.

The second function makes a frame-space comparison judgement. It finds the area between the objects and calculates a ratio of the area to the overall area of the image. This ratio is then assigned to a truth value. Example (ratio to truth value): <0.020 to 1.0, <0.025 to 0.8, <0.033 to 0.5, <0.05 to 0.2, and otherwise a value of 0.0.

The two judgements are united using a function which takes the belief value of each judgement, assigns a weight to them, and then outputs their combined belief value.

By modeling spatial relationships after spatial terms in natural language we are able to capture linguistic parameters associated with different spatial relations as well as some of the perceptual mechanisms behind them.

## References

1. Hayward, W.G., Tarr, M.J.: Spatial Language and Spatial Representation. Cognition 55, 39–84 (1995)
2. von Ahn, L., Liu, R., Bloom, M.: Peekaboom: a Game for Locating Objects in Images. In: Proceedings of the Special Interest Group on Computer-Human Interaction Conference on Human Factors in Computing Systems, pp. 55–64. Association for Computing Machinery, Montreal QC (2006)
3. Bloch, I.: Fuzzy Spatial Relationships for Image Processing and Interpretation: a Review. Image Vision and Computing 23, 89–110 (2005)
4. Rosenfeld, A., Klette, R.: Degree of Adjacency or Surroundedness. Pattern Recognition 18(2), 169–177 (1985)
5. Sedgwick, H.A.: Visual Space Perception. In: Goldstein, E.B. (ed.) Blackwell Handbook of Perception. Blackwell Publishers Ltd., Massachusetts (2001)
6. Boselie, F., Wouterlood, D.: A Good-Continuation Model of Some Occlusion Phenomena. Psychological Research 54, 267–277 (1992)
7. Bloch, I., Colliot, O., Cesar Jr., R.M.: On the Ternary Spatial Relation “Between”. IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics 36, 312–327 (2006)

# Diagram Interpretation and e-Learning Systems

Neil Smith, Pete Thomas, and Kevin Waugh

Computing Research Centre, The Open University
Milton Keynes, MK7 6AA, UK
{n.smith,p.g.thomas,k.g.waugh}@open.ac.uk

**Abstract.** We describe a system capable of grading free-form diagrammatic answers. Our matches meaningful parts of a diagram with equivalents in a model solution. This is complicated by errors, omissions, and superfluous items in the student answer. The result of matching is used to calculate the grade and generate appropriate feedback; it performs at least as well as a human marker on a variety of diagram types. We describe tools that allow the easy creation of questions, marking schemes, and diagram editors suitable for embedding in a VLE quiz engine.

**Keywords:** automatic grading, imprecise diagrams, e-learning.

## 1 Introduction

We have produced a system that automatically assesses (summatively and formatively) student-produced diagrams. Such diagrams are normally *imprecise*: they do not match the expected diagram in some way. We handle graph-based diagrams commonly found in technical subjects (e.g. UML class and sequence diagrams). Such diagrams encode meaning in the connection of lines and boxes, adornments to diagram elements, and the spatial positioning of elements.

Our work contrasts with semi-automatic marking systems [1,2] (which present the human marker with an ungraded answer and apply the same grade to equivalent answers) and deduction systems for diagrammatic formulae [3]. In our work, the emphasis is on extracting as much information as possible from an imprecise diagram.

Our approach compares a student *answer* diagram with one or more model *solution* diagrams. Comparison is based on identifying the *minimal meaningful units* (MMUs) in each diagram. An MMU is a partial diagram which, if any element is deleted, no longer has meaning. The first step is to identify MMUs in the answer that correspond to MMUs in the solution. However, as the student diagram is imprecise, we use a measure of similarity between MMUs, based on properties of the MMUs and, most significantly, the identifying label.

By default, we model object labels as noun phrases and relationship labels as verb phrases. Processing identifies head words in the label, which are stemmed and compared using edit distance, taking synonyms into account. This process also deals with punctuation and abbreviations.

Handling synonyms is difficult as there are many equivalents for terms occurring in the answer. Common and specialist synonyms are stated explicitly in the

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 331–333, 2010.

mark scheme. The multiple synonym problem [4] is ameliorated by stemming (to reduce words to a canonical form) and similarity measures (to estimate the closeness of words). We can capture many synonyms from corpora of student answers [5,6].

## 2 The Matching Process

The similarity between two diagrams is determined by finding the best overall correspondence of MMUs. As far as possible, each MMU in the answer is matched with an MMU in the solution to give a diagram correspondence. This will be incomplete due to omissions and errors in the answer. The chosen diagram correspondence is the one which maximises the sum of the similarities of matched MMUs. MMUs unmatched after this stage are compared on the basis of their context (the set of related objects). Once a correspondence has been determined, a mark scheme, based on MMUs, is applied and feedback is given.

Thresholds are used to judge whether or not two aspects are sufficiently similar to be considered to match. Thresholds are also used in marking to determine whether to award marks for some aspect of an answer. Weights indicate the relative importance of diagram elements for both matching and marking.

Experiments performed on corpora of several hundred student answers show that our system performs at least as well as a human marker [5].

## 3 Supporting Applications

Question setting, answering, and marking are performed by domain-specific tools. Using another application, the teacher specifies the drawing tool (Fig. 1) by describing the domain and the amount of freedom students have to create incorrect diagrams, based on pedagogic requirements. The teacher must also specify the characteristics of the marking tool, including relevant weights and thresholds.

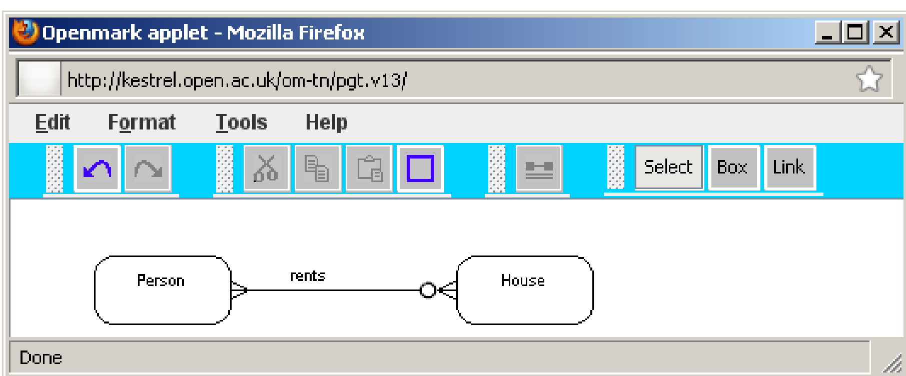



**Fig. 1.** The diagram drawing tool. It can run stand-alone or as an applet

Another application supports the teacher in developing questions, model solutions, mark schemes, and feedback. The marking, drawing, and question-setting tools are all derived automatically from generic ones, based on the specifications given.

Prototypes have been incorporated into the Open University's Moodle VLE. The VLE presents the student with a question and an applet for drawing an answer. The applet submits the finished diagram to the making engine. In formative mode, the marking engine returns feedback which is displayed in the browser and the student may make multiple attempts at the question.

## 4 Conclusions and Future Work

We have built an automatic marking engine for graph-based diagrams which gives good performance and good agreement with human markers. Nevertheless, we want to perform similar experiments with more diagram types and less constrained questions.

Our work to date has been based on developing bespoke diagram parsers generalising them to support a wider range of diagrams. While effective and useful, this approach will be limiting in future. Therefore, we will reformulate our approach as a constraint-based parser (e.g [7]), extended to use soft constraints to handle imprecision.

## References

1. Batmaz, F., Hinde, C.J.: A Web-Based Semi-Automatic Assessment Tool for Conceptual Database Diagrams. In: Proceedings of the Sixth IASTED International Conference on Web-Based Education, pp. 427–432 (2007)
2. Higgins, C.A., Bligh, B.: Formative Computer Based Assessment in Diagram Based Domains. In: Proceedings of the 11th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE), pp. 98–102 (2006)
3. Howse, J., Stapleton, G.: Visual Mathematics: Diagrammatic Formalization and Proof. In: Proceedings of International Conference on Mathematical Knowledge Management (2008)
4. Jayal, A., Shepperd, M.: The Problem of Labels in e-Assessment of Diagrams. ACM J. Educational Resources in Computing 8(4) (2008)
5. Thomas, P., Smith, N., Waugh, K.: Automatically assessing graph-based diagrams. J. Learning, Media & Technology 33(3), 249–267 (2008)
6. Jordan, S.: Assessment for learning: pushing the boundaries of computer-based assessment. Practitioner Research in Higher Education 3(1), 11–19 (2009)
7. Henning, C.: CHR grammars. Theory and Practice of Logic Programming 5(4-5), 467–501 (2005)

# An Examination of Cleveland and McGill's Hierarchy of Graphical Elements

Brandie M. Stewart[1] and Lisa A. Best[2]

[1] Queens University Centre for Neuroscience 18 Stuart Street Kingston, Ontario, K7L 3N6
brandie@biomed.queensu.ca
[2] University of New Brunswick, Department of Psychology, P.O. Box 5050, Saint John, NB E2L 4L5
lbest@unbsj.ca

**Abstract.** Two experiments were conducted to examine Cleveland and McGills' theory of graph perception. In Experiment 1 participants made judgments about the individual perceptual elements. In Experiment 2, participants were presented with graphs that isolated specific perceptual elements. Although the original hierarchy included ten elements, the current research suggested that, depending on the task, there may be no more than three or four individual rankings. This research presents the first attempt at a comprehensive examination of the relationship between the perception of isolated graph elements and how these elements affect graph reading.

**Keywords:** graph perception; hierarchy of graph elements; psychophysics.

## 1 Basic Geometrical Properties and Graph Comprehension

Cleveland and McGill (1), (2) proposed a theory of graph perception that focused on the perception of the basic geometrical properties, such as position, length, and slope, used to encode quantitative information (see Figure 1). Cleveland and McGill ordered the elements into a hierarchy, ranked from easy to difficult to encode. Carswell (3) examined the predictive power of the basic task model and found that the model best to predicted performance for point-reading or comparison tasks. Overall, her results suggest that the distinction between the different elements may not be as well defined as suggested by Cleveland and McGill's (1), (2).

Although researchers have evaluated the efficacy of Cleveland and McGill's hierarchy, to date, no research has evaluated the ability to distinguish between isolated elements, presented individually. Experiment 1 was designed to examine the accuracy associated with each of original elements. The purpose of Experiment 2 was to determine how well participants were able to use the isolated elements to complete graph comprehension tasks.

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 334–337, 2010.


**Fig. 1.** Cleveland and McGill's hierarchy of graph elements

## 2 Methods

**Participants.** Seventy-two participants ($M_{age}$=20.1 years, *SD* = 4.89) were recruited.

**Materials.** In Experiment 1, the elements were presented on aligned scales and non-aligned scales. Based on pilot testing, four levels of difficulty were selected. In Experiment 2, graphs were constructed to isolate specific elements. Cleveland and McGill's (1), (2) theory was used to identify the main task required for each graph and elements were probed by using different graphs, including bar and divided bar charts, line graphs, scatterplots, pie and doughnut charts, and maps. A comprehension test included read-off, spatial transformation, and interpretation questions (see 4, 5).

**Procedure.** In Experiment 1, participants judged size differences between two geometrical figures and indicated if they were the same or different. In Experiment 2, participants completed a 100-item multiple choice test of graph comprehension.

## 3 Results and Discussion

In Experiment 1, judgements were more accurate when the figures were aligned, $F$ (1,71)=262.22, $p$=.001. Significant differences were observed between the eight elements, $F$ (1, 71) =146.85, $p$=.0001, and post hoc tests suggested that the elements could be grouped into four categories (see Table 1). These results are consistent with previous studies that used psychophysical tasks (7) and whole graphs (7).

**Table 1.** A comparison of the different hierarchies. Ranks in the current study are based on significant differences (absolute rank).

| | *Original* | *Perceptual* | *RO* | *Spatial* | *INT* |
|---|---|---|---|---|---|
| Aligned scale | 1 | 1 (2) | 1 (3) | 1 (2) | 2( 5) |
| Angle | 5 | 3 (9) | 2 (7) | 1 (6) | 2 (3) |
| Area | 4 | 2 (4) | 1 (4) | 1 (5) | 3 (10) |
| Colour saturation | 9 | 2/3 (5) | 1 (2) | 1 (3) | 2 (4) |
| Curvature | 7 | 4 (10) | 3 (9) | 2 (7) | 2 (7) |
| Length | 3 | 3 (6) | 1 (5) | 1 (1) | 3 (8) |
| Non-aligned scale | 2 | 3 (8) | 1 (1) | 3 (8) | 3 (9) |
| Shading | 8 | 1 (1) | 1 (1) | 1 (4) | 1 (1) |
| Slope | 5 | 2 (3) | 2/3 (8) | 3 (9) | 1 (2) |
| Volume | 6 | 3 (7) | 1 (6) | 3 (10) | 2 (6) |

In Experiment 2, there was a statistically significant main effect for question type, $F(2, 71)=204.74$, $p=.001$, and post hoc tests showed that, overall, read-off questions led to highest accuracy, followed by spatial transformation and interpretation questions. The main effect for graph element, $F(9,71)=32.68$, $p=.0001$ was statistically significant and Figure 2 shows that there were fewer differences between the elements than would be predicted by the original theory (also see Table 1).

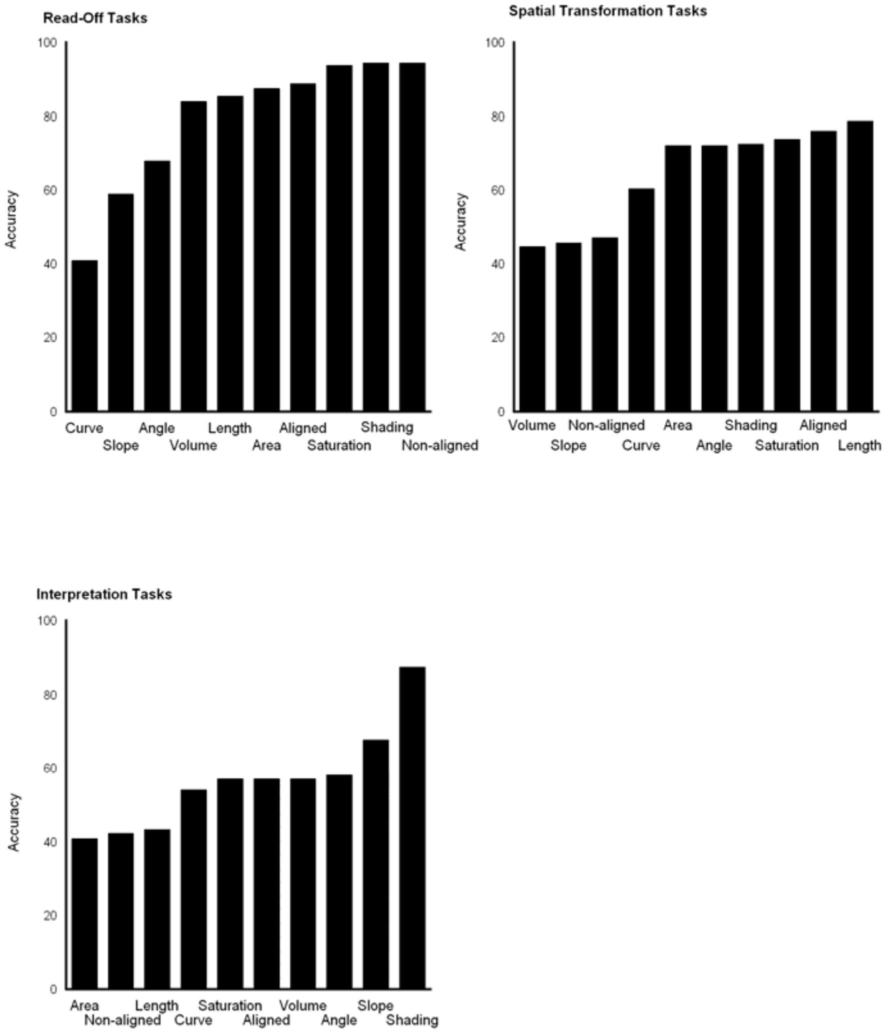



**Fig. 2.** Accuracy for Read Off (Panel A), Spatial Transformation (Panel B), and Interpretation (Panel C) questions

## 4 Conclusions

The purpose of the current research was to conduct an empirical evaluation of Cleveland and McGill's theory of graph perception. Although Cleveland and McGill did not discuss more cognitive aspects of graph reading, the read-off and spatial transformation tasks used in the current study were fairly similar to those used previously. Furthermore, the relatively low levels of accuracy associated with angle, slope, and curvature were consistent with previous theories. There were several inconsistencies between the original (1), (2) and current hierarchies (see Table 1); most noteworthy is the number of ranked elements. Although the original hierarchy included 10 distinguishable elements, the current results suggest that, depending on the task, there may actually be only three or four individual rankings. In the original studies, ranks were based on mean error and statistically significant differences were not reported. In the current studies, statistically significant differences were considered, and fewer differences were observed.

This research presents the first attempt at a comprehensive examination of the relationship between the perception of graphical elements and the effect these elements have on graph cognition. The strong correlation ($r = .81$, $p = .005$) between the rankings in Experiments 1 and 2 suggests that graph perception and cognition are closely related. Furthermore, they also suggest that the effectiveness of perceptual elements depends, to a certain degree, on the required task (see also 8).

## References

1. Cleveland, W.S., McGill, R.: Graphical perception: Theory, experimentation, and application to the development of graphical methods. JASA 79, 531–554 (1984)
2. Cleveland, W.S., McGill, R.: Graphical perception and graphical methods for analyzing scientific data. Science 229, 828–833 (1985)
3. Carswell, C.M.: Choosing specifiers: An evaluation of the basic tasks model of graph perception. Hum Factors 34, 535–554 (1992)
4. Trickett, S.B., Trafton, J.G.: Spatial transformations in graph comprehension. In: Blackwell, A., Marriot, K., Shimojima, A. (eds.) Diagrams 2004. LNCS (LNAI), vol. 2980, pp. 1–19. Springer, Heidelberg (2004)
5. Trickett, S., Trafton, G.: Toward a comprehensive model of graph comprehension: Making the case for spatial cognition. In: Barker-Plummer, D., Cox, R., Swoboda, N. (eds.) Diagrams 2006. LNCS (LNAI), vol. 4045, pp. 286–300. Springer, Heidelberg (2006)
6. Croxton, F., Stein, H.: Graphic comparisons by bars, squares, circles, and cubes. JASA 27, 54–60 (1932)
7. Stewart, B.M., Cipolla, J., Best, L.A.: Extraneous information and graph comprehension: Implications for effective design choices. CWIS 26, 191–200 (2009)
8. Simkin, D., Hastie, R.: An Information-Processing Analysis of Graph Perception. JASA 82, 454–465 (1987)

# Does Manipulating Molecular Models Promote Representation Translation of Diagrams in Chemistry?

Andrew T. Stull[1], Mary Hegarty[1], Mike Stieff[2], and Bonnie Dixon[3]

[1] Department of Psychology,
University of California
Santa Barbara, CA 93106, USA
{stull,hegarty}@psych.ucsb.edu
[2] Department of Curriculum and Instruction
[3] Department of Chemistry and Biochemistry,
University of Maryland
College Park, MD 20742, USA
{mstieff,bdixon1}@umd.edu

**Abstract.** Chemists use many different types of diagrams to represent molecules and must develop skills to accurately translate between such diagrams. Translating between such diagrams can potentially involve the intermediate step of forming an internal 3-d representation of the molecule, so we hypothesized that performance would be enhanced when concrete models were used. Thirty students were provided with models as they translated one molecular diagram into a second and their spontaneous use of the models was recorded. Students' model use was coded for behaviors, such as moving, holding, reconfiguring, pointing to, or gesturing about the model. Results showed a great diversity in whether and how students used the models. Although performance on the representational translation task was generally poor, using the models was positively correlated with performance.

**Keywords:** diagrams, representation translation, individual differences, mental representations.

## 1 Introduction

Diagrams and models are powerful tools in chemistry. Theories in chemistry are based on empirical observations about the macroscopic world that are used to infer the sub-microscopic nature of atoms and molecules [1]. Because atoms and molecules are invisible, chemists rely heavily on the ability to create and use mental models, constructed from empirical evidence and abstracted through the manipulation of external representations, such as molecular diagrams, concrete models and chemical formulas [2]. Diagrams have been and continue to be essential tools, and even a defining characteristic, of chemists.

Chemists use a variety of different diagrams to symbolize and describe the molecules they study [3]. Figure 1 shows three common diagrams, which are used to show the three-dimensional nature of a molecule in the two dimensions of the page. Modeling is no less important as a tool in chemistry where molecular models are commonly

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 338–344, 2010.

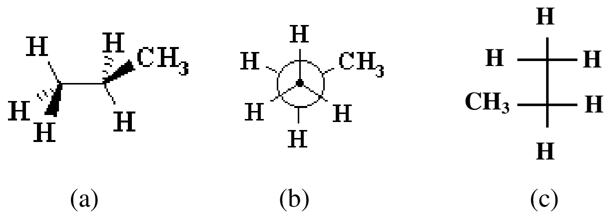


**Fig. 1.** (a) Dash-Wedge diagram, (b) Newman projection and (c) Fischer projection of propane

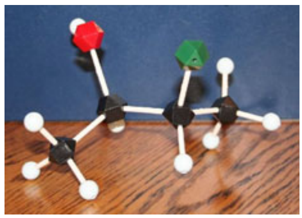


**Fig. 2.** A ball-and-stick form of a concrete model. Color is used to denote different atoms.

used to help scientists design new drugs and materials [4]. A typical three-dimensional representations is the ball and stick model, shown in Figure 2**.** This type of model can be a tangible, real object that exists in real three-dimensional space or a virtual model represented on a computer screen, which can be manipulated using some interface, such as a computer mouse or other input device.

The wide variety of representations employed by chemists in professional and academic settings have resulted from the historical invention of unique diagrams for solving specific conceptual challenges facing the community of chemical scientists. While any one of the chemical representations available to students can be used to represent any given molecule, each representation was created for a specific purpose. In 1891, Emil Fischer created the now ubiquitous "Fischer Projection" to depict all possible spatial configurations of carbohydrates [5]. Similarly, Melvin Newman [6] created the Newman projection to illustrate that any given molecule could assume unique spatial conformations that resulted from rotating specific bonds. As Fischer had, Newman used his new representation to show the community that changes in the spatial conformation of a molecule produced strain on chemical bonds that could alter the thermodynamics of a chemical reaction.

Given the important role of diagrams and molecular models for professional scientists, it is not surprising that these tools are commonly employed in teaching chemistry [7]. These external representations are thought to enhance the ability of the learner to construct mental representations of the molecular structure [8]. With a coherent mental representation of a molecule's structure, students are better able to discover the molecule's functional and behavioral characteristics, which are required to predict its reactive nature [9]. To be successful, chemistry students must master these diagrams.

To understand how diagrams and models operate as representations in chemistry, we must specify the represented world (i.e., the referent, in our case, a molecule), the representing world (i.e., the thing that represents, in this case a diagram or model) and the mapping between these two worlds [10]. We must also specify what aspects of the represented world are explicit in the representation and which are not, because all representations abstract and simplify to some extent and thus represent, or emphasize, some aspects of the information while omitting others. In the case of molecules, the represented world is something tiny that we never see. Although, atoms are made up of subatomic particles that are constantly in motion this is not represented in molecular models, which often represent atoms as solid balls, an example of abstraction. Similarly, the bonds between atoms involve constantly moving electrons, but are simplified in some concrete models as solid sticks. Different chemistry diagrams abstract further from the reality they represent (for example use letters to represent molecules or groups of molecules), show views of models from different perspectives and introduce conventions to depict three dimensional objects in the two dimensions of the printed page. An example is the dash-wedge diagram, in which dashes indicate bonds that extend behind the plane of the page whereas wedges indicate bonds that extend towards the viewer. By representing some aspects of the referent (in this case a molecule), but not others, different diagrams or models make different aspects of molecules explicit or salient in the external representation, while omitting other information [11].

Learning to use representations in chemistry involves learning the conventions of each diagram or model and what is and is not explicitly represented in each. Because chemical diagrams and models are abstractions of reality, they are potentially misleading [12]. We know of no well controlled empirical research that has tested the suggested cognitive benefits of 3-d aids, such as concrete models, when performing translations of 2-d diagrams. Although mastering these representations is central to success in organic chemistry, instructors do not typically spend a lot of class time introducing or comparing the different representations. Furthermore, not all instructors use models in their teaching and instructors differ in whether they encourage their students to use a model kit or the computer models that are available on-line with textbook packages.

A study of the role of diagrams and models in chemistry is an important avenue of research. This study focused on four types of chemistry representations that students have to master in basic courses in organic chemistry; Fisher Diagrams, Newman Diagrams, Dash-Wedge Diagrams, and concrete Ball and Stick Models. These diagrams represent the 3-d structure of the molecule with different graphical conventions and from different viewing perspectives. For example, the Dash-Wedge projection illustrated in Figure 1 shows a 4 carbon molecule from a side perspective, with the line of sight perpendicular to the carbon backbone. The same molecule, as shown in the Newman projection, is represented at an orientation that is a 90° vertical-axis rotation from the dash-wedge projection, with the line of sight parallel to the carbon backbone. In the Fischer projection, the carbon backbone is vertical with atoms on horizontal lines above the plane of the page, atoms at the ends of the vertical lines below the plane of the page, and atoms (not shown) at the intersections of the horizontal and vertical lines in the plane of the page.

Because of the inherent 3-d nature of the molecules represented by the diagrams and the challenge in translating between such diagrams, we hypothesized that students would actively use 3-d concrete models, such as that illustrated in Figure 2, to help them translate between the 2-d diagrams, and that this in turn would lead to improved accuracy on the translation tasks. Specifically, by rotating a physical model and observing the results, a student can be relieved of the need to perform difficult internal spatial transformations (mental rotation or perspective taking). Rotating or otherwise transforming a representation of a molecule is therefore an example of a *complementary* action, that is, an action performed in the world that relieves the individual of the need to perform a mental computation [13, 14].

We conducted a correlational study to examine how students spontaneously use concrete models while translating between different molecular diagrams, and how this spontaneous use is correlated with task performance. The study addressed the following questions: Do students spontaneously use concrete models to help them translate molecular diagrams? How do students use concrete models? Is use of concrete models correlated with drawing accuracy when translating between molecular diagrams?

## 2 Method

**Participants.** Participants were 30 students (12 male, 17 female) who were currently or previously enrolled in an introductory organic chemistry class that did not emphasize the use of models.

**Materials and Procedure.** The task was to translate a 2-d diagram into a second 2-d diagram. As illustrated in Figure 1, the three diagrams used in this study were Dash-Wedge, Newman, and Fischer. Diagrams for six different 4- and 5-carbon straight chain molecules were each presented twice for translation into two different projections to create 12 unique problems. Each problem was displayed separately on an 8.5" x 11" sheet of paper. For each problem, a concrete ball-and-stick model (see Fig. 2) of the molecule was placed on the table in front of the participant. The concrete model was always placed with its long axis perpendicular (wide face) to the observer's line of sight in order to minimize viewing occlusions, but otherwise placement was random.

During the trials, participants were neither encouraged nor hindered from viewing, moving, or manipulating the concrete models. Participants were videotaped with their permission and their behaviors were coded for analysis. Videotapes were coded for behaviors such as moving the model from its original position (move), holding the model in a specific orientation (hold), or reconfiguring the concrete models (reconfigure). In addition, deictic gestures toward the model (deictic gestures) or representational gestures in which the hands represented the molecule (representational gestures) were coded. Drawings of the molecular projections were analyzed for accuracy.

## 3 Results

As Table 1 shows, there were large individual differences in use of the models.

**Table 1.** Incidences of actions toward the concrete model and diagram during representation translation

| Action | Number of participants who performed this action (out of 30) | Number of trials on which this action occurs (out of 360) |
|---|---|---|
| Move the model | 14 (47%) | 93 (26%) |
| Hold the model | 10 (33%) | 52 (15%) |
| Reconfigure the model | 5 (17%) | 23 (6%) |
| Deictic gesture towards model | 7 (23%) | 23 (6%) |
| Deictic gesture towards diagram | 15 (50%) | 87 (24%) |
| Representational gesture | 5 (17%) | 6 (2%) |

**Did students spontaneously use concrete models?** Students varied in whether and how much they spontaneously used the models. During the 12 translation problems, 14 of the 30 participants moved the models at least once. Eight participants moved the models on at least half of the trials, but only one participant moved the models on every problem. Sixteen participants (53%) performed no actions with or about the provided models. A few participants reported that they did not think that moving the models would be helpful. Others reported that they did not look to the model for help because they did not need to, did not understand how it could be helpful, or did not wish to depend on models because models would not be provided on their exams.

**How did students use concrete models?** Other than changing the orientation of the model on the table (move), ten (33%) participants picked-up and held the model in a specific orientation (hold). Of these, only three held the model on more than half of the trials. On almost all (95%) of trials on which this behavior occurred, participants held the model in the orientation that matched that of the diagram to be produced.

Five of the 30 participants (see Table 1) reconfigured the model, that is, rotated an atom or subgroup around a bond, such that the subgroup changed position. This behavior is most useful when translating between a Fischer diagram and one of the other two diagrams, because the Fisher diagram uses a different convention to represent the relative orientation of the carbons than the other diagrams and the given model. Participants made such adjustments more often when translating between Dash-Wedge and Fischer diagrams (48%) and between Newman and Fischer projections (30%) than between Dash-Wedge and Newman projections (22%).

A third behavior, was to point a finger or a pencil (deictic gestures) to specific locations on the concrete model (6% of trials) or on the diagram (24% of trials). Finally, on 2% of trials participants used representational gestures in which they used their hands to represent the molecule. These participants reported that the concrete models were too complex and they could perform the necessary manipulations more easily by gesturing with their hands. Some participants also held their hands over substructures of the molecule and when queried afterward, reported that they were trying to identify the carbon backbone. It is possible that these participants were addressing the complexity of the models by isolating relevant parts while also occluding parts that were not represented in the diagram.

**Is use of concrete models correlated with drawing accuracy?** On transformation accuracy, the average score was 3.77 out of a possible 12 correct diagrams ($SD = 3.03$, $N = 30$). Drawing accuracy was significantly correlated with the number of times a participant moved the model ($r = .49$, $p = .01$). The correlation of drawing accuracy with the number of times a model was held in the orientation that matched the diagram was significant ($r = .39$, $p = .04$). Further, the number of times a participant reconfigured the model was significantly correlated with accuracy ($r = .50$, $p = .01$).

## 4 Discussion

This study makes it clear that it is important to systematically study the use of models in chemistry education. Students had relatively poor performance in translations between the different diagrammatic representations. Contrary to expectations, few participants actively engaged in the use of concrete models in this translation task, although the results suggest that using models improved performance on this task. The low incidence of model use in this study is especially vexing when considering the perceived importance of concrete models for professional scientists. The general lack of model use by many participants and the diversity of spontaneous use observed in this study suggest that concrete models, although heavily employed, are an understudied educational resource in the arena of chemistry education.

In post-task informal interviews, some students reported that they were not sure if they were allowed to pick-up the models during the task. We did not emphasize this in the current study, because we wished to observe purely spontaneous use, but we will emphasize it in future studies. A few of these students also reported that they did not ask to pick-up the models because they did not think that it was necessary or helpful. Others reported that because they had not used models in their class, they did not understand which colors represented which atoms. This suggests the possibility that a subset of the participants did not fully understand how the models related to the diagrammatic translation task. Although all of the students reported familiarity with the three 2-d projections used in the study, the instructors of the participants solicited for this study emphasized the Dash-Wedge projection more than the Fischer and Newman projections. The imbalance of familiarity by students may have had an influence over the results. Another explanation of our results is that students do not use 3-D models because they do not think of the diagrams as representing 3-D structures, and instead have developed heuristics for translating between the different 2-D diagrams without visualizing the 3-D structure that they represent [15].

The implications of this study for chemistry education is that providing models without providing model training may not be helpful for all students. Although some students actively engaged with the models and although use was correlated with drawing accuracy, a large percentage of students did not use the models. Lack of use may have been due to the judgment that they were not needed by some students but other students may have been unable to use the models.

The positive correlations between some of the use behaviors and drawing accuracy suggest that active engagement with concrete models helps with the diagrammatic translation task. In follow-up studies, we plan to compare the benefit of actively manipulating, active reconfiguration, and simple viewing of the models. In addition, because of the growing prevalence of virtual models, future studies will compare concrete and virtual models. Finally, in order to better understand students' difficulty with this task, future research will analyze and characterize the errors made by participants in order to develop intervention strategies.

## References

1. Gilbert, J.K.: Visualization: An emergent field of practice and enquiry in science education. In: Gilbert, J.K., Reiner, M., Nakhleh, M. (eds.) Visualization: Theory and Practice in Science Education, Springer, Dordrecht (2008)
2. Hoffmann, R., Laszlo, P.: Representation in chemistry. Angewandte Chemie International Edition 30, 1–16 (1991)
3. Cheng, M., Gilbert, J.K.: Towards a better utilization of diagrams in research into the use of representative levels of chemical education. In: Gilbert, J.K., Treagust, D. (eds.) Multiple Representations in Chemical Education: Models and Modeling in Science Education, Springer, Dordrecht (2009)
4. Gillet, A., Sanner, M., Stoffler, D., Olson, A.: Tangible interfaces for structural molecular biology. Structure 13, 483–491 (2005)
5. Lichtenthaler, F.W.: Emil Fischer's Proof of the Configuration of Sugars. A Centennial Tribute 31, 1541–1556 (1997)
6. Newman, M.: A useful notation for visualizing certain stereospecific reactions. Record of Chemical Progress 13, 111 (1952)
7. Dori, Y.J., Barak, M.: Virtual and physical molecular modeling: Fostering model perception and spatial understanding. Educational Technology & Society 4, 61–74 (2001)
8. Justi, R., Gilbert, J.K., Ferreira, P.F.M.: The application of a 'model of modeling' to illustrate their importance of metavisualization in respect of the three types of representation. In: Gilbert, J.K., Treagust, D. (eds.) Multiple Representations in Chemical Education: Models and Modeling in Science Education, Springer, Dordrecht (2009)
9. Khan, S.: Model-based inquiries in chemistry. Science Education 91, 877–905 (2007)
10. Palmer, S.E.: Fundamental aspects of cognitive representation. In: Rosch, E.E., Lloyd, B.B. (eds.) Cognition and Categorization, pp. 259–303. Erlbaum, Mahwah (1978)
11. Ealy, J.B.: Students' understanding is enhanced through molecular modeling. Journal of Science Education and Technology 13, 461–471 (2004)
12. Bent, H.A.: Uses (and abuses) of models in teaching chemistry. Journal of Chemical Education 61, 774–777 (1984)
13. Kirsh, D.: Interactivity and multimedia interfaces. Instructional Science 25, 79–96 (1997)
14. Keehner, M., Hegarty, M., Cohen, C.A., Khooshabeh, P., Montello, D.R.: Spatial reasoning with external visualizations: What matters is what you see, not whether you interact. Cognitive Science 32, 1099–1132 (2008)
15. Stieff, M.: Mental rotation and diagrammatic reasoning in science. Learning and Instruction 17, 219–234 (2007)

# Heterogeneous Reasoning in Real Arithmetic

Matej Urbas and Mateja Jamnik

University of Cambridge
{Matej.Urbas,Mateja.Jamnik}@cl.cam.ac.uk

**Abstract.** Diagrams often complement sentential proofs in mathematics. However, diagrams are rarely used as standalone reasoning tools. Thus we propose to integrate diagrammatic reasoning with an existing sentential theorem prover, thus enabling so-called heterogeneous reasoning, particularly in real arithmetic. We will study a set of diagrammatic proof examples from which we will construct a diagrammatic language, inference rules and communication procedures between the diagrammatic and sentential reasoners. The resulting framework will allow the use of diagrammatic proof steps in the same way as the sentential ones, all within the same attempt to construct a proof.

## 1 Introduction

Most diagrammatic reasoning approaches are strictly informal (e.g., sketches or specific illustrations of a general problem). This lead to numerous diagrammatic formalisation efforts [2,3,5]. However, proofs "on paper" rarely consist exclusively of drawings. Diagrams are often accompanied by sentential formulae. This motivated some to investigate heterogeneous reasoning [1,8].

Our goal is to introduce diagrammatic reasoning techniques into an *existing* sentential theorem prover, thus devising a heterogeneous reasoner. We first study heterogeneous proof examples[1] in real arithmetic, from which we will then construct diagrammatic inference rules and language. Finally we will integrate the two modes of reasoning – the diagrammatic logic into the sentential prover.

One of our goals is to show whether heterogeneous reasoning can improve proof intuitiveness in sentential provers. We also believe that heterogeneous methods can provide better or entirely novel proof hints. Hints in homogeneous sentential systems are provided in residual statements of an unsuccessful proof attempt. The unresolved statements can be inspected for clues on how to proceed [4]. However, such hints are often not easily discernible even for experts.

Additionally, naive general inferences from specific diagrams can result in incorrect conclusions.[2] Thus it is essential to provide a suitable diagrammatic formalism. Our aims can be broken down into several sub-goals:

[1] Examples were taken from Nelsen's *Proofs without words* [7].

[2] A famous example is Cauchy's erroneous proof of the Euler characteristic for all polyhedra [6] This "proof" remained unchallenged for decades.

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 345–348, 2010.

**Diagrammatic formalisation:** Introduce diagrammatic inference rules for our logic, check soundness and ensure that the language is powerful enough to cover a sufficiently large subset of problems in the target domain.

**Diagrammatic reasoner:** Construct a diagrammatic reasoner that can either prove a goal or produce a transformed one, which can again be used in the sentential reasoner or act as a hint.

**Integration:** We have to establish a bidirectional translation between the two modes of reasoning and integrate the diagrammatic reasoner into the chosen sentential theorem prover. The truth of all statements must be preserved during the translation. Also, integration must allow not only diagrammatic proof steps, but also conversion of theorems and statements between the two realms.

## 2 Heterogeneous Reasoning

We currently identify three types of heterogeneous interactions (non-exhaustive, based on our proof examples):

**1.) Statement transform:** Diagrammatic transformations are used to rewrite a sentential statement into some other equivalent statement (Fig. 1).

**2.) Introduction of new goals:** Diagrammatic transformations introduce a set of new statements or lemmas (Fig. 2).

**3.) Proving lemmas:** Entire lemmas can be proved with diagrammatic methods. An example is the proof of $\lim_{n\to\infty} \sum_{i=1}^{n} \frac{1}{2^i} = 1$. Fig. 3 illustrates a possible lemma for this theorem, which is proved diagrammatically. Afterwards we use the sentential reasoner to prove $\lim_{n\to\infty} \frac{1}{2^n} = 0$. This last rule is then used to eliminate $\mathrm{B} = \frac{1}{2^n}$ in the diagrammatic proof.

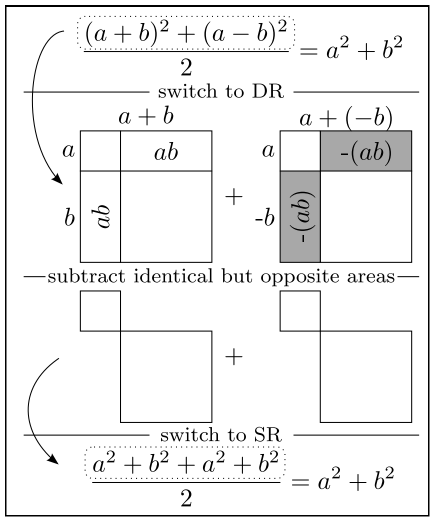



**Fig. 1.** Diagrammatic statement rewrite. Gray denotes the opposite sign of the area – same size areas of opposite sign cancel each-other when combined

Our target domain is the field of real arithmetic, that is formulae from the ordered field $[\mathbb{R}, +, \cdot, 0, 1, <]$. Numbers and variables are represented as edges and rectangular areas. Areas also act as multiplication of edges. Summation is represented by multiple areas and connected edges that extend in the same direction. Also, we use gray to denote the sign of objects. Universal quantification is implicit in the diagram for all variables.

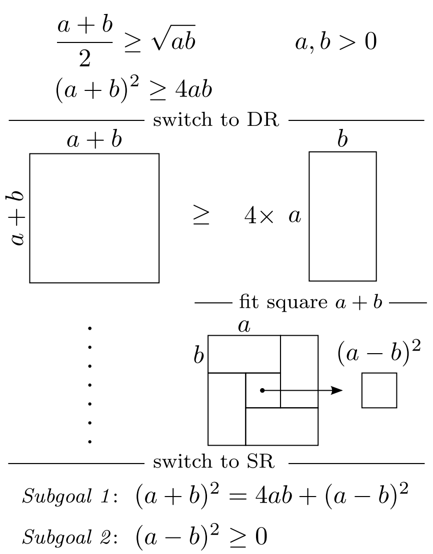



**Fig. 2.** Diagrammatic introduction of new sub-goals

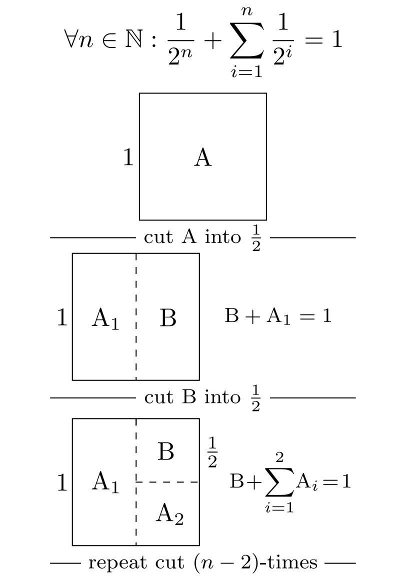



**Fig. 3.** Diagrammatic proof of a lemma

## 3 Methodology

In order to devise a framework with which we can construct heterogeneous proofs as the ones described above, we need to complete the following tasks:

• Define a precise description of the diagrammatic language and its formal inference rules. We will study several examples to determine the required features of the language and the set of inference rules.

• Secondly, we will study the reasoning and theory formalisms in the sentential reasoner. With this, we will determine how the logic of the reasoner influences our diagrammatic language. Because of local expertise, we chose Isabelle [9] as the underlying interactive prover. Initially, the user will choose when to switch between the two modes. Later, we will examine the automation of this choice.

• In the last phase, we will design a communication link between the diagrammatic and symbolic representations. We have chosen a heterogeneous framework architecture where the diagrammatic tactics and statements represent extensions to the built-in native symbolic set of instructions in Isabelle. This will require translation or reuse of internal structures of Isabelle.

In summary, there are many ways in which heterogeneous reasoning can complement sentential approaches, e.g.: more intuitive proofs and proof hints, novel proof tactics, and greater expressive power. We believe that extending a sentential theorem prover with diagrammatic reasoning is viable and advantageous.

## References


1. Barwise, J., Etchemendy, J.: A Computational Architecture for Heterogeneous Reasoning. In: TARK, pp. 1–11. Morgan Kaufmann, San Francisco (1998)
2. Hammer, E.: Reasoning with Sentences and Diagrams. NDJFL 35(1), 73–87 (1994)
3. Howse, J., Stapleton, G.: Visual Mathematics: Diagrammatic Formalization and Proof. In: Autexier, S., Campbell, J., Rubio, J., Sorge, V., Suzuki, M., Wiedijk, F. (eds.) AISC 2008, Calculemus 2008, and MKM 2008. LNCS (LNAI), vol. 5144, pp. 478–493. Springer, Heidelberg (2008)
4. Ireland, A., Jackson, M., Reid, G.: Interactive Proof Critics. JFAC 11(3), 302–325 (1999)
5. Jamnik, M., Bundy, A., Green, I.: On Automating Diagrammatic Proofs of Arithmetic Arguments. JOLLI 8(3), 297–321 (1999)
6. Lakatos, I.: Proofs and Refutations: The Logic of Mathematical Discovery (1976)
7. Nelsen, R.B.: Proofs without Words: Exercises in Visual Thinking (1997)
8. Shin, S.-J.: Heterogeneous Reasoning and its Logic. BSL 10(1), 86–106 (2004)
9. Wenzel, M., Paulson, L.C., Nipkow, T.: The Isabelle Framework. In: Mohamed, O.A., Muñoz, C., Tahar, S. (eds.) TPHOLs 2008. LNCS, vol. 5170, pp. 33–38. Springer, Heidelberg (2008)

# “The Molecules are Inside the Atoms”: Students’ Personal External Representations of Matter

Jessica K. Weller and Mary B. Nakhleh

Purdue University, Department of Curriculum and Instruction, 100 N. University St.
West Lafayette, IN, USA
wellerj@purdue.edu, mnakhleh@purdue.edu

**Abstract.** An understanding of students’ external representations of their internal representations of matter is important for the further development of visual representations used as teaching tools. Therefore, we examined undergraduate students’ personal external representations of matter. Fourteen undergraduates enrolled in general chemistry courses were interviewed about their drawings of matter in the solid, liquid, and gas phases. Data indicated that students spontaneously drew four different types of representations. We have developed rubrics for our classifications of students’ representations based on their drawings. Implications of visual representations in teaching will also be discussed.

**Keywords:** Nature of matter, Representation, Chemistry.

## 1 Introduction

Research in chemical education has documented student misconceptions about the particulate nature of matter from elementary students through college students [1], [2], [3]. Some researchers have even argued that visual representations such as the animations, diagrams, and drawings used in the teaching and learning of chemistry may contribute to the types of misconceptions students hold [4]. For example, visual representations of the nature of matter can be created at three distinct levels that students must master. These levels are the macroscopic observable world, the molecular world of atoms and molecules, and the symbolic world of equations and symbols [5]. Students also need to be able to move between the three levels of representation to be able to explain why an observable property such as melting occurs at the molecular level. In addition, students need to know the conventional symbols used to represent this phenomenon in chemistry language. Clearly, confusion at one level may translate to non-understanding at one or more of the other levels.

### 1.1 Significance of the Study

Our students’ personal external representations of the particle nature of matter provided insight into their understanding of these concepts. We studied these representations in terms of chemistry content and organization of the ideas. This analysis indicated the types of representations that students tended to use when constructing

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 349–351, 2010.

their conceptual frameworks about the particulate nature of matter. From these representations we were able to detect misconceptions in students' conceptual understandings of the nature of matter. We hope that our work will help researchers and teachers create accurate instructional representations of the particulate nature of matter that integrate the three levels of representation used in chemistry.

### 1.2 Theoretical Perspective

Our theoretical perspective for this research was based in constructivism. Students have formed conceptual understandings about the nature of matter. Their understandings may or may not align with the scientifically accepted paradigm of the particulate nature of matter. However, through instruction and interactions with more knowledgeable peers, teaching assistants and professors, students can adapt and develop their understandings to align with the scientifically accepted version. Having students create their own personal external representations of their conceptual understandings and analyzing those representations may help students in their process of conceptual change and development.

## 2 Data Collection Methods

We conducted a qualitative exploratory study with undergraduate students from a major Midwestern University. Students were recruited from several different introductory general chemistry courses. We included students with different levels of expertise and interest in chemistry and science to construct a more complete understanding of undergraduate students' beliefs about the nature of matter. Fourteen students participated in our study. Seven of those were taking chemistry courses geared toward a science or engineering major. Of these seven, four were male and three were female. The other seven students were taking a chemistry course designed for elementary education majors. All of these students were female.

The first author individually interviewed the students using a semi-structured guide created by Nakhleh and Samarapungavan [1].The interview guide consisted of three sequences. Sequence A consisted of questions that asked students to describe properties of pure substances after the substance was shown to them. Each phase of matter was represented by the different substances. Solids were represented by sugar cubes, wooden toothpicks, copper wires and ice cubes. Liquid was represented by water and gas was represented by a clear helium filled balloon. Sequence B consisted of questions about processes the above pure substances could undergo. Specifically we were concerned with fluidity, rigidity, and malleability. Sequence C focused on phase changes of water and the dissolution of table salt into water.

The semi-structured nature of the guide allowed the first author to probe students with additional questions to gain better understandings of participants' use of terms such as atoms, molecules, and bonding. The questions were open-ended and allowed us to probe participants' understanding at the macroscopic observable level, the molecular level, and symbolic level. We asked descriptive questions that directed participants to describe a substance. We were interested in both their initial spontaneous response as well as their responses to further probes. We also asked explanatory questions that were designed to illicit participants' understandings of phenomena such as ice melting. Again, we were interested in both their initial spontaneous responses as well as their responses to probes.

Before the interview, each participant was given a set of labeled drawing sheets. When a student described an object or process, they were asked to draw what they were trying to explain. The first author used the drawings to illicit further verbal explanations about the students' beliefs regarding the nature of matter. The drawing sheets were collected by the first author at the end of the interview. All interviews were audio-taped and lasted between 45 minutes and 2 hours.

## 3 Analysis and Results

All interviews were transcribed and coding schemes were developed for both the interviews and the drawings. An inter-rater reliability of 80% was obtained. Our initial phase of analysis indicated that students spontaneously drew two types of macroscopic representations; those that indicated particles and those that did not. In addition to the macroscopic representations, students drew mixtures of the macroscopic and molecular levels as well as purely molecular level representations. Some students also used the symbolic level in their representations. Further analysis of our initial results indicated that students' representations could be placed along a continuum of Novice to Expert Level 1.We believe these personal external representations to be representative of students' conceptual frameworks of the nature of matter.

## References

1. Nakhleh, M.B., Samarapungavan, A.: Elementary School Children's Beliefs about the Nature of Matter. J. Res. Sci. Teach. 36, 777–805 (1999)
2. Griffith, A.K., Preston, K.R.: Grade 12 Students Misconceptions Relating to Fundamental Characteristics of Atoms and Molecules. J. Res. Sci. Teach. 29, 611–628 (1992)
3. Williamson, V.M., Abraham, M.R.: The Effects of Computer Animation on the Particulate Mental Models of College Chemistry Students. J. Res. Sci. Teach. 32, 521–534 (1995)
4. Gilbert, J.K., Treagust, D.: Multiple Representations in Chemical Education. Springer, New York (2009)
5. Kozma, R., Russell, J.: Students Becoming Chemists: Developing Representational Competence. In: Gilbert, J.K. (ed.) Visualization in Science Education, pp. 121–145. Springer, Dordrecht (2005)

# Discovering Perceptions of Personal Social Networks through Diagrams

Lixiu Yu[1], Jeffrey V. Nickerson[1], and Barbara Tversky[2]

[1] Stevens Institute of Technology, Hoboken, NJ
[2] Teachers College, Columbia University, New York, NY
lyu3@stevens.edu, jnickerson@stevens.edu, btversky@stanford.edu

**Abstract.** By examining diagrams created by study participants, we can gain insight into their perceptions of their personal social networks. In this study, we found that participants made use of both position and distance to differentiate the roles of those in their networks and express intimacy. This work has implication for both the elicitation and visualization of social networks.

**Keywords:** Diagram understanding, personal social networks.

## 1 Introduction and Methods

Abstract diagrams such as networks are interesting to study for several reasons. Although abstract diagrams contain a minimum of depictive information, they take advantage of spatial reasoning processes that verbal, descriptive representations do not normally afford. Observers can follow the lines from node to node to assess relationships, temporal, social, causal, or more. Normally, these relations are not in and of themselves spatial, but rather metaphorically spatial, a mapping to diagrammatic space that even preschoolers can do [1].

People add to diagrams inessential spatial information, even metaphoric spatial information. In a set of studies, students in classes in design of information systems were asked to sketch their designs of the interrelationships of the components, computers, cell phones, satellites, trucks, buildings, and the like [2]. All that is needed is labeled boxes and lines in arbitrary locations. Nevertheless, students used location and proximity in the space of the page to convey inessential information.

Including relevant, even if inessential information, may help users express, understand, and make inferences from diagrams. The inessential information that people add to their diagrams serves another role, not for the producers and users of diagrams, but for researchers. That information can reveal how people think about the concepts and relations conveyed in the diagrams. An especially interesting context for using diagrammatic productions to reveal thought is social network diagrams. A variety of social relationships, notably agency (e. g., [3]), are thought of in spatial terms. In particular, more power is mapped to higher and greater agency is mapped leftwards in languages that are read from left to right. Will people use spatial location and distance in social networks to express more than simple connections?

In the research to be reported, respondents are asked to produce social networks or to select appropriate networks. Because position and proximity in space are heavy

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 352–354, 2010.

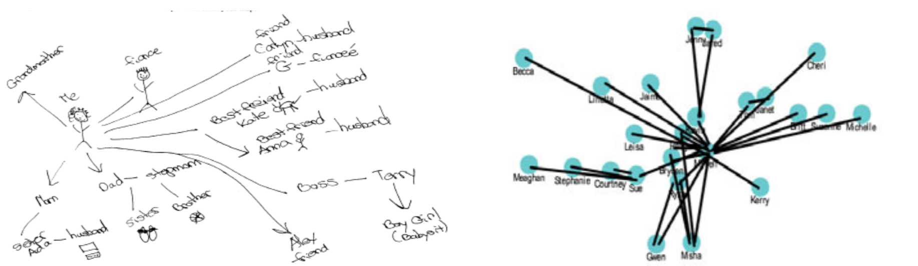
with metaphoric meanings, we expect that proximity on the page will be used to group closely related groups of alters, that is, to convey intimacy.

We collected data using two techniques: paper and pencil, and online sketching. The first (paper-and-pencil) experiment was performed with 35 face-to-face participants. Twenty-five participants were graduate students; the rest were recruited at public places in an urban area and came from many walks of life.

They were asked: *Please draw your personal social network, and try to include the following people (if applicable): best friends, someone whose name you barely know, children, boss, mother, boyfriend/girlfriend/spouse, sibling, someone you don't like, father and distant friends.* They were provided with a pen and an 11 x 17 inch piece of paper for drawing their social networks. In the second experiment, 39 participants used a simple online drawing applet to create diagrams in response to the above question. The applet provided menu choices for the drawing of lines, circles, and rectangles, as well as the ability to label the components of the diagram. The participants ranged in age from 18 to 50, and 17 participants were female.

## 2 Results and Discussion

Superficially, the paper and screen-based sketches look quite different. In particular, participants were more likely to sketch recognizable icons with a pen than with a mouse, as shown in Fig. 1 below. However, structurally, the two sets of diagrams were similar. There were no significant differences between the numbers of nodes drawn or the topologies, so the data sets were combined for analysis.




(a) (b)

**Fig. 1.** (a) A paper-and-pen diagram (b) An online diagram

The research on spatial social schemas suggests that closer relations should be closer, that prior generations should be higher, and that greater power should be higher. Thus, parents should be mapped above ego, and spouses and parents should be mapped closer than acquaintances and especially antagonists. In order to examine these predictions, diagrams were centered on the ego; the positions of the roles are shown in Fig. 2. We see that mothers ($p<0.01$), fathers ($p<0.01$) and spouses ($p<0.05$) are usually positioned above the ego, but antagonists, landlords, acquaintances are not. We next examined the distances to the ego, and results are plotted in Fig. 3. As we expected, the average distances of mothers, fathers and spouses to the ego are significantly shorter than those of the other roles to the ego (*p<0.01*). One participant explained this: "*My graph is more like concentric circles, where the farther you get away from the center (Me), the less intimate my friends are"*. Distance, at least for this participant, is a reflection of intimacy.

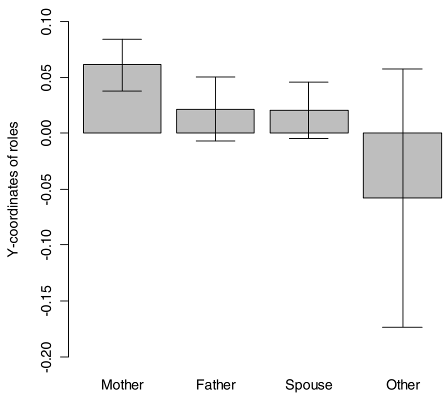



**Fig. 2.** Means and standard errors of the y coordinates of roles, after recentering

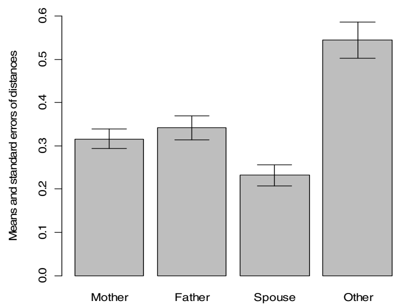



**Fig. 3.** Means and standard errors of distances between ego and alter by role

Abstract diagrams convey the intended information clearly without the clutter and distraction of unnecessary information. But people think concretely, so that including unnecessary but meaningful information may support reasoning. That this might be so is indicated by previous findings as well as those reported here. When people produce diagrams, they often spontaneously include meaningful information not specifically requested. Previous research showed that students go to the trouble of drawing pictures of various concrete objects and beings even when there is nothing in the instructions or the task to suggest doing that [2]. The current research extends those findings by showing that people also spontaneously include quite abstract information, information about intimacy, generation, and power, in their diagrammatic productions of social networks.

**Acknowledgments.** Portions of this research were supported by grants from National Science Foundation IIS-0725223, IIS-0855995, and REC-0440103, the Stanford Regional Visualization and Analysis Center, and Office of Naval Research NOOO14-PP-1-O649, N000140110717, and N000140210534.

## References


1. Tversky, B., Kugelmass, S., Winter, A.: Cross-cultural and developmental trends in graphic productions. Cognitive Psychology 23, 515–557 (1991)
2. Nickerson, J.V., Corter, J.E., Tversky, B., Zahner, D., Rho, Y.: The Spatial Nature of Thought: Understanding information systems design through diagrams. In: Proceedings of the International Conference on Information Systems (2008)
3. Chatterjee, A.: Language and space: Some interactions. Trends in Cognitive Science 5, 55–61 (2001)

# Erratum: The Effects of Perception of Efficacy and Diagram Construction Skills on Students' Spontaneous Use of Diagrams When Solving Math Word Problems⋆

Yuri Uesaka[1], Emmanuel Manalo[2], and Shin'ichi Ichikawa[3]

[1] Japan Society for the Promotion of Science, Tokyo Institute of Technology
[2] Waseda University
[3] The University of Tokyo
y.uesaka@nm.hum.titech.ac.jp, emmanuel.manalo@gmail.com
ichikawa@p.u-tokyo.ac.jp

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, pp. 197–211, 2010.


**DOI: 10.1007/978-3-642-14600-8_51**

Table 2 and Table 3 of the paper starting on page 197 of this volume are incorrect. Here are the correct versions:

**Table 2.** Ratios of Participants Referring to the Efficacy of Diagram Use in the Free Comments Section of the Post-instruction Survey

| | With PD | Without PD | Totals |
|---|---|---|---|
| With VE | 0.46 (11/24) | 0.42 (8/19) | 0.44 (19/43) |
| Without VE | 0.10 (2/21) | 0.18 (4/22) | 0.14 (6/43) |
| Totals | 0.29 (13/45) | 0.29 (12/41) | 0.29 (25/86) |

*Note*. In parentheses are the number of participants who referred to the efficacy of diagram use, and the total number of participants, in each condition.

**Table 3.** The Total Number of Problems for which Participants Produced High Quality Diagrams When Asked to Use Diagrams in the Post-instruction Assessment

| | With PD | Without PD | Totals |
|---|---|---|---|
| With VE | 2.46 (0.59) | 1.95 (0.91) | 2.23 (0.78) |
| Without VE | 2.19 (0.75) | 1.77 (1.06) | 1.98 (0.94) |
| Totals | 2.46 (0.59) | 1.95 (0.91) | 2.23 (0.78) |

In the original version, the chi-square symbol ($\chi$) had inadvertently become the division sign ($\div$) on pages 205 (at the bottom), 206 and 207 (at the top).

⋆ The original online version for this chapter can be found at: http://dx.doi.org/10.1007/978-3-642-14600-8_19

A.K. Goel, M. Jamnik, and N.H. Narayanan (Eds.): Diagrams 2010, LNAI 6170, p. E1, 2010.

# Author Index