Smart Innovation, Systems and Technologies 42

Jeffrey W. Tweedale
Rui Neves-Silva
Lakhmi C. Jain
Gloria Phillips-Wren
Junzo Watada
Robert J. Howlett *Editors*

Intelligent Decision Technology Support in Practice

KES International

Springer

Smart Innovation, Systems and Technologies

Volume 42

Series editors

Robert J. Howlett, KES International, Shoreham-by-Sea, UK
e-mail: rjhowlett@kesinternational.org

Lakhmi C. Jain, University of Canberra, Canberra, Australia, and
University of South Australia, Adelaide, Australia
e-mail: Lakhmi.jain@unisa.edu.au

About this Series

The Smart Innovation, Systems and Technologies book series encompasses the topics of knowledge, intelligence, innovation and sustainability. The aim of the series is to make available a platform for the publication of books on all aspects of single and multi-disciplinary research on these themes in order to make the latest results available in a readily-accessible form. Volumes on interdisciplinary research combining two or more of these areas is particularly sought.

The series covers systems and paradigms that employ knowledge and intelligence in a broad sense. Its scope is systems having embedded knowledge and intelligence, which may be applied to the solution of world problems in industry, the environment and the community. It also focusses on the knowledge-transfer methodologies and innovation strategies employed to make this happen effectively. The combination of intelligent systems tools and a broad range of applications introduces a need for a synergy of disciplines from science, technology, business and the humanities. The series will include conference proceedings, edited collections, monographs, handbooks, reference books, and other relevant types of book in areas of science and technology where smart systems and technologies can offer innovative solutions.

High quality content is an essential feature for all book proposals accepted for the series. It is expected that editors of all accepted volumes will ensure that contributions are subjected to an appropriate level of reviewing process and adhere to KES quality principles.

More information about this series at http://www.springer.com/series/8767

Jeffrey W. Tweedale · Rui Neves-Silva
Lakhmi C. Jain · Gloria Phillips-Wren
Junzo Watada · Robert J. Howlett
Editors

Intelligent Decision Technology Support in Practice

Springer

Editors
Jeffrey W. Tweedale
School of Engineering, KES Centre
University of South Australia
Adelaide, SA
Australia

Rui Neves-Silva
FCT Campus
UNINOVA
Caparica
Portugal

Lakhmi C. Jain
School of Engineering
University of South Australia
Adelaide, SA
Australia

Gloria Phillips-Wren
Sellinger School of Business
and Management
Loyola University Maryland
Baltimore, MD
USA

Junzo Watada
Graduate School of IPS
Waseda University
Fukuoka
Japan

Robert J. Howlett
KES International
Shoreham-by-Sea
UK

ISSN 2190-3018 ISSN 2190-3026 (electronic)
Smart Innovation, Systems and Technologies
ISBN 978-3-319-21208-1 ISBN 978-3-319-21209-8 (eBook)
DOI 10.1007/978-3-319-21209-8

Library of Congress Control Number: 2015945130

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

This book is dedicated to all researchers seeking inspiration on how they can generate practical applications containing Intelligent Decision-making Technology. Its content reflects the unwavering support and innovative success of numerous scientists within the Knowledge-Based, Intelligent Information and Engineering Systems community.

Foreword

The Intelligent Decision-making Technologies (IDT) domain is a fast growing area of research that integrates various aspects of computer science and information systems. This includes intelligent systems, intelligent technology, intelligent agents, artificial intelligence, fuzzy logic, neural networks, machine learning, knowledge discovery, computational intelligence, data science, big data analytics, inference engines, recommender systems or engines, and a variety of related disciplines.

Innovative applications that emerge using IDT often have a significant impact on decision-making processes in government, industry, business, and academia in general. This is particularly pronounced in finance, accounting, healthcare, computer networks, real-time safety monitoring and crisis response systems. Similarly, IDT is commonly used in military decisions-making systems, security, marketing, stock market prediction, and robotics.

This is the main reason why KES International (UK) made a great conceptual decision in 2009 when it organized the annual international conference on Intelligent Decision making Technologies (KES-IDT). The main goal of this professional meeting is to provide scholars, developers and practitioners with an excellent opportunity to meet well-known experts from all over the world and discuss innovative ideas, findings and results of research project, applications, and best practices in emerging intelligent decision making technology area.

The main content of this book is based on research papers and presentations of the 2013 KES-IDT international conference. It covers important IDT topics that include, but not limited to:

- knowledge visualization and reasoning mechanisms to support decision-making process;
- decision support systems and their applications in management;
- neural networks to support data analysis and classification;
- applications of decision-making technologies in scalable and evolvable environments;
- intelligent systems to support social networking and media;
- 3D human-machine interfaces, 3D modeling, and visualization technology;

- Intelligent agents and multi-agent systems;
- collaborative decision making technologies and collaborative systems;
- service-oriented architectures to implement decision making components;
- recommender systems; and
- cognitive models, processes and interfaces.

The book provides a well-thought out and balanced combination of carefully selected papers that cover various challenging theoretical aspects, sounding technological achievements, real-world practical applications, and implementations of intelligent decision making technologies. The chapters deliver practical and quite often complementary problem-solving techniques. The depth and breadth of this ground-breaking content employs numerous IDT tools, techniques, and methodologies to generate innovative applications. Intelligent Decision Technology Support in Practice will undoubtedly be a valuable reference for experts, researchers, and Ph.D. students in the intelligent decision-making technologies domain.

Prof. Vladimir Uskov
Computer Science and Information Systems
Director, InterLabs Research Institute
Bradley University
Peoria, IL, USA

Preface

Intelligent Decision-making Technology (IDT) is a field of study related to gathering, processing, and exploiting data to achieve intelligent tasks. Research is predominantly software based, however, it requires and incorporates technology to build new or innovative tools that ultimately benefit society. Industry continues to invest in IDT, with many creating internal research facilities. Big data is the latest field to attract a significant investment from industry and a lot of research is being targeted to solve real-world problems.

IDT relies on innovative use of artificial intelligence (AI) techniques to employ human-like intelligence to emulate decision-making and solve data intensive problems. Researchers continue to improve existing techniques within the domain. This evolution in information processing and data centric decision-making has become a pervasive phenomenon within the community, especially for mobile computing. Technology is enabling the community to use mobile devices to access information on-the-go, forcing researchers to deliver more intelligent systems. Researchers continue to leverage off recent advances in information technologies and now collaborate across several domains to solve problems. The major threads in the book discuss the generation of strategy, controlling energy, interacting with social media, reasoning and research, or industry inspired applications. Intelligent Systems are becoming ubiquitous in a wide range of situations. These include facets of simple everyday actions on mobile devices through to more advanced enterprise-level applications in numerous domains. Society benefits daily, through digital news, socialization of relations, and enhancements derived from expert decision-making within knowledge-based systems.

All contributions in this book were sourced from authors supporting the series of activities associated with IDT. A call for chapters was issued following the 5th Conference on Intelligent Decision Technologies, held at Sesimbra, Portugal during 26–28 June 2013. Based on the submissions received, select authors were invited to enter a competitive process prior to acceptance. Following an iterative review process, an innovative subset of topics and techniques have been included in this volume. The editors believe these promote or enhance decision-making techniques using AI techniques.

July 2015

Jeffrey W. Tweedale
Lakhmi C. Jain

Acknowledgments

This publication would not be possible without the dedication and innovation of researchers within the field of intelligent decision-making technologies. The editors are proud to present a number of hand selected papers that have been expanded into chapters. These contributions span a number of domains and research environments that solve real-world problems. We must thank a large number of people who have contributed to the success of this endeavor as reviewers of one or more submissions, especially the former papers and final submissions. Similarly, for the efforts of the chairs who organized the 5th KES International Conference on Intelligent Decision Technologies, held in 2013 at Sesimbra, Portugal [1]. The scene for this topic was stimulated through the rich source of contributions to this annual event. The authors were selected based on their initial contributions published by KES. These include the following subjects:

- Stability of Financial Market: The Introduction to the JAVA Simulation of Agent-Based Model [2]
- Approach for decision support for energy savings and emissions trading based on industrial requirements [3]
- Hybrid RBF-ART Model and Its Application to Medical Data Classification [4]
- MobiS—Personalized Mobility Services for energy efficiency and security through advanced Artificial Intelligence techniques [5]
- Advancing collaborative decision making through alternative visualizations and reasoning mechanisms [6]
- Multilingual Agents in a Dynamic Environment [7]
- Defining a Decision-Support Framework in AC^{3}M [8]
- Developing Intelligent Agents with Distributed Computing Middleware [9]
- Rendering of wind effects in 3D landscape scenes [10]
- SOAda: Service-Oriented Architecture with a Decision Aspect [11]
- Method to share responsibility knowledge of dependability cases [12]
- A TV Program Recommender Framework [13]
- Incompleteness and fragmentation in spoken language syntax and its relation to prosody and gesturing: cognitive processes vs. possible formal cues [14]

The continuous evolution of techniques and methodologies are rapidly being adapted to more diverse problems, ultimately enhancing the features in modern-day applications. This culmination of multi-disciplined skills are being employed to create innovative solutions for classifiers, data mining, knowledge management, Intelligent Decision Technology, and a number of models that can be used to solve industrial-level problems. Obviously without the contribution of the founding subject matter experts, these existing researchers may have pursued completely different disciplines. This book provides prospective researchers with a vital source for motivation and help formulate their vision to ultimately produce innovative solutions in the future.

The editors extend special thanks to the chairs of the 5th KES International Conference on Intelligent Decision Technologies for hosting the conference in Sesimbra, Portugal from 26 to 28 June 2013. They include: Neves-Silva, R., Watada, J., Phillips-Wren, G., Jain, L.C., Howlett, R.J. We also wish to acknowledge the support of the UNINOVA research institute for contributing to the success of this symposium.

References

1. Neves-Silva, R., Watada, J., Phillips-Wren, G. (eds.): Intelligent Decision Technologies: Proceedings of the 5th KES International Conference on Intelligent Decision Technologies. IOS Press, Amsterdam (2013)
2. Šperka, R., Spišák, M., Górecki, J.: Stability of financial market: The introduction to the java simulation of agent-based model. In: Neves-Silva, R., Watada, J., Phillips-Wren, G., Jain, L. C., Howlett, R.J., eds.: Intelligent Decision Technologies. Volume 241 of Frontiers in Artificial Intelligence and Applications, pp. 380–389. IOS Press, Amsterdam (2013)
3. Marques, M., Neves-Silva, R.: Approach for decision support for energy savings and emissions trading based on industrial requirements. In: Neves-Silva, R., Watada, J., Phillips-Wren, G., Jain, L.C., Howlett, R.J., eds.: Intelligent Decision Technologies. Volume 255 of Frontiers in Artificial Intelligence and Applications, pp. 74–83. IOS Press, Amsterdam (2013)
4. Tan, S.C., et al.: A hybrid RBF-ART model and its application to medical data classification. In: Neves-Silva, R., Watada, J., Phillips-Wren, G., Jain, L.C., Howlett, R.J., eds.: Intelligent Decision Technologies. Volume 255 of Frontiers in Artificial Intelligence and Applications, pp. 21–30. IOS Press, Amsterdam (2013)
5. Costa, R., et al.: Mobis - personalized mobility services for energy efficiency and security through advanced artificial intelligence techniques. In: Neves-Silva, R., Watada, J., Phillips-Wren, G., Jain, L.C., Howlett, R.J., (eds.): Intelligent Decision Technologies. Volume 241 of Frontiers in Artificial Intelligence and Applications, pp. 296–306. IOS Press, Amsterdam (2013)
6. Christodoulou, S., Karacapilidis, N., Tzagarakis, M.: Advancing collaborative decision making through alternative visualizations and reasoning mechanisms. In: Neves-Silva, R., Watada, J., Phillips-Wren, G., Jain, L.C., Howlett, R.J., (eds.): Intelligent Decision Technologies. Volume 241 of Frontiers in Artificial Intelligence and Applications, pp. 38–47. IOS Press, Amsterdam (2013)

7. Tweedale, J., Jain, L.: Multilingual agents in a dynamic environment. In: Nakamatsu, K., Phillips-Wren, G., Jain, L., Howlett, R., (eds.): New Advances in Intelligent Decision Technologies. Volume 199 of Studies in Computational Intelligence, pp. 193–202. Springer Berlin Heidelberg (2009)
8. Consoli, A., Tweedale, J., Jain, L.: Defining a decision-support framework in AC^3M. In: Nakamatsu, K., Phillips-Wren, G., Jain, L., Howlett, R., (eds.): IDT'09: First KES International Symposium in Intelligent Decision Technologies, pp. 203–212. (2009)
9. Sioutis, C., Dominish, D.: Developing intelligent agents with distributed computing middleware. In: Watada, J., Phillips-Wren, G., Jain, L., Howlett, R., (eds.): Intelligent Decision Technologies. Volume 10 of Smart Innovation, Systems and Technologies, 863–872. Springer Berlin Heidelberg (2011)
10. Favorskaya, M., Tkacheva, A.: Rendering of wind effects in 3d landscape scenes, 17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems. Procedia Comput. Sci. **22**, 1229–1238 (2013)
11. Boumahdi, F., Chalal, R.: Soada: Service oriented architecture with a decision aspect, 17th international conference in knowledge based and intelligent information and engineering systems. Procedia Comput. Sci. **22**, 340–348 (2013)
12. Saruwatari, T., Hoshino, T., Yamamoto, S.: Method to share responsibility knowledge of dependability cases, 17th international conference in knowledge based and intelligent information and engineering systems. Procedia Comput. Sci. **22**, 1073–1082 (2013)
13. Chang, N., Irvan, M., Terano, T.: A TV program recommender framework, 17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems. Procedia Comput. Sci. **22**, 561–570 (2013)
14. Hunyadi, L.: Incompleteness and fragmentation in spoken language syntax and its relation to prosody and gesturing: Cognitive processes vs. possible formal cues. In: Cognitive Infocommunications (CogInfoCom), 2013 IEEE 4th International Conference on, 211–218. December 2013

Contents

Contributors

Fatima Boumahdi LRDSI Laboratory, Sciences Faculty, Saad Dahlab University, Blida, Algeria

Luka Bradesko Jozef Stefan Institute, Jubljana, Slovenia

Rachid Chalal Laboratory LMCS, Algiers, Algeria

Na Chang Tokyo Institute of Technology, Yokohama, Kanagawa, Japan

Spyros Christodoulou Computer Technology Institute & Press “Diophantus”, and University of Patras, Rio Patras, Greece

Angela Consoli Defence Science and Technology Organisation, Edinburgh, Australia

Ruben Costa UNINOVA, Centre of Technology and Systems, Caparica, Portugal

Margarita Favorskaya Siberian State Aerospace University, Krasnoyarsk, Russia

Paulo Figueiras UNINOVA, Centre of Technology and Systems, Caparica, Portugal

Carlos Gutierrez UNINOVA, Centre of Technology and Systems, Caparica, Portugal

T. Hoshino Software Innovation Center NTT, Konan, Minato-ku, Tokyo, Japan

L. Hunyadi University of Debrecen, Debrecen, Hungary

Mhd Irvan Tokyo Institute of Technology, Yokohama, Kanagawa, Japan

Lakhmi C. Jain Faculty of Education, Science, Technology and Mathematics, University of Canberra, Belconnen, Australia

Nikos Karacapilidis Computer Technology Institute & Press “Diophantus” and University of Patras, Rio Patras, Greece

H. Kiss University of Debrecen, Debrecen, Hungary

Chee Peng Lim Centre for Intelligent Systems Research, Deakin University, Geelong, Australia

Maria Marques UNINOVA-FCT Campus, Caparica, Portugal

Rui Neves-Silva FCT-UNL-FCT Campus, Caparica, Portugal

Gloria Phillips-Wren Sellinger School of Business and Management, Loyola University Maryland, Maryland, USA

T. Saruwatari Graduate School of Information Science, Nagoya University, Nagoya, Japan; Software Innovation Center NTT, Konan, Minato-ku, Tokyo, Japan

Christos Sioutis Defence Science and Technology Organisation, Edinburgh, SA, Australia

Roman Šperka School of Business Administration in Karviná, Silesian University in Opava, Opava, Czech Republic

I. Szekrenyes University of Debrecen, Debrecen, Hungary

Shing Chiang Tan Multimedia University, Cyberjaya, Malaysia

Takao Terano Tokyo Institute of Technology, Yokohama, Kanagawa, Japan

Anastasia Tkacheva Siberian State Aerospace University, Krasnoyarsk, Russia

Jeffrey W. Tweedale Aerospace Division, Defence Science and Technology Organization, Adelaide, Australia; School of Engineering, University of South Australia, Adelaide, Australia

Manolis Tzagarakis Computer Technology Institute & Press "Diophantus", and University of Patras, Rio Patras, Greece

Junzo Watada Waseda University, Tokyo, Japan

S. Yamamoto Strategy Office, Information and Communications Headquarters, Nagoya University, Nagoya, Japan

Acronyms

AAF	Agent Architecture Framework
ABMS	Agent-Based Modeling and Simulation
AC^{3}M	Agent Coordination and Cooperation Cognitive Model
ACL	Agent Communication Language
ACT	Australian Capital Territory
ADSS	Adaptive Decision Support System
ADSTG	Automatic Diagnosis System Based on Thyroid Gland
AHP	Analytic Hierarchy Process
AHP	Analytical Hierarchy Processing
AI	Artificial Intelligence
AIP	Advanced Information Processing
AIRS	Artificial Immune Recognition System
AJAX	Asynchronous JavaScript and XML
ALD	Atomic Layer Deposition
ANN	Artificial Neural Network
API	Application Programming Interface
APL	A Programming Language Interface
ARPANET	Advanced Research Projects Agency Network
ART	Adaptive Resonance Theory
ARTMAP	Adaptive Resonance Theory MAP
ATLAS	Agent Transportation Layer Adaption System
ATM	Automatic Teller Machine
AWB	Australian Workplace Barometer
BDI	Beliefs, Desires, Intentions
BISC	Berkeley Initiative in Soft Computing
BPMN	Business Process Management Notation
C^2	Command and Control
C^{4}ISR	Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance
CATI	Computer Aided Telephone Interviews
CBR	Case-Based Reasoning

CCM	CORBA Component Model
CHRIS	Cognitive Hybrid Reasoning Intelligent Agent System
CI	Computational Intelligence
CO-O^2	Coordination and Cooperation
CORBA	Common Object Request Broker Architecture
COT	Computational Organizational Theory
COTS	Commercial Off-The-Shelf
CRUD	Creation, Reading, Updating, and Deleting
CSE	Cognitive Systems Engineering
CWA	Cognitive Work Analysis
DAI	Distributed Artificial Intelligence
DAML	DARPA Agent Markup Language
DAML-OIL	DAML—Ontology Interface Language
DAML-OWL	DAML—Ontology Web Language
DAML-S	DARPA Agent Markup Language—Services
DAML-W	DAML-OWL
DA	Decision Analyzer
DCE	Distributed Computing Environment
DCOM	Distributed Component Object Model
DDA	Dynamic Decay Adjustment
DDAA	Dynamic Decay Adjustment Algorithm
DMoS	Decisional Model of Service
DMS	Data-base Management System
DOC	Distributed Object Computing
DON	Dicode ONtology
DPS	Distributed Problem Solving
DSS	Decision Support System
EC	European Commission
ECC	Energy Cost Centers
EIS	Executive Information System
EOQ	Economic Order Quantity
EP	Evolutionary Programming
ES	Expert System
ESTDD	Expert System for Thyroid Disease Diagnosis
ETS	Emissions Trading Scheme
EURIDICE	Euridice Knowledge Model—FP7 IP
FAM	Fuzzy ARTMAP
FCA	Frequency Clearance Agreement
FIPA	Foundation of Intelligent Physical Agents
FL	Fuzzy Logic
FMM	Fast Marching Method
GA	Genetic Algorithm
GDSS	Group Decision Support Systems
GIS	Geographic Information Systems
GP	Genetic Programming

GPS	Global Positioning System
GSN	Goal Structuring Notation
GTFS	Googles Generic Transit Feed Specifications
GUI	Graphical User Interface
HCI	Human Computer Interaction
HDD	Hard Disk Drive
HM	Harmonic Means
HMI	Human Machine Interface
HTML	Hypertext Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IA	Intelligent Agent
IAF	Intelligent Agent Framework
I-MAS	Intelligent Multi-Agent
ICD	Interface Control Documents
IDC	Intelligent Design Choice
IDE	Integrated Development Environment
IDL	Interface Description Language
IDS	Intelligent Decision Support
IDSS	Intelligent Decision Support System
IDT	Intelligent Decision Technologies
iLand	Individual-based LANdscape and Disturbance model
IMADA	Intelligent Multi-Agent Decision Analyser
IMS	Information Management System
INS	Inertial Navigation System
IoT	Internet of Things
IRR	Internal Rate of Return
IS	Information System
ISR	Intelligence, Surveillance and Reconnaissance
IT	Information Technology
ITS	Intelligent Transport System
JADE	Java Agent Development Framework
JTIDS	Joint Tactical Information Distribution System
JWO	J.W. Ostendorf GmbH & Co. KG
KA	Knowledge Acquisition
KAOS	Keep All Objectives Satisfied
KB	Knowledge Base
KBS	Knowledge Based Systems
KES	Knowledge-Based Intelligent Information and Engineering Systems
KIF	Knowledge Interchange Format
KM	Knowledge Management
KMS	Knowledge Management Systems
KQML	Knowledge Query Manipulation Language
L16	Link 16

LC	Learning Component
LDM	Lexicographic Decision-Making
LISP	LISt Processing
LOD	Levels of Detail
LOS	Line of Sight
MANET	Mobile ad-hoc NETwork
MAPE-K	Monitor, Analyse, Plan, Execute, and Knowledge
MAS	Multi-Agent System
MAUT	Multi-Attribute Utility Theory
MCDA	Multi-Criteria Decision Analysis
MCDM	Multiple Criteria Decision-Making
MIACS	Manufacturing Information and Control System
MIDS	Multi-functional Information Distribution System
MI	Machine Intelligence
MIT	Massachusetts Institute of Technology
MTBF	Mean Time Between Failure
MTTR	Mean Time To Repair
NE	Named Entity
NER	Named Entity Recognition
NLP	Natural Language Processing
NMS	Network Management System
NN	Neural Network
NPV	Net Present Value
NSW	New South Wales
NT	Northern Territory
OEM	Original Equipment Manufacturer
OLAP	Online Analytical Processing
OMG	Object Management Group
OODA	Observe-Orient-Decide-Act
OPM	Option Performance Matrix
OSM	Open Street Maps
OWL	Ontology Web Language
OWL-S	OWL-Services
PAI	Parallel Artificial Intelligence
PDA	Personal Digital Assistant
PHP	Hypertext Preprocessor
PNN	Probabilistic Neural Network
RAM	Random Access Memory
RBF	Radial Basis Function
RBFN	Radial Basis Function Network
RBFNDDA	RBFNN trained with DDA
RBFNN	Radial Basis Function Neural Network
RCEN	Restricted Coulomb Energy Network
RF	Radio Frequency
RMI	Remote Method Invocation

RP	Risk Percentage
RPC	Remote Procedure Call
RVV	Relative Value Vectors
SA	South Australia
SCA	Space Colonization Algorithm
SD	Standard Deviation
SIDATE	Supervised Interoperable Dynamic Agent Teaming Environment
SOAda	Service-Oriented Architecture with decisional aspect
SoaML	Service-Oriented architecture Modeling Language
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SOP	Standard Operating Procedure
SVM	Support Vector Machine
TAS	Tasmania
TDL	Tactical Data Link
TDLNM	TDL Network Manager
TDMA	Time Division Multiple Access
TIDS	Tactical Information Distribution System
TNC	Trust, Negotiation, Communication
TSDF	Time Slot Duty Factor
UAV	Unmanned Aerial Vehicle
UE-FTT	EU–Financial Transaction Tax
UML	Unified Modeling Language
UPnP	Universal Plug and Play
URI	Uniform Resource Identifier
VFM	Vector FM
WA	Western Australia
WBC	Wisconsion Breast Cancer
Web-MAT	Web-based Multi-Agent System Test-bed
WSDL	Web-Services Description Language
WSM	Weighted Sum Model
WWW	World Wide Web
XML	eXtensible Markup Language

Chapter 1
Advances in Intelligent Decision-Making Technology Support

Jeffrey W. Tweedale, Gloria Phillips-Wren and Lakhmi C. Jain

Abstract A succession of data-bases, Advanced Information Processing (AIP) and Intelligent Decision-Making Technologies (IDT) have evolved rapidly over the past five decades. This cumulative evolution of Intelligent Decision Support Systems (IDSSs) has served to stimulate industrial activity and enhanced the lives of most people that the modern world. This publication highlights a series of current contributions to enhance the collective body of knowledge. Artificial Intelligence (AI) and Computational Intelligence (CI) techniques continue to be successfully employed to generate human-like decision-making, while simultaneously providing greater access to information to solve data intensive problems. This book documents innovative contributions that represent advances in Knowledge-Based and Intelligent Information and Engineering Systems. New research recognises that society is familiar with modern AIP and increasingly expect richer IDT systems. Today there is a growing reliance on automatically processing information into knowledge with less human input. Society has already digitised its past and continues to progressively automate knowledge management for the future and increasingly expect to access this information using mobile devices. This book seeks to inform and enhance the exposure of research on intelligent systems and intelligence technologies that utilize

J.W. Tweedale (✉)
School of Engineering,
University of South Australia, Adelaide, Australia
e-mail: jeffrey.tweedale@unisa.edu.au

J.W. Tweedale
Defence Science and Technology Organization,
Edinburgh, Adelaide, Australia
e-mail: jeffrey.tweedale@dsto.defence.gov.au

G. Phillips-Wren
Sellinger School of Business and Management, Loyola University Maryland,
Maryland, USA
e-mail: gwren@loyola.edu

L.C. Jain
Faculty of Education, Science, Technology and Mathematics,
University of Canberra, Belconnen, Australia
e-mail: lakhmi.jain@canberra.edu.au

J.W. Tweedale et al. (eds.), *Intelligent Decision Technology Support in Practice*, Smart Innovation, Systems and Technologies 42,
DOI 10.1007/978-3-319-21209-8_1

novel and leading-edge techniques to improve decision-making. Each chapter concentrates on the theory, design, development, implementation, testing or evaluation of IDT techniques or applications. These approaches have the potential to support decision making in the areas of management, international business, finance, accounting, marketing, healthcare, production, networks, traffic management, crisis response, human interfaces and, military applications. All fourteen chapters represent a broad spread of topics across the domain and highlight how research is being realised to benefit society. Students, professionals and interested observers within the knowledge-based and intelligent information management domain will benefit from the diversity and richness of the content.

Keywords Artificial intelligence · Computational intelligence · Data · Decision-making · Decision support · Information · Intelligence · Knowledge · Wisdom

1.1 Introduction

When McCarthy developed LISP, it enabled Bachman to create the first information data storage system [1]. Although this research stimulated the creation of the first Database Management System (DMS), Feigenbaum quickly exploited the technique to apply constraints within a prescribed search space. This Decision Support System (DSS) employed Heuristic techniques to explain experimental data in organic chemistry. Buchhanan and Lederberg reported the benefits of this Heuristic DENDRAL program in 1971 [2].[1] DMSs were used to collect, fuse and analyze data in order to aggregate information and ultimately generate knowledge. The context influences the technique used. These typically included: communication-driven, data-driven, document-driven, knowledge-driven and model-driven architecture. This era was accompanied with an expansion in research into a variety of intelligent decision support systems that where created to derive greater confidence in the decision being generated [3].

After the introduction of desktop computers in the mid-eighties, most Intelligent Decision-Making Technologies (IDT) research involved the progressive use of technology to collect and interpret data, prior to translating this into information based on 'folk law' or 'symbology'. The wealth of data continues to become unwieldy, forcing researchers to explore data-mining, warehousing and Knowledge Based Systems (KBS), however the key research domains remained focused on problem solving using formal/structured or reasoning systems [4]. This era was accompanied with an expansion in research into a variety of intelligent decision support systems that were created to derive greater confidence in the decision being generated [3]. The growing

[1] The term DENDRAL originally stood for 'DENDRitic ALgorithm'. This invoked a procedure engaged in exhaustive topologically search used to arrange any given set of atoms into a consistent configuration based on their chemical valence.

density of data had an overall effect on the efficiency of these systems. Conversely a series of measures were created to report on the performance of DSS. Factors such as; accuracy, response time and explain-ability were raised as constraints to be considered before specifying courses of action [8]. Since the eighties, Artificial Intelligence (AI) applications have concentrated on problem solving, machine vision, speech, natural language processing/translation, common-sense reasoning and robot control [6]. In the nineties there was a flurry of activity using 'firmware' solutions to overcome speed and compiler complexities, however around the turn of the century, a return to distributed computing techniques has prevailed [7]. Researchers continue to explore the interaction between *Intelligence and Decision-Making* [8]. Hence, the next generation of applications began to focus on new IDT techniques, such as those described in this volume.

The field of IDT has evolved into an interdisciplinary domain of research, bridging concepts from computer science, AI, DSS, and systems engineering. Support for IDT within the Knowledge-Based Intelligent Information and Engineering Systems (KES) community continued to dominate conferences until the first KES International Symposium on IDT in 2009. This symposium series now hosts an annual conference, with the 7th aiming to bridge soft computer science techniques with innovative AI developments to enhance DSS, and Intelligent Decision Support System (IDSS). The conference provides an excellent opportunity for the novice, experts and professionals in the field to present interesting new research, results and discussions about leading edge knowledge management ideation [9, 10]. KES still promotes IDT through its parent conference,[2] the International Journal of Intelligent Decision Technologies[3] and numerous books [11–15]. These success stories include the 5th KES International Conference on Intelligent Decision Technologies, held in Sesimbra, Portugal. This conference supports a mixed-discipline society, who contribute significant skills and knowledge into current applications and theory relating to IDT.

1.2 Decision Support System

A DSS is a set of technologies that aims to improve individual or group decision making by combining knowledge from the decision maker(s) with relevant data from identified sources and applying mathematical and statistical methods and models to suggest preferred decisions. IDSS utilize advanced intelligent methods from the computer science and engineering fields to extend the applicability of DSS to complex problems and enable sophisticated capabilities such as distributed computing. These systems, then add value to the data being collected or promulgated using a knowledge-base to create an inference. For example, clinical DSS combine patient or healthcare data with professional knowledge to assist physicians and other healthcare professionals to diagnose or treat patients. By contrast, a DSS is not a

[2] See http://idt-15.kesinternational.org.

[3] See http://www.iospress.nl/journal/intelligent-decision-technologies/.

Table 1.1 Decision Support System domains categorised

Category	Domain	Example
Data mining	Data driven	Lists, data bases and data management systems
Evolutionary	Genetic algorithms	Optimisation and search spaces
Thought	Neural networks	Learning and pattern matching
Constraints	Rule based systems	Expert and knowledge based systems
Symbolic	Fuzzy logic	Transforming the ambiguity into fuzzy sets
Temporal	Case based systems	Reasoning and analogy based systems
Inductive	Machine learning	Iterative creation of dynamic rule sets

decision making system. Hence, techniques that use control systems in which no human judgement is needed are not considered DSS, although the underlying set of intelligent technologies may be similar.

DSS have matured over a number of data-based eras that can be categorised based on the degree of intelligence provided as shown in Table 1.1. This shows the method listed against each category as interpreted by Dharl and Stein [5].

1.2.1 Data-Driven

Data-driven methodologies were based on data mining techniques. They evolved with lists, databases and data warehousing, which developed into a domain of research that is still titled data mining. The inputs are based on a list of related facts (even when held in databases) that need to be associated and queried prior to being interpreted. For example, On-line Analytical Processing (OLAP), Executive Information Systems (EIS) and Geographic Information Systems (GIS) are types of data-driven DSS.

1.2.2 Genetic Algorithms

This style of data processing originated as a result of an excessive number of parameters. The exponential growth of combinatorial statistics called for a new approach to deriving answers. The use of *Darwinian theory* relating to *survival of the fittest* concept was proven to efficiently find the optimal or best solution of a given probable subset and has been adopted to solve many search space problems.

1.2.3 Neural Networks

A neural network simulates cognitive approach to learning using trial and error. It involves the creator maintaining a prediction loop from which comparisons and an adjustment process can be used to reconcile transfer functions, internal weightings and swagging functions. Forward and backward chaining neural networks have been created to minimise overall network errors.

1.2.4 Rule Based Systems

Systems like *Internist*[4] and *XCON*,[5] use a combination of knowledge management tools (ultimately termed KBS), working memory and rule interpreters[6] to discriminate between a collection of facts represented as evidence in knowledge bases and queries input by operators. Forward chaining (branching down the tree) and backward chaining (evidence matching up through working memory).

1.2.5 Fuzzy Logic

The problem with many of the existing systems is that they do not approximate information the way humans interpret their surroundings (especially RBS). Fuzzy logic introduces natural language membership functions, based on underlying truth statements represented by the Boolean values of '0' or '1'. For example, a numerical temperature reading can be expressed as gradations between 'hot' and 'cold'. This scale could also be expanded to include 'warm' or 'cool'. This enables humans to accept information (measure against a range, without being burdened with the need to process minor errors or fluctuations). The membership is transformed from *crisp sets* into *fuzzy sets*.

1.2.6 Case Based Reasoning

A Case Based Reasoning (CBR) system takes advantage of previous attempts to solve a problem or improve the accuracy of the system. Training is gained using a predefined data set and matured using new patterns as time passes. Reasoning is generated using a series of measures that are used to describe the situation within a scenario. Cases are determined using a pattern matching approach of analysis.

---

[4] Internal Medicine.

[5] A Rule-Based Systems (RBS) created by DEC to manage computing customers needs.

[6] Logic source used as the control strategy.

1.2.7 Machine Learning

Rules are recursively created and connected using trees to represent possible outcomes, based on statistical patterns and algorithms. Tic-tac-toe and chess are good examples of branching and pruning to derive a number of possible results (end-means).

1.3 Intelligent Decision Support System

As intelligent technologies have matured, IDSS have incorporated new capabilities that mimic and extend human cognitive abilities in some manner. Artificial intelligence tools within a DSS can be used to reason to solve problems, plan toward a goal, remember past events, learn from experience, make sense out of ambiguous data, infer based on disparate information, and analyze to predict or optimize outcomes. For example, relevant information to a decision problem from widely distributed data can be surveyed, selected and retrieved; generalized models can be constructed from uncertain data and applied to new problems; associations can be inferred from multiple data sources; unstructured data can be analyzed and incorporated into decision making. This domain is growing rapidly to service the global enterprise and Internet enabled applications that provide access to distributed information and mobile devices.

In todays environment, information needed for decision-making tends to be distributed [16]. Networks exist within and outside of enterprises, homes, the military, government, and national boundaries. Information is segmented for logistical and security reasons onto different machines, different databases, and different systems. Informed decisions may require integration of information from various internal and/or external sources. As enterprises become multi-national, information tends to be distributed geographically across national boundaries and yet collectively required for decision making. The speed of communication in the 21st century requires fast response to be competitive, so the integration of information for decision-making needs to be fast and accurate. Technology can aid decisions that are complex and semi-structured by integrating information flow with analysis using both conventional and artificial intelligence techniques. Agents can assist these efforts by such actions as anticipating the needed information, using the Internet to collect data, and assisting the user in analysis [9].

As an example, intelligent agents can facilitate information processing and user interaction. Agents may serve as task assistants to the user, consultants, or as global assistants that collect information. Agents as task assistants may, for example, retrieve specific data requested by the user, schedule meetings, or work with other Agents on behalf of the user. Agents as consultants could be used for such operations as querying the user, presenting personalised views of data, or ascertaining the information needed for a decision. Agents as global assistants may access distributed information

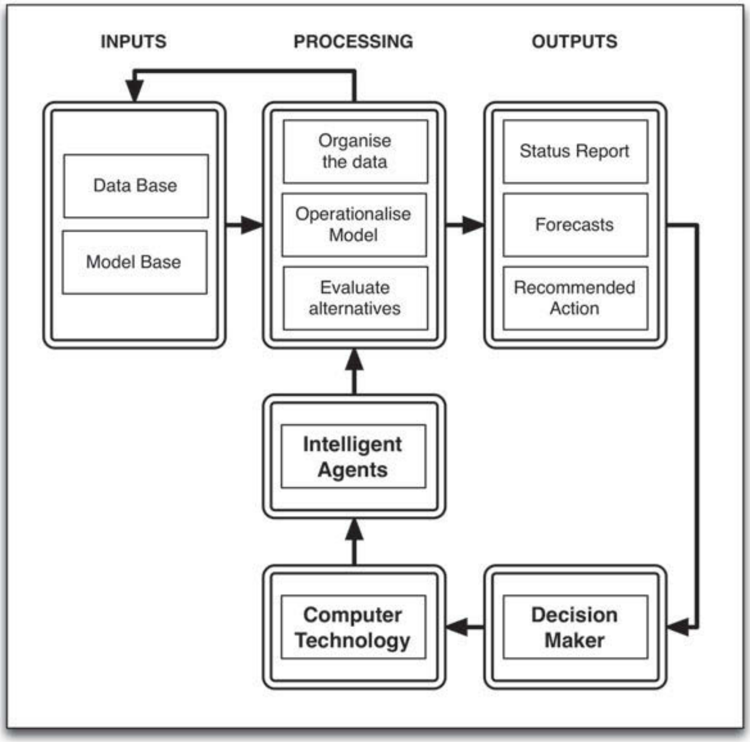


Fig. 1.1 Components of an IDDS enabled by agents [17]

in the enterprise, maintain a current status for the system by updating information in real-time, or use the Internet to bring external information to the decision problem. The major components of an agent-mediated IDSS are are shown in Fig. 1.1 with agents as integral components of the IDSS.

The IDSS has inputs that include the database(s) needed for the decision problem and a model base that includes, for example, the statistical techniques needed for the analysis [18]. Agents may be used to interact with the user or to learn what types of data are needed and assemble them. The processing component permits analysis of the data, including what-if scenarios that the user may desire. During processing, agents may acquire the needed models or consult with the user. The feedback loop indicates interaction between the processing and input components such as real-time updating or user requirements for additional information. The output component provides the result of the analysis to the user and possibly recommendations about the decision. In this component agents may, for example, personalise the output to particular user so that it is presented in a desired way or so that drill-down information is available. The decision maker and the computer technology are components of the overall system and are recognised explicitly in the diagram. The decision maker is usually a human user, although robotic applications increasingly utilise automated decisions that may be implemented with agents [17].

IDSS use a variety of intelligent techniques to meet the demands of the decision problem. For example, Artificial Neural Network (ANN) are useful for their universal approximation capabilities. A major advantage of ANN compared to other methods is their ability to approximate any bounded continuous function to any arbitrarily small approximation error [19]. Thus, ANN are universal approximators and do not make any assumption about the underlying structure or association between a set of inputs and outputs. In addition, ANN learn as they encounter new input-output data so they can be updated and modified in near real-time to incorporate new information. In cases in which data cannot be described as either binary (0 or 1) or with a specific value, fuzzy logic provides the ability to deal with uncertainty. Decision makers are also interested in future states that are not based on past knowledge. In these case, genetic algorithms offer the opportunity to optimize an output for a decision problem by comprehensively exploring the search space to find the global maximum [19]. New intelligent techniques have also shown the ability to discover patterns or associations in big data, automatically apply that knowledge to new situations, and provide decisional guidance. Indeed, the many exciting and useful applications of IDSS demonstrate the rich potential of this field of research.

1.4 Contributions

All contributions in this book were sourced from authors supporting the series of activities associated with Intelligent Decision Technologies (IDT) conference under the auspices of Knowledge Engineering Systems International (KES). A call for chapters was issued following the 5th Conference on Intelligent Decision Technologies, held at Sesimbra, Portugal on 26–28 June 2013. Based on the submissions received, selected authors were invited to enter a competitive process prior to acceptance and extend their papers into a book chapter. Following an iterative review process, an innovative subset of topics and techniques are included in this volume. A full list of chapters is show in Table 1.2.

This book seeks to inform and enhance the exposure of research on intelligent systems and intelligent technologies that utilize novel and leading-edge techniques to improve decision-making. Each chapter concentrates on the theory, design, development, implementation, testing or evaluation of IDT techniques or applications. These approaches have the potential to support decision making in the areas of management, international business, finance, accounting, marketing, healthcare, production, networks, traffic management, crisis response, human interfaces and, military applications. A guide to understand some of these trends and future directions intelligent decision-making technology is provided by Multi-agent Systems (MAS) [9]. Several of the editors also highlighted their thoughts about future directions of IDT in 2008 [10]. Although the 2008 article focuses on building a decision-support framework using MAS to operate as teams, these trends have been realised around the world.

Table 1.2 List of topics within this contribution

Chapters	Title
2	Asset Management Strategies: Risk and Transaction Costs in Simulation
3	Decision Support System for Energy Savings and Emissions Trading in Industrial Scenarios
4	A Parsimonious Radial Basis Function-Based Neural Network for Data Classification
5	Personalized Intelligent Mobility Platform: An Enrichment Approach using Social Media
6	Exploiting Alternative Knowledge Visualizations and Reasoning Mechanisms to Enhance Collaborative Decision Making
7	Decision-Making in a Distributed and Dynamically Scalable Environments
8	Enhancing the Tactical Data Link Decision Support System
9	AC^{3}M: The Agent Coordination and Cooperation Cognitive Model
10	Wind Rendering in 3D Modeling Landscape Scenes
11	Extending the Service Oriented Architecture to include a decisional aspect
12	An extended dependability case to share responsibility knowledge
13	Designing a Hybrid Recommendation System for TV Content
14	Incompleteness and fragmentation: Possible Formal Cues to cognitive processes behind Spoken Utterances

Several examples provided in this volume include: 'assessing Asset Management Strategies' in Chap. 2 and 'Knowledge Visualization Decision-Making in a Distributed and Dynamically Scalable Environments' in Chap. 6.

Computational social science involves the use of agent-based modeling and simulation to study complex issues [20]. It is related to a variety of techniques, methodologies and approaches. There are two accepted approaches, these include 'synthesis' and 'analysis'. Simulation generally involves stochastic (deterministic), continuous (discrete) and dynamic (event-based) simulation. The later typically involves MAS, incorporated in distributed artificial intelligence components. Chapter 2 provide background knowledge on MAS, Mathematical modelling, an extension of the transaction costs, its implementation using JADE.[7] This shows that the probability of an agent switching from a fundamental to technical behavior depends on the historic trend of assets prices. Šperka et al. hypothesis for this research was based previous simulation results and focused on the fundamental rules [21]. Future effort extends to investigating risk and parameterization of the model in order to prove that the Tobin tax has a positive impact on the stability of financial market.

[7] See http://jade.tilab.com.

Emission Trading System (ETS) have been subject to significant rigor over the past decade. The volume of data being collected has increased dramatically. Government policy continues to influence industrial compliance and legislation, forcing companies to collect data to measure both energy savings and carbon emissions. Chapter 3 proposes a decision support approach to determine societal reactions and any adjustments required to the ETS. Marques and Neves-Silva [22] implement a multi-criteria decision analysis using MACBETH. They use a CBR approach that is stimulated via probabilistic analysis and describe multiple case studies used to analyse their concepts. The results obtained provide excellent correlation which is encouraging the utilization of this technique in future applications.

ANN have been used to conduct supervised learning in AI application to solve problems where regression, classification and time series prediction are required. Radial Basis Function (RBF) algorithms were introduced as a simplified approach to support approximation functionality to interpret forward selection [23]. Chapter 4 introduced a RBF neural network that incorporates greedy insertion behavior with a dynamic decay adjustment. Tan et al. [24] incorporate this algorithm in an application they call Radial Basis Function Network with the Dynamic Decay Adjustment (RBFNDDA). Hidden nodes of RBFNDDA are re-organized through a supervised Fuzzy ArtMAP (FAM) classifier, and those parameters are adapted using the Harmonic Means (HM) algorithm. They discuss FAM, the experiment, the comparative performance against other classifiers prior to summarising their results. They indicate that the proposed model is able to produce a compact structure network with high performance when classifying subject data.

Intelligent Transportation Systems (ITS) refer to a variety of mechanisms used to support safe and coordinated use of transport networks. Some countries are experimenting with innovative services that deliver ‘smarter’ transport and traffic management systems. These include: infrastructure, platforms, monitoring, rescue, evacuation, mobility and management. Europe is leading this innovation under EU Directive 2010/40/EU (7 July 2010). Chapter 5 present a technical approach for developing a personalized mobility knowledge base. This is supported by mechanisms for extracting and processing tweets related with traffic events by commuters. Costa et al. use a step-wise approach to enrich their knowledge model using heterogeneous data sources derived from commuter Personal Digital Assistants (PDAs) [25]. They progressively introduce their concept and objectives, which are supported using existing literature. The methodology discusses base graphs, smart graphs and personalized smart graphs. Using a formalized vocabulary, the authors apply real-time traffic events from twitter prior to making their comparison. At present, their results do not address the final conclusion, but do form the basis for the formalization of the domain knowledge being acquired. Ongoing effort is being funded under EU FP7 MobiS project.

Engaging in collaborative decision making in today’s knowledge intensive and complex environments is a challenging task. The diversity of these environments and the associated plurality of decision makers perceptions require multi-disciplined support. Chapter 6 reports on an innovative approach that offers a number of interrelated visualization tools to facilitate the knowledge exchange and during a collaborative decision making process. The authors define the concept of multi-criteria decision making, the dicode approach, and scenarios to explain its use. Christodoulou et al. believe that the major benefit of this work is the implementation of a number of mechanisms capable of displaying the result of the decision making process with a user-friendly what-if analysis [26]. These visualizations incorporate suitable reasoning mechanisms that exploit human to machine understandable knowledge and aid stakeholders towards reaching consensus and collective decision.

Chapter 7 discusses interoperability within and across ubiquitous platforms. Modern computing operations have evolved to a level where plug 'n' play protocols can be used to invoke common interfaces. Designers are able to create dynamic interfaces using reflectance to effective and efficient conduct distributed decision-making. Many applications now use mobile agents to support web-centric activities. This capability enables decentralised data mining and supports IDSS. Tweedale provides a brief description relating to the evolution of data processing, from Lisp to IDSS, prior to discussing methodologies used to support dynamic environments [27]. This chapter seeds many ideas for researchers who are increasingly being asked to provide more adaptive intelligent services.

Defence employs Tactical Data Links (TDLs) to maintain situation awareness of assets during a mission and to convey the intent of command and control information. Network Managers are use to optimise the link. A DSS is already used to enhance their ability to process the vast amount of information generated in real-time. The introduction of a DSS to support an Network Management System (NMS) is expected to reduce the burdens on the TDL Network Manager (TDLNM) and also mitigate the effects of operators who may have reduced experience. Sioutis and Dominish use Cognitive Work Analysis (CWA) to acquire subject knowledge that will ultimately be incorporated into a DSS to compliment the experience required to efficiently operate the network effectively [28]. Chapter 8 discusses the TDL network, the process of managing the network, the decision processes used, design patterns and a suitable agent architecture to be used to design a DSS service. Although, this research is time consuming, four modules have already been identified for future implementation.

To promote IDT within industry it will be necessary to emulate the human ability to coordinate and cooperate. Here coordination is referred to as the management of the interdependencies of activities and cooperation as voluntary relationship of two parties to share resources used to achieve a common goal. Chapter 9 describes the evolution of the Agent Coordination and Cooperation Cognitive Model (AC^3M) and its successful application in team automation when operating within dynamic environments. The aim of AC^3M is to emphasise how Coordination and Cooperation ($CO–O^2$) models improve an agent's autonomy and decision-making capabilities. The AC^3M's $CO–O^2$ model ensures that a decision to act can be made quickly and effectively. Consoli uses a Stimulus/Perceptor framework to envoke coordination and cooperation on two levels: Intelligence, surveillance and reconnaissance, and command and control. The author highlights the importance of identifying the link between an agent's belief, desires and intentions with coordination and cooperation [29]. She also believes she has proven that AC^3M can enhance the automation of a team and their situational awareness. This was demonstrated in an application to coordinate multiple flight mission systems and promote cooperation between components of individual Unmanned Aerial Vehicles (UAVs).

A space colonization algorithm was applied to make compact and realistic tree models. Wind rendering is a necessary procedure in modeling realistic scenes with minimal computational costs. Chapter 10 uses various tree shapes and wind parameters to simulate three levels of wind conditions (weak, mild and storm-force). Favorskaya and Tkacheva describes the process of transforming a 3D cloud of laser scanned scenery points into digital models [30]. They explain the rendering process for each wind strength and the experiments used to verify the process. This OpenGL tool successfully restores a view of the earths surface and the natural objects within scenes landscape. There experiments show the new architecture accurately delivers decisional needs to the enterprise providing greater agility. Future plans indicate this research will integrate the decisional aspects in cloud-base computing applications.

The emergence of Service-Oriented Architecture (SOA) components within organisational applications is increasingly being used to provide decision processes with support systems. Boumahdi and Chalal introduce a meta-model called Decisional Model of Service (DMoS) using three views to each concept (business, information and decision) [31]. Chapter 11 provides a case study that illustrates an inventory management system using the proposed architecture. The authors provide background on current decision-making processes within an organization, prior to linking each concpet to their architectural dimensions. In the future works, the integration of decisional aspect in cloud computing is envisaged.

Information related to dependability is important knowledge that must be shared among stakeholders. Existing methods cannot clearly describe dependability cases using relationship between a claim and a responsibility. Chapter 12 introduces a proposal that uses the d* framework to define relationship between attributes for sharing knowledge and those for achieving agreements among stakeholders. Saruwatari et al. use three example applications to deomonstrate the effectiveness of the d* framework. They use a meta-model to map responsibility attributes [32]. The case studies use a simple graphical notation to mapan elevator, AP download and LAN device management systems. Future plans include evaluating the effectiveness of the proposed concept using 'practical' dependability cases.

Traditional TV recommendation systems are based on an individual's viewing activity and at present, only recommend program choices. These systems have not be modified to include Smart TV capabilities. Given access to the internet, applications on Smart TV's can be used to provide a hybridised recommendation system. This would include, personal preferences, other users viewing habits and information related to additional content, such as; TV programs, movies, and even music. Chapter 13 includes a detailed history of existing systems, prior to describing the proposed framework for component analysis and learning system. Chang et al. discuss how these components can be used to identify diversity, novelty, explanation and recommendations [33]. The three components presented will be incorporated into a physical recommendation system and additional research conducted to verify the process.

Spontaneous communication is an essential element of social speech. Natural Language Programming (NLP) is difficult because the sounds you hear and words you read are often hard to identify using speech. Currently synthesised speech is often fragmented and not considered to be real unless grammatically correct. Chapter 14 presents numerous forms of generating dialogue using their syntax, prosody and overall multi-modality for the Hungarian corpus. The research aims at improving the robustness of the spoken form of natural language technology. Hunyadi discuss a generative model of human to human communication followed by a schema supporting a generative Theoretical-Technological speech model [34]. They identify the minimum requirements (building blocks) required to create structured speech non-verbal modalities of Prosody in Hungarian speech. They then explore duration, intonation and intensity before creating a temporal alignment of gestures and clauses. Further work is required to gain complete insight into the real nature of multimodal communication. This understanding is considered essential prior to researchers successfully generating human to machine interactions.

1.5 Conclusion

The editors believe that this book represents a valuable resource to the novice, students, professionals and anyone seeking further information about knowledge management and innovative methods of employing Intelligent Decision-Making Technologies to deliver sound knowledge management to consumers. The editors recommend that the reader uses both the table of contents, list of acronyms and index to help navigate each topic. For those focusing on specific topics, it is recommended that the reader also explores the surrounding chapters for associated tools, techniques or methodologies. We hope you enjoy the content and are as inspired as much as the editors and subject matter experts with the potential of IDT to improve our world. All authors would be delighted to field questions or discuss their research with you. Their contact details are provided in the 'List of Contributors' and at the bottom of the first page of each chapter.

References

1. Bachman, C.W.: Integrated data store: the information processing machine the we need! In: General Electric Computer Users Symposium, General Electric, New York (1962)
2. Buchanan, B.G., Lederberg, J.: The Heuristic DENDRAL program for explaining empirical data. In: Technical Report, Stanford University, Stanford, CA, USA (1971)
3. Yen, J., Fan, X., Sun, S., Hanratty, T., Dumer, J.: Agents with shared mental models for enhancing team decision makings. Decis. Support Syst. **41**(3), 634–653 (2006)
4. Bigus, J.P., Bigus, J.: Constructing Intelligent Agents Using Java. Professional Developer's Guide Series. Wiley, New York (2001)
5. Dhar, V., Stein, R.: Intelligent Decision Support Methods: The Science of Knowledge Work. Prentice-Hall Inc, Upper Saddle River (1997)

6. Rich, E., Knight, K.: Artificial Intelligence. McGraw-Hill College, New York (1991)
7. Jain, L., Tweedale, J.: Research directions in the KES centre. In: Chan, C.C., Grzymala-Busse, J., Ziarko, W. (eds.) Rough Sets and Current Trends in Computing. Lecture Notes in Computer Science, vol. 5306, pp. 13–20. Springer, Berlin (2008)
8. Tweedale, J.W., Jain, L.C. (eds.): Recent Advances in Knowledge-based Paradigms and Applications: Enhanced Applications Using Hybrid Artificial Intelligence Techniques. Advances in Intelligent Systems and Computing, vol. 234. Springer, Berlin (2013)
9. Tweedale, J., Ichalkaranje, N., Sioutis, C., Jarvis, B., Consoli, A., Phillips-Wren, G.: Innovations in multi-agent systems. J. Netw. Comput. Appl. **30**(3), 1089–1115 (2006)
10. Tweedale, J., Sioutis, C., Phillips-Wren, G.E., Ichalkaranje, N., Urlings, P., Jain, L.C.: Future directions: building a decision making framework using agent teams. In: Phillips-Wren, G.E., Ichalkaranje, N., Jain, L.C. (eds.) Intelligent Decision Making: An AI-Based Approach. Studies in Computational Intelligence, vol. 97, pp. 387–408. Springer, New York (2008)
11. Nakamatsu, K., Phillips-Wren, G., Jain, L.C., Howlett, R.J. (eds.): New Advances in Intelligent Decision Technologies: Results of the First KES International Symposium IDT, Studies in Computational Intelligence, vol. 199. Springer, Heidelberg (2009)
12. Phillips-Wren, G. (ed.): Advances in Intelligent Decision Technologies: Proceedings of the 2nd KES International Symposium IDT, in Smart Innovation, Systems and Technologies, vol. 4. Springer, Heidelberg (2010)
13. Watada, J., Phillips-Wren, G., Jain, L., Howlett, R. (eds.): Proceedings of the 3rd International Conference on Intelligent Decision Technologies: Smart Innovation, Systems and Technologies, vol. 10. Springer, Heidelberg (2011)
14. Watada, J., Watanabe, T., Phillips-Wren, G., Howlett, R.J., Jain, L.C. (eds.): Intelligent Decision Technologies: Proceedings of the 4th International Conference on Intelligent Decision Technologies, in Smart Innovation, Systems and Technologies, vol. 16. Springer, Heidelberg (2012)
15. Neves-Silva, R., Watada, J., Phillips-Wren, G. (eds.): Intelligent Decision Technologies: Proceedings of the 5th KES International Conference on Intelligent Decision Technologies. IOS Press, Amsterdam (2013)
16. Du, H., Jia, X.: A framework for web-based intelligent decision support enterprise. In: 23rd IEEE International Conference on Distributed Computing Systems Workshops, pp. 958–961. IEEE Press, New York (2003)
17. Phillips-Wren, G., Forgionne, G.: Advanced decision-making support using intelligent agent technology. J. Decis. Syst. **11**, 165–184 (2002)
18. Phillips-Wren, G.: AI tools in decision making support systems: a review. Int. J. Artif. Intell. Tools **21**(2), 1–13, 1240005 (2012)
19. Pedrycz, W., Gomide, F.: Fuzzy Systems Engineering: Toward Human-Centric Computing. Wiley, Hoboken (2007)
20. Tweedale, J.W., Jain, L.C. (eds.): Advanced Techniques for Knowledge Engineering and Innovative Applications: 16th International Conference, KES 2012, San Sebastian, Spain, 10–12 Sept 2012. Revised Selected Papers. Communications in Computer and Information Science, vol. 246. Springer, Berlin (2013)
21. Šperka, R., Spišák, M., Górecki, J.: Stability of financial market: the introduction to the java simulation of agent-based model. In: Neves-Silva, R., Watada, J., Phillips-Wren, G., Jain, L.C., Howlett, R.J. (eds.) Intelligent Decision Technologies. Frontiers in Artificial Intelligence and Applications, vol. 241, pp. 380–389. IOS Press, Amsterdam (2013)
22. Marques, M., Neves-Silva, R.: Approach for decision support for energy savings and emissions trading based on industrial requirements. In: Neves-Silva, R., Watada, J., Phillips-Wren, G., Jain, L.C., Howlett, R.J. (eds.) Intelligent Decision Technologies. Frontiers in Artificial Intelligence and Applications, vol. 255, pp. 74–83. IOS Press, Amsterdam (2013)
23. Broomhead, D., Lowe, D.: Multivariate functional interpolation and adaptive networks. Complex Syst. **2**, 321–355 (1988)
24. Tan, S.C., Lim, C.P., Watada, J.: A hybrid rbf-art model and its application to medical data classification. In: Neves-Silva, R., Watada, J., Phillips-Wren, G., Jain, L.C., Howlett, R.J.

(eds.) Intelligent Decision Technologies. Frontiers in Artificial Intelligence and Applications, vol. 255, pp. 21–30. IOS Press, Amsterdam (2013)
25. Costa, R., Figueiras, P., Maló, P., Jermol, M., Kalaboukas, K.: Mobis—personalized mobility services for energy efficiency and security through advanced artificial intelligence techniques. In: Neves-Silva, R., Watada, J., Phillips-Wren, G., Jain, L.C., Howlett, R.J. (eds.) Intelligent Decision Technologies. Frontiers in Artificial Intelligence and Applications, vol. 241, pp. 296–306. IOS Press, Amsterdam (2013)
26. Christodoulou, S., Karacapilidis, N., Tzagarakis, M.: Advancing collaborative decision making through alternative visualizations and reasoning mechanisms. In: Neves-Silva, R., Watada, J., Phillips-Wren, G., Jain, L.C., Howlett, R.J. (eds.) Intelligent Decision Technologies. Frontiers in Artificial Intelligence and Applications, vol. 241, pp. 38–47. IOS Press, Amsterdam (2013)
27. Tweedale, J., Jain, L.: Multilingual agents in a dynamic environment. In: Nakamatsu, K., Phillips-Wren, G., Jain, L., Howlett, R. (eds.) New Advances in Intelligent Decision Technologies. Studies in Computational Intelligence, vol. 199, pp. 193–202. Springer, Berlin (2009)
28. Sioutis, C., Dominish, D.: Developing intelligent agents with distributed computing middleware. In: Watada, J., Phillips-Wren, G., Jain, L., Howlett, R. (eds.) Intelligent Decision Technologies. Smart Innovation, Systems and Technologies, vol. 10, pp. 863–872. Springer, Berlin (2011)
29. Consoli, A., Tweedale, J., Jain, L.: Defining a decision-support framework in AC^{3}M. In: Nakamatsu, K., Phillips-Wren, G., Jain, L., Howlett, R. (eds.): IDT'09: First KES International Symposium in Intelligent Decision Technologies, pp. 203–212 (2009)
30. Favorskaya, M., Tkacheva, A.: Rendering of wind effects in 3D landscape scenes, 17th international conference in knowledge based and intelligent information and engineering systems. Proc. Comput. Sci. **22**, 1229–1238 (2013)
31. Boumahdi, F., Chalal, R.: Soada: service oriented architecture with a decision aspect, 17th international conference in knowledge based and intelligent information and engineering systems. Proc. Comput. Sci. **22**, 340–348 (2013)
32. Saruwatari, T., Hoshino, T., Yamamoto, S.: Method to share responsibility knowledge of dependability cases, 17th international conference in knowledge based and intelligent information and engineering systems. Proc. Comput. Sci. **22**, 1073–1082 (2013)
33. Chang, N., Irvan, M., Terano, T.: A TV program recommender framework, 17th international conference in knowledge based and intelligent information and engineering systems. Proc. Comput. Sci. **22**, 561–570 (2013)
34. Hunyadi, L.: Incompleteness and fragmentation in spoken language syntax and its relation to prosody and gesturing: cognitive processes versus possible formal cues. In: 2013 IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom), Dec 2013, pp. 211–218

Chapter 2
Asset Management Strategies: Risk and Transaction Costs in Simulation

Roman Šperka

Abstract In recent years, there has been rising interest in a field called behavioral finance, which incorporates psychological methods in analysing investor behavior. The aim of this chapter is to study the technical and the fundamental investing strategy of financial market participants dealing with assets. The motivation of the presented research is to simulate the financial market in the form of agent-based model and to investigate various impacts of risk and transaction costs on its stability. Computational social science involves the use of agent based modeling and simulation to study complex social systems. It is related to a variety of other simulation techniques, including discrete event simulation and distributed artificial intelligence or Multi-Agent Systems (MAS). In practice, each agent has only partial knowledge of other agents and each agent makes its own decisions based on the partial knowledge about other agents in the system. For purposes of this chapter, a MAS will be implemented as a simulation framework in JADE development platform. The hypothesis was that transaction costs introduction will stabilize the financial market. The results obtained show that in the case of risk involvement into the system the hypothesis can be fulfilled only partially.

Keywords Simulation · Modelling · ABMS · JADE · Tobin tax · Risk · Transaction costs

2.1 Introduction

As the globalization proceeds, financial markets follow this process in the way of integration and growth. In the asset management area different participants invest their capital into financial markets. The reward for the investments is the revenue they could cumulate. It is obvious that these investors have important impact on the

Roman Šperka (✉)
School of Business Administration in Karviná, Silesian University in Opava, Opava, Czech Republic
e-mail: sperka@opf.slu.cz

J.W. Tweedale et al. (eds.), *Intelligent Decision Technology Support in Practice*, Smart Innovation, Systems and Technologies 42,
DOI 10.1007/978-3-319-21209-8_2

asset prices. The speculations they could realise influence the stability on financial markets in both ways: positively, and negatively. Technical investment strategy tries to maintain an average return using benchmarks based on market indices. It is one of the most popular investment strategies in the asset management business and is consistent with traditional asset pricing theories. This strategy is also considered to be an effective method in efficient markets [1].

Recently, there has been rising interest in a field called behavioral finance, which incorporates psychological methods in analysing investor behavior. There are numerous arguments in behavioral finance that investors' decision making bias can explain phenomenon in the financial market which until now had gone unexplained. Such arguments often point out the limit of arbitrage and the existence of systematic biases in decision-making [2–4]. Behavioral finance has examined a wide range of phenomena in the market and among investors, drawing a number of provocative conclusions.

This research employs a Multi-Agent Systems (MAS) to deal with unpredictable phenomena surrounding every company nowadays able of agents behavior investigation. MAS will be developed and managed as a simulation framework in the JADE development environment (JAVA programming language). This chapter deals with the agent-based simulation of the Tobin tax introduction together with the risk analysis and their impact on the stability of financial market. The motivation is to investigate the reaction of financial market on the higher transaction costs and risk application. A multi-agent financial market model and simulation is introduced. Intelligent Agents (IA) follow technical and fundamental trading rules to determine their speculative investment positions. The authors consider direct interactions between speculators due to which they may decide to change their trading behavior. For instance, if a technical trader meets a fundamental trader and they realize that fundamental trading has been more profitable than technical trading in the recent past, the probability that the technical trader switches to fundamental trading rules is relatively high. In particular the influence of transaction costs and risk is studied. This chapter is structured as follows. Some literature background is given in Sect. 2.2. Section 2.3 firstly describes the original mathematical model, secondly informs about previous simulation results, and lastly represents the hypothesis. In Sect. 2.4 is the JADE environment and JAVA implementation presented. Section 2.5 discusses the original simulation results of the agent-based model of financial market.

2.2 Literature Background

Computational social science involves the use of Agent-Based Modeling and Simulation (ABMS) to study complex social systems [5, 6]. ABMS consists of a set of agents and a framework for simulating their decisions and interactions. ABMS is related to a variety of other simulation techniques, including discrete event simulation and distributed artificial intelligence or multi-agent systems [7, 8] Although many traits are shared, ABMS is differentiated from these approaches by its focus on finding the set of basic decision rules and behavioral interactions that can produce the complex results experienced in the real world [9]. ABMS tools are designed to simulate the interactions of large numbers of individuals so as to study the macro-scale consequences of these interactions [10]. Each entity in the system under investigation is represented by an agent in the model. An agent is thus a software representation of a decision-making unit. Agents are self-directed entities with specific traits and typically exhibit bounded rationality, that is, they make decisions by using limited internal decision rules that depend only on imperfect local information. In practice, each agent has only partial knowledge of other agents and each agent makes its own decisions based on the partial knowledge about other agents in the system [11, 12].

Intelligent agent technology used in this chapter has deeper roots in economic theory history, mainly in the ideas of Hayek and Simon. One of the main ideas of Hayek is that the economic system should be studied from bottom. He stresses the need to look at the market economy as to a decentralized system consisting of mutually influencing individuals (the same goes for financial markets) in his work. In “Individualism and Economic Order” Hayek [13] writes: “There is no other way to understand social phenomena such as through the understanding of the actions of individuals who are oriented towards other people and management according to their expected behaviours.” He opposed mainly against collectivist theories which claim to be able to fully understand the social right, regardless of the individuals who constitute them. This approach builds a contrast with the assumption of perfect information, which is used in traditional equilibrium analysis. In the theory of complex systems, where Agent-based Modelling and Simulation (ABMS) clearly falls, is this idea the primary principle [14]. Agents, unlike classical equilibrium approach have not perfect information about all processes in the system.

No strict rules are conducted to the agents. They themselves select those practices that lead to the best results according to the success of strategies and rules. They are not looking for universal general rule. They are governed by a method that has proven in the environment under given conditions in the past. Multi-agent approach uses various scientific methods for introducing the adaptive behavior of the program structures [15]. The basic feature of complex adaptive systems is that their global properties can be easily derived from the characteristics of individual units. Although each agent structure is simple, the behavior of the system as a whole can be very difficult. A complex system is not the same as a chaotic system. Generally, a complex system tends to evolve away from both extremes—full of randomness on the one hand and absolute order on the other. Kochugovindan [16] MAS are based on the selection of behavior rules that are subjectively optimal in certain environment for each agent functioning. MAS implemented through ABMS consists of two types of rules—spontaneous and created. The agent should be determined what their purpose is, what variable or group of variables to be monitored and optimized. On the other hand the way for reaching goals, is already left full of them.

The transaction costs on the financial market are mainly the costs of the obtaining and the interpreting of the information, the time required for decision making, various types of fees. Transaction costs according to Burian [17] are often viewed as negative phenomena, but there are cases where the increase in the transaction costs can be viewed positively and can contribute to the stability of the market. The increase in the transaction costs may also occur in the form of non-market regulation such as the taxes. In the early seventies the Nobel laureate in the economics James Tobin drafted the regulation of currency markets. Tobin suggested that all short-term transactions should be taxed at a low fixed rate (the proposal was later identified as the so-called Tobin tax). The results according to Tobin would avoid short-term currency speculation and stabilize the market. Currency speculation can lead to the sudden withdrawal of the currency from the circulation in order to artificially increase the price. The consequence for the economy of the countries that use this currency may be a temporary reduction in liquidity, problems in obtaining loans and other phenomena that can lead to the reduced growth or even to the recession.

Tobin suggested his currency transaction tax in 1972 in his Janeway Lectures at Princeton, shortly after the Bretton Woods system of monetary management ended in 1971 [18]. He summarized his idea like follows: "The tax on foreign exchange transactions was devised to cushion exchange rate fluctuations. The idea is very simple: at each exchange of a currency into another a small tax would be levied—let's say, 0.5 % of the volume of the transaction. This dissuades speculators as many investors invest their money in foreign exchange on a very short-term basis. If this money is suddenly withdrawn, countries have to drastically increase interest rates for their currency to still be attractive. But high interest is often disastrous for a national economy, as the nineties' crises in Mexico, Southeast Asia and Russia have proven. My tax would return some margin of manoeuvre to issuing banks in small countries and would be a measure of opposition to the dictate of the financial markets" [19]. More variations on Tobin tax idea occurred. According to Paul Bernd Spahn in 1995, "Analysis has shown that the Tobin tax as originally proposed is not viable and should be laid aside for good." Furthermore, he said: "...it is virtually impossible to distinguish between normal liquidity trading and speculative "noise" trading. If the tax is generally applied at high rates, it will severely impair financial operations and create international liquidity problems, especially if derivatives are taxed as well. A lower tax rate would reduce the negative impact on financial markets, but not mitigate speculation where expectations of an exchange rate change exceed the tax margin" [20].

Wrobel's paper [21] highlighted the Swedish experience with financial transaction taxes. In January 1984, Sweden introduced a 0.5 % tax on the purchase or sale of an equity security. Thus a round trip (purchase and sale) transaction resulted in a 1 % tax. In July 1986 the rate was doubled. In January 1989, a considerably lower tax of 0.002 % on fixed-income securities was introduced for a security with a maturity of 90 days or less. On a bond with a maturity of five years or more, the tax was 0.003 %. The revenues from taxes were disappointing; for example revenues from the tax on fixed-income securities were initially expected to amount to 1,500 million Swedish kronor per year. They did not amount to more than 80 million Swedish kronor in any year and the average was closer to 50 million [22]. In addition, as taxable trading volumes fell, so did revenues from capital gains taxes, entirely offsetting revenues from the equity transactions tax that had grown to 4,000 million Swedish kronor by

1988 [23]. On the day that the tax was announced, share prices fell by 2.2 %. But there was leakage of information prior to the announcement, which might explain the 5.35 % price decline in the 30 days prior to the announcement. When the tax was doubled, prices again fell by another 1 %. These declines were in line with the capitalized value of future tax payments resulting from expected trades. It was further felt that the taxes on fixed-income securities only served to increase the cost of government borrowing, providing another argument against the tax. Even though the tax on fixed-income securities was much lower than that on equities, the impact on market trading was much more dramatic. During the first week of the tax, the volume of bond trading fell by 85 %, even though the tax rate on five-year bonds was only 0.003 %. The volume of futures trading fell by 98 % and the options trading market disappeared. On 15 April 1990, the tax on fixed-income securities was abolished.

The EU Financial Transaction Tax (EU FTT) is a proposal made by the European Commission in September 2011 to introduce a financial transaction tax within the 27 member states of the European Union by 2014. The tax would only impact financial transactions between financial institutions charging 0.1 % against the exchange of shares and bonds and 0.01 % across derivative contracts. According to the European Commission it could raise €57bn every year [24], of which around €10bn (£8.4bn) would go to Great Britain, which hosts Europe's biggest financial centre [25]. It is unclear whether a financial transaction tax is compatible with European law [26].

The difference between Tobin tax and financial transaction tax is particularly in the tax subject. Tobin tax concentrates on the currency operations, while financial transaction tax deals with assets like shares, bonds and derivative contracts. Both terms are used within this chapter, however only assets were used for calculations during this research.

2.3 Mathematical Model

2.3.1 Original Model

The model developed by Westerhoff [27] was chosen for the implementation. It is an agent-based model, which simulates the financial market. Two base types of traders are represented by agents:

- **Fundamental traders**—their reactions are based on fundamental analysis—they believe that asset prices in long term approximate their fundamental price—they buy assets when the price is under fundamental value.
- **Technical traders**—decide using technical analysis—prices tend to move in trends—by their extrapolating there comes the positive feedback, which can cause the instability.

Price changes are reflecting current demand excess. This excess is expressing the orders amount submitted by technical and fundamental traders each turn and the rate between their orders evolves in a time. Agents regularly meet and discuss their trading performance. One agent can be persuaded to change his trading method, if his rules relative success is less than the others one. Communication is direct talk one agent with other. Communicating agents meet randomly—there is no special relationship between them. The success of rules is represented by current and past profitability. Model assumes traders ability to define the fundamental value of assets and the agents behave rationally.

The price is reflecting the relation between assets that have been bought and sold in a turn and the price change caused by these orders. This can be formalized as a simple log-linear price impact function.

$$P_{t+1} = P_t + a(W_t^C D_t^C + W_t^F D_t^F) + a_t \tag{2.1}$$

where a is positive price adjustment coefficient, D^C are orders generated by technical agents while D^F are orders of fundamental ones. W^C and W^F are weights of the agents using technical respectively fundamental rules. They are reflecting current ratio between the technical and fundamental agents. α brings the random term to the Eq. 2.1. It is an IID normal random variable with mean zero and constant standard deviation σ^α.

As was already said, technical analysis extrapolates price trends—when they go up (price is growing) agents buy the assets. So the formalization for technical order rules are expressed in Eq. 2.2.

$$D_t^C = b(P_t - P_{t-1}) + \beta_t \tag{2.2}$$

The parameter b is positive and presents agent sensitivity to price changes. The difference in brackets reflects the trend and β is the random term—IID normal random variable with mean zero and constant standard deviation σ^β.

Fundamental analysis permits the difference between price and fundamental value for short time only. In long run there is an approximation of them. So if the price is below the fundamental value—the assets are bought and vice versa—orders according fundamentalists are formalized in Eq. 2.3.

$$D_t^F = c(F_t - P_t)\gamma_t \tag{2.3}$$

The parameter c is positive and presents agent sensitivity to reaction. F represents fundamental value—the authors maintain a constant value to keep the implementation as simple as possible. γ is the random term—IID normal random variable with mean zero and constant standard deviation σ^γ.

If we say that N is the total number of agents and K is the number of technical traders, then Eq. 2.4 defines the weight of technical traders.

$$W_t^C = K_t/N \tag{2.4}$$

And the weight of fundamental traders in Eq. 2.5.

$$W_t^F = (N - K_t)/N \tag{2.5}$$

Two traders meet at each step and they discuss about the success of their rules. If the second agent rules are more successful, the first one changes its behavior with a probability K. Probability of transition is defined as $(1 - \delta)$. Also there is a small probability ε that agent changes his mind independently. Transition probability is formalized in Eq. 2.6.

$$K_t = (K_{t-1} + 1) \text{ with probability } p_{t-1}^{+} = \frac{N - K_{t-1}}{N}\left[\varepsilon + (1-\sigma)_{t-1}^{F\to C}\frac{K_{t-1}}{N-1}\right],$$

$$K_t = (K_{t-1} + 1) \text{ with probability } p_{t-1}^{-} = \frac{K_{t-1}}{N}\left[\varepsilon + (1-\sigma)_{t-1}^{C\to F}\frac{N - K_{t-1}}{N-1}\right],$$
$$K_t = K_{t-1}, \text{ with probability } 1 - p_{t-1}^{+} - p_{t-1}^{-}. \tag{2.6}$$

where the probability that fundamental agent becomes technical one is shown in Eq. 2.7.

$$\begin{aligned}\left(1 - \delta_{t-1}^{F\to C}\right) &= 0,5 + \lambda \quad \text{for } A_t^C \rangle A_t^F,\\ \left(1 - \delta_{t-1}^{F\to C}\right) &= 0,5 - \lambda \quad \text{otherwise}\end{aligned} \tag{2.7}$$

Respectively that technical agent becomes fundamental one is shown in Eq. 2.8.

$$\begin{aligned}\left(1 - \delta_{t-1}^{C\to F}\right) &= 0,5 - \lambda \quad \text{for } A_t^C \rangle A_t^F,\\ \left(1 - \delta_{t-1}^{C\to F}\right) &= 0,5 + \lambda \quad \text{otherwise}\end{aligned} \tag{2.8}$$

Success (fitness of the rule) is represented by past profitability of the rules that are formalized in Eq. 2.9.

$$A_t^C = \left(\exp\left[P_t\right] - \exp\left[P_{t-1}\right]\right) D_{t-2}^C + dA_{t-1}^C \tag{2.9}$$

for the technical rules see Eq. 2.10.

$$A_t^F = \left(\exp\left[P_t\right] - \exp\left[P_{t-1}\right]\right) D_{t-2}^F + dA_{t-1}^F \tag{2.10}$$

for the fundamental rules. Agents use most recent performance (at the end of A^C formula resp. A^F) and also the orders submitted in period $t-2$ are executed at prices started in period $t - I$. In this way the profits are calculated. Agents have memory, which is represented by the parameter d. Values are $0 \leq \mathrm{d} \leq 1$. If $d = 0$ then agent has no memory, much higher value is, much higher influence the profits have on the rule fitness.

2.3.2 Extension of Original Model

Original model [27] has (in the authors parameterization) tendency to stabilize itself in a long term—if the fundamental trading rules are overbearing the technical trading method, although the bubbles and the crashes occur, their values are going to be smaller because the price is targeting near the fundamental value and the volatility is going to be less too.

After introduction of the transaction cost influence on the price—the price is going up to the bubble while technical traders are overtaking the market. Then possible two scenarios can occur:

- Transaction costs value is low—the price starts to be falling according the fundamental traders' weight growth. In this moment volatility falls down and the market stabilizes.
- Transaction costs value is high—fundamental traders' weight = 0, the system destabilizes and the price grows without limit.

The authors incorporated the risk into original model. The risk was implemented as a price risk percentage (*RP*) which is generated each turn from given interval according uniform random distribution <-1, 3>. So for risk influence the price formula has changed as shown in Eq. 2.11.

$$P_{t+1} = (P_t + \alpha(W_t^C D_t^C + W_t^F D_t^F) + \alpha_t)^* RP \tag{2.11}$$

Transaction costs were implemented in the same way as in previous simulations with adding constant value 0.001 to the price (Eq. 2.12).

$$P_{t+1} = (P_t + \alpha(W_t^C D_t^C + W_t^F D_t^F) + \alpha_t)^* RP + TC \tag{2.12}$$

The hypothesis was that transaction costs (Eq. 2.12) will bring the same effect to the market as in the case of pure model without risk involvement—with small amount of TC it will stabilize the market as in Figs. 2.1, 2.2, and 2.3 [28]. The agent-based simulation of the financial model implemented in Šperka and Spišák [28] has the tendency to stabilize itself in a long term, if the fundamental trading rules are overbearing the technical trading method. Although the bubbles and the crashes occur in the model, their tendencies are going to be less dangerous, because the price is targeting near the fundamental value and the volatility is going to be less. This description is similar to the current situation on the financial markets (Fig. 2.1).

By adding transaction costs (Tobin tax) to the model an observer can observe price changes. In the first situation Tobin tax was defined to be equal to 1, 5 %. The price grows up with the transaction costs to the bubble, while technical traders overtake the market. But the price starts to fall down according to the technical analysis growth. In this moment the volatility falls down and the market stabilizes. This is the main positive contribution of Tobin tax introduction into financial market in this agent-based simulation (Fig. 2.2). On the other hand when the value of the transaction

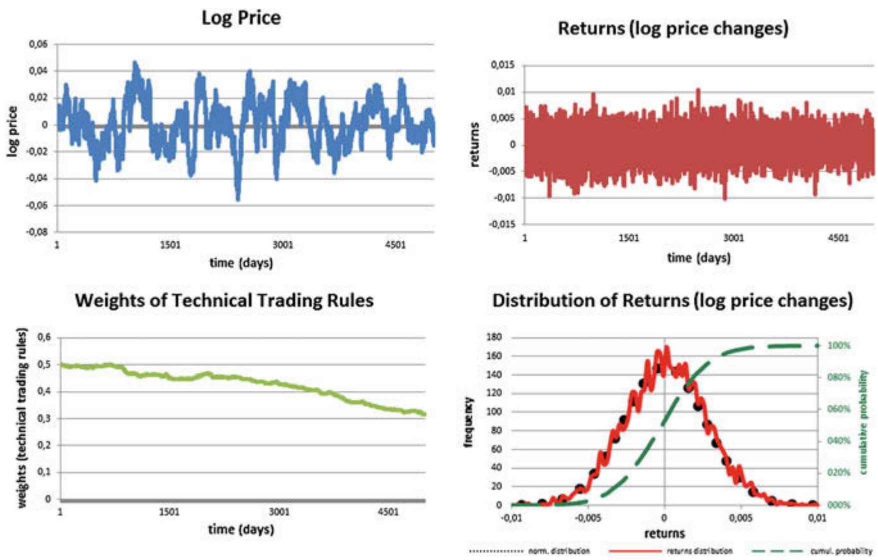


Fig. 2.1 Simulation results in original model. *Source* [28]

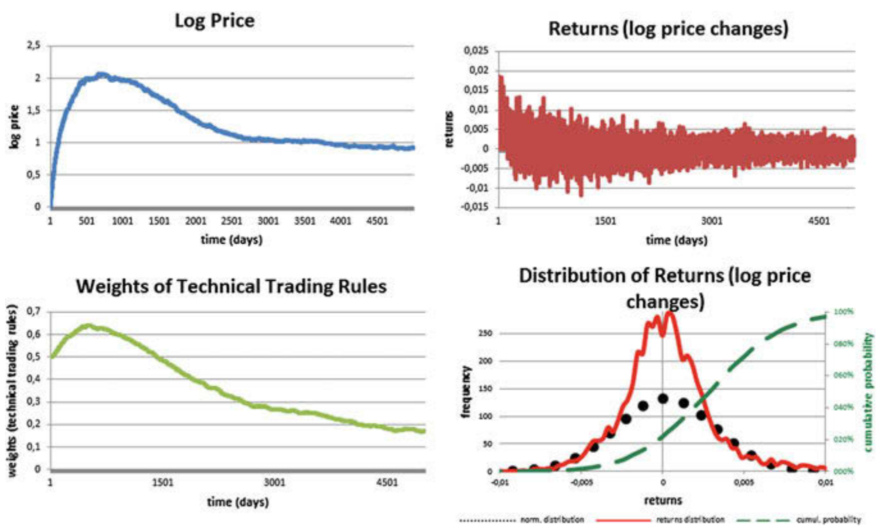


Fig. 2.2 Simulation results with transaction costs 0.015. *Source* [28]

costs is disproportionately high (3 % and higher), the system destabilizes and the price grows without limit (Fig. 2.3).

Two types of simulations were done in Sect. 2.4 using (Eq. 2.11) and (Eq. 2.12)—one only with risk percentage and the second one with transaction costs to see the difference.

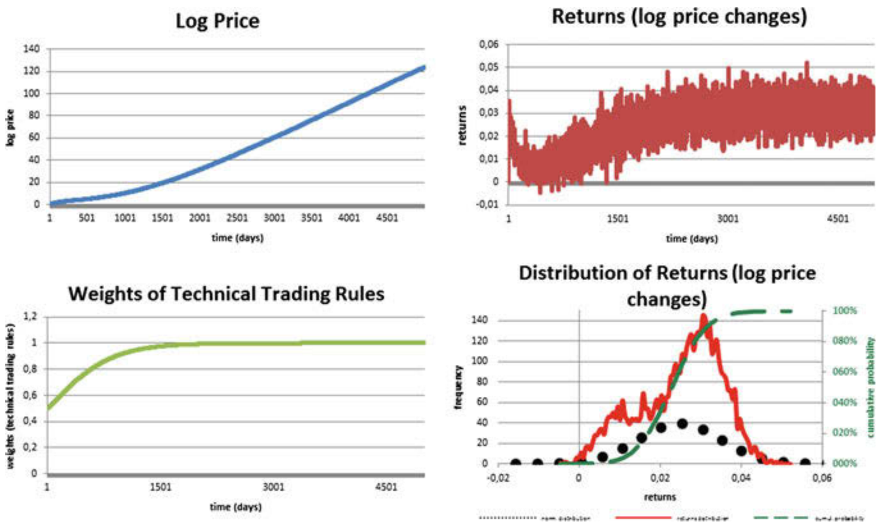


Fig. 2.3 Simulation results with higher transaction costs 0.03. *Source* [28]

2.4 JADE Implementation

JADE (Java Agent Development Framework) integrated development environment (IDE) was used for the implementation. JADE is the sophisticated solution, which is an implementation of the runtime agent environment and a communication platform. It includes runtime environment, where agents exist, libraries to write them and also graphical tools to administrate them and monitor their state. Wooldridge [29] says that it is best-known and the most widely used platform for agent modeling. Agent communication language is FIPA ACL [30].

This framework was developed by Telecom Italia in 1998 and it is still in development progress. The current version used is 4. Runtime environment running instance is called a container. At once it is possible that more than one container is running. All active containers are called a platform. Agents are located in these containers. The agent is a Java class—descendant of base JADE class Agent. Its behavior is implemented in private subclasses of the concrete Agent class extension. This behavior extends JADE class *Behaviour* [30].

Two levels of agent hierarchy were implemented. FMM base agent (*BaseFmmAgent*) contains base functionality, such as registering to the yellow pages, searching for other agents, clean-up and so on. There exist two descendants:

- **Trading agent**—represents trader in the market with his decision making.
- **Market agent**—represents market itself. Manages the turns begin and start, price making calculation, rules fitness and their weights, analysis market volatility via price differences each turn and also writes these values to result output.

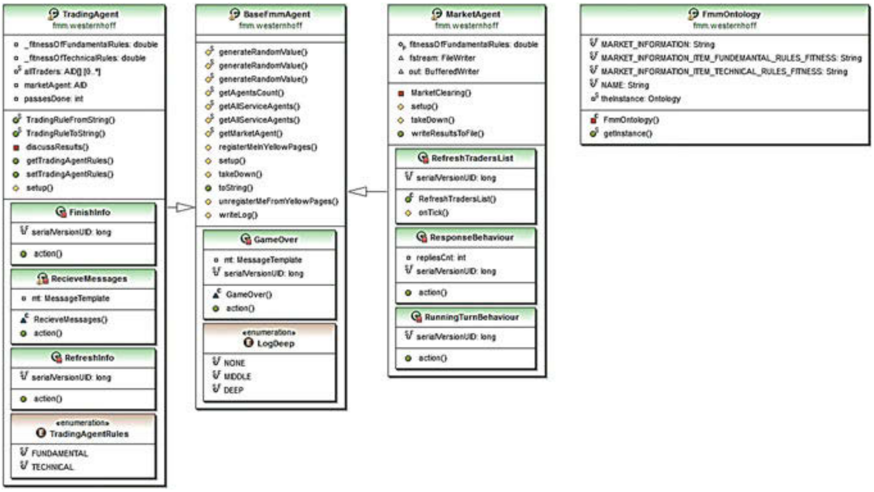


Fig. 2.4 Class model of implementation

Market agent communicates with trading agents—they are informed about the rules fitness. To understand each other, agents need ontology. "Ontology is a specification of a set of terms, intended to provide a common basis of understanding about some domain" [29]. JADE offers the base Ontology class that is extended by the implementation of singleton class *FmmOntology* (Fig. 2.4).

2.5 Simulation Results and Discussion

Simulation was done with 1 marketing agent and 500 trading agents. The rest of the parameters remained same as in original Westerhoff model [27]:

$a = 1, b = 0.05, c = 0.02, d = 0.95, \varepsilon = 0.1, \lambda = 0.45, \sigma\alpha = 0.0025, \sigma\beta = 0.025$, and $\sigma\gamma = 0.0025$

With these parameters the model is calibrated to the daily data. Number of ticks, resp. time steps is 360 days, which represents one year. Each generation (risk only and risk with transaction costs) was done 25 times. Results were aggregated to obtain more accurate results. In graphs are shown average data. Interval for price risk percentage values was decided as <–1, 3>; when 1 (as 100 %) means not changed price, the result is that price can change ±200 % each day. Results can be seen in the Fig. 2.5.

In Figs. 2.5 and 2.6 on the top left position the price values can be seen. Top right graph represents changes of the price in a time. The bottom left graph shows the weights of technical trading rules (in a long time there is a tendency to prefer fundamental than technical trading rules). Bottom right graph includes the distribution of returns (log price changes) compared with the normal distribution.

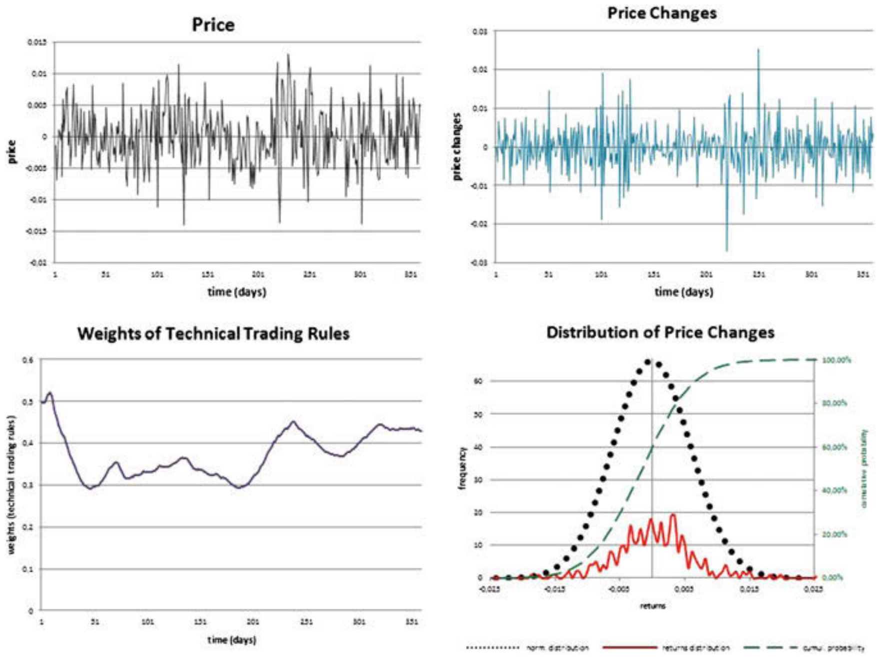


Fig. 2.5 Results for risk involvement

Fig. 2.6 Results for risk involvement together with TC

In the next step the authors added TC to the model formalization. All the parameters are the same. Newly added TC is the constant value equal to 0.001. From the following graphs in Fig. 2.6 it can be declared that transaction costs have partial influence on the model. The price refers to no changes in a time. The technical weights evolution is other. In a short time it grows, but after it starts to fall—as the agents prefer the fundamental strategy. Results are depicted in the Fig. 2.6.

2.6 Conclusion

Agent-based simulation of financial market was introduced in this chapter. Intelligent agents representing financial market participants followed fundamental and technical rules. The probability that agent switches from the fundamental to the technical behavior depends on the historic trend of asset's prices. The hypothesis for this research was based on previous simulation results proving that transaction costs influence (Tobin tax) stabilizes the financial market. The authors incorporated the risk into original model and assumed that transaction costs introduction would lead to the predominance of fundamental rules, which will automatically cause price lowering and market stability (measured by volatility in price changes).

The hypothesis was fulfilled only partially—the fundamental rules have growing tendency in time, but the prices and their differences are nearly the same in both simulations. The authors will focus on the risk and parameterization of the model in future research steps in order to prove that Tobin tax has positive impact on the stability of financial market.

Acknowledgments This work was supported by grant of Silesian University no. SGS/6/2013 "Advanced Modeling and Simulation of Economic Systems".

References

1. Sharpe, W.F.: Capital asset prices: a theory of market equilibrium under condition of risk. J. Financ. **19**, 425–442 (1964)
2. Shiller, R.J.: Irrational Exuberance. Princeton University Press, Princeton (2000)
3. Shleifer, A.: Inefficient Markets. Oxford University Press, Oxford (2000)
4. Tversky, A., Kahneman, D.: Advances in prospect theory: cumulative representation of Uncertainty. J. Risk Uncertainty **5**, 297–323 (1992)
5. Kaegi, M.: Risk analysis of information systems by agent-based modeling of business processes. Disertation, Swiss Federal Institute of Technology, Zurrich (2009)
6. Epstein, J.M., Axtell, R.: Growing Artificial Societies: Social Science from the Bottom Up. Brookings Institution Press, Washington, DC (1996). ISBN 0-262-55025-3
7. Law, A.M., Kelton, W.D.: Simulation Modeling and Analysis, 3rd edn. McGraw-Hill, New York (2000). ISBN 0070592926
8. Pritsker, A.A.B.: Introduction to Simulation and SLAM II. Wiley, New York (1995). ISBN 0470234571

9. Sallach, D.L., Macal, C.M.: Introduction: the simulation of social agents. Soc. Sci. Comput. Rev. **19**(3), 15–29 (2001). Sage Publications, Thousand Oaks, CA
10. Tesfatsion, L.: Agent-based computational economics: growing economies from the bottom up. Artif. Life 8(1), 55–82 (2002). MIT Press, Cambridge, MA
11. Conzelmann, G., North, M., Boyd, G., Cirillo, R., Koritarov, V., Macal, Ch., Thimmapuram, P., Veselka, T.: Simulating strategic market behavior using an agent-based modeling approach. In: Result of a Power Market Analysis for the Midwestern United States, 6th IAEE European Energy Conference on Modelling in Energy Economics and Policy (2004)
12. Lote, R.: Agent based modeling of electronical markets to analyze the sustainability of mutual cooperation. Master Theses, Paper 13, School of the University of Massachusetts. Industrial Enginering and operations research (2007)
13. Hayek, F.A.: Individualism and Economic Order. Routledge and Kegan Paul Ltd., London (1949)
14. Macal, C.M., North, M.J.: Tutorial on agent-based modeling and simulation part 2: how to model with agents. In: Proceedings of the Winter Simulation Conference, pp. 73–83 (2006)
15. Holland, J.H.: Adaptation in Natural and Artificial Systems. The University of Michigan, Boston (1975)
16. Kochugovindan, S., Vriend, N.J.: is the study of complex adaptive systems going to solve the mystery of adam smith's 'invisible hand'? Indep. Rev. **3**(1), 53–66 (1998)
17. Burian, J.: Multiagentní model transakčních nákladu na finančních trzích. VŠE Praha (2010)
18. Tobin, J.: A proposal for international monetary reform. East. Econ. J. **4**, 153–159 (1978)
19. Tobin, J.: The antiglobalisation movement has highjacked my name. Spiegel, Jubilee Research, Hamburg. Available from http://archive.is/CdcBN (2001)
20. Spahn, P.B.: International financial flows and transactions taxes: survey and options. In: Working Paper WP/95/60. International Monetary Fund, Frankfurt/Main (1995)
21. Wrobel, M.G.: Financial transaction taxes: the international experience and the lessons for Canada. In: Working paper BP-419E. Government of Canada Publications, Ottawa (1996)
22. Campbell, J.Y., froot, K.A.: International experiences with securities transaction taxes. In: NBER Working Paper No. W4587, pp. 277–308. University of Chicago Press, Chicago (1993)
23. Umlauf, S.: Transaction taxes and the behavior of the Swedish stock market. J. Financ. Econ. **33**(2), 227 (1993)
24. Kovacheva, R.: The EU Expects 57 Billion Euros a Year from a New Financial Tax. EU inside. Available from http://www.euinside.eu/en/news/the-eu-expects-57-billion-euros-a-year-by-a-new-bank-tax (2011)
25. Wilson, H.: Financial transaction tax would raise & #x20AC;10bn. The Telegraph, London. Available from http://www.telegraph.co.uk/finance/newsbysector/banksandfinance/9087264/Financial-transaction-tax-would-raise-10bn.html (2012)
26. Dietlein, G.: National approaches towards a financial transaction tax and their compatibility with European law. EC Tax Rev. **21**(4), 207–211 (2012)
27. Westerhoff, F.: A simple agent-based financial market model: direct interactions and comparisons of trading profits. In: Working Paper No. 61. BERG Working Paper Series on Government and Growth. Bamberg University (2009)
28. Šperka, R., Spišák, M.: Transaction costs influence on the stability of financial market: agent-based simulation. J. Bus. Econ. Manage. (Taylor & Francis, London, United Kingdom) **14**(Suppl. 1), S1–S12 (2013)
29. Wooldridge, M.: MultiAgent Systems: An Introduction, 2nd edn. Wiley, Chichester (2009)
30. Bellifemine, F., Caire, G., Trucco, T.: Jade Programmer's Guide, Java Agent Development Framework. Available from http://jade.tilab.com/doc/programmersguide.pdf (2010)

Chapter 3
Decision Support System for Energy Savings and Emissions Trading in Industrial Scenarios

Maria Marques and Rui Neves-Silva

Abstract This chapter proposes a decision support approach for energy savings and emissions trading based on the requirements collected through a set of industrial users. These requirements served as guideline for identification of needs that should be addressed with respect to the decision support approach, constituting a fundamental step for future platform development. The decision support approach proposes two different perspectives: *support for immediate reaction* (based on the paradigm of intelligent decision support implemented through the use of Case-based Reasoning together with probabilistic analysis); and *support for process reconfiguration and Emission Trading System (ETS)* (implemented through the use of multi-criteria decision analysis using MACBETH). The chapter illustrates a categorization approach using main criteria involved in the process and associated algorithms. Moreover the approaches proposed were successfully tested in industrial environment and the results obtained are here presented.

Keywords Decision Support · Energy savings · Industrial test cases

3.1 Introduction

The development of decision support methods for energy savings has been strongly oriented for the application in building energy efficiency. For this the European energy policy has contributed with its clear orientation towards the preservation of energy and the improvement of indoor environmental quality in buildings through

M. Marques (✉)
UNINOVA-FCT Campus, 2829-516 Caparica, Portugal
e-mail: mcm@uninova.pt

R. Neves-Silva
FCT-UNL-FCT Campus, 2829-516 Caparica, Portugal

J.W. Tweedale et al. (eds.), *Intelligent Decision Technology Support in Practice*, Smart Innovation, Systems and Technologies 42,
DOI 10.1007/978-3-319-21209-8_3

the adoption of the European Commission's (EC) Energy Performance of Buildings Directive [1]. To this end, in the past decades, there have been significant efforts towards designing, operating and maintaining energy efficient and environmentally conscious buildings [2]. Examples of this research can be found in [3–12].

Additionally, some research has been conducted in the development of decision support systems for environmental management [13, 14].

Some of the concepts used for energy efficiency in buildings can be applied in the industrial sector but truth is that, specificities of the problems are not the same. Thus, methods specifically developed for the industrial problem are quite few and are mainly based in modelling and simulation of the industrial facility and do not consider important aspects such as keeping the values of the decision-making criterion close to normal business conditions [15].

The overall objective of the LifeSaver project is to develop a methodology and platform to support companies in optimising their operations and enabling them to increase energy savings and decrease CO_2 emissions.

With the aim to produce an industry oriented platform the business cases defined within the project have been analysed and the end users' expectations and requests have become necessary requirements for the LifeSaver platform and have been included in the LifeSaver concept. Combining the business and technical objectives for each business cases generic LifeSaver scenario has been generated.

Also, appropriate decision support enabling companies to increase their energy savings and decrease emissions have been analysed.

3.2 General Requirements

The industrial partners have defined use cases within their companies, where they envisage the use of LifeSaver platform [16]. This also led to the identification of requirements that the industrial partners want to see fulfilled by LifeSaver. Although each industrial partner has identified individual needs, it is possible to describe common concerns. Thus, the LifeSaver industrial partners are interested in a system that will mainly provide:

- contextualized information about the energy consumption within the production process aiming to reduce this consumption, decrease related CO_2 emissions and improve raw material usage,
- prediction capability of both energy consumption and CO_2 emissions to enable the development and implementation of strategic solutions aiming at overall reduction of energy consumption and related emissions;
- decision support functionalities to support the user in finding the best options for optimization of energy and environmental performance of operations as well as selecting the most appropriate CO_2 emissions strategies within Emission Trading System (ETS);
- learning capabilities so that the LifeSaver platform based on the data from the production process can, along time, continuously contribute for the refinement of

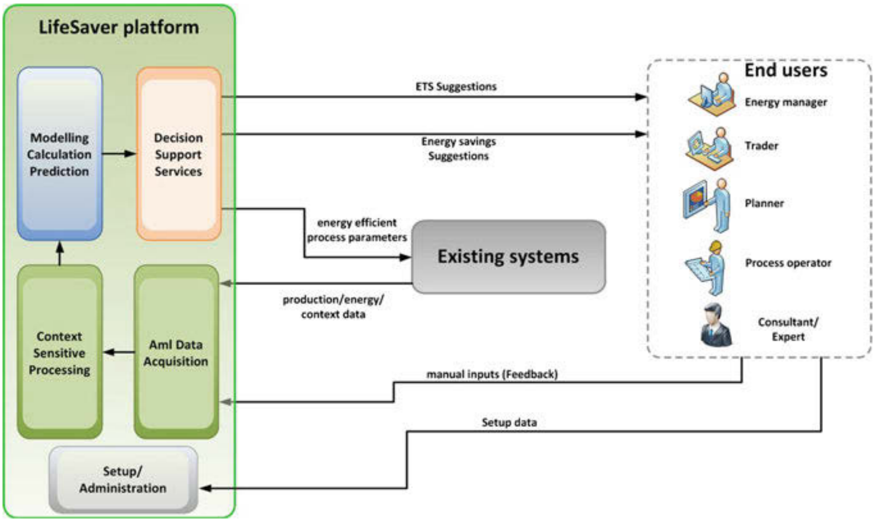


Fig. 3.1 LifeSaver generic scenario

the suggestions provided to the user and enable sustainable improvements of the production process.

The collection of requirements enabled the development of a proposition for general scenario of LifeSaver, which is presented in Fig. 3.1.

3.2.1 Requirements for Decision Support Services

Based on the scenarios built for each LifeSaver business case and taking into account common needs from other companies, a general scenario was proposed, summing up the identified requirements. Note that these requirements are particularly focused on the decision support, since this is the channel through which LifeSaver will render its results to end users. Thus, understanding expectations and needs for this service was a key issue in order to develop the LifeSaver concept.

Here a list of the collected requirements, in terms of the types of suggestions expected, is presented:

- **Suggestions on production orders**: which production orders should run (or not run) at the same time; optimal sequence to start the production orders; minimum time delay between two consecutive production orders;
- **Suggestions on energy management**: which machines are (or will be) running longer than necessary and could be switched off earlier (or switched on later); new methods/technologies that can be installed to save energy, with associated cost/benefit analysis; when the use of low tariff can reduce electricity costs;

- **Suggestions on production process parameters**: optimal settings/adjustments of process control parameters in order to improve energy efficiency and CO_2 emissions; optimal mixture of available alternative fuels for each clinker burning process without jeopardising requested/expected production quantity and stable operation of process; optimal structure of different chemical additives and mineral mixtures which could enable additional energy, CO_2 emissions and raw material savings;
- **Suggestions on Emissions Trading System**: best options (in terms of schedule, price) to buy or sell CO_2 allowances.

3.3 Approach for Decision Support

LifeSaver Decision Support Services will provide systematic mechanism for decisions on how to save energy in current operations and how to reduce emissions in a company and also on how to trade emissions.

This kind of decision support needs to congregate aspects such as risk, costs, legal issues, etc. Their primary purpose is to validate that time, effort and money invested to reduce energy usage or environmental improvement in industry provides the expected results. As a consequence of reducing the energy consumption during the production process, it is important to understand the behaviour of the EU ETS (European Union Emission Trading System) in order to plan the strategic performance of companies.

To achieve the envisaged level of decision the Decision Support Services must provide the user with:

- possibility to define "what-if" scenarios;
- identification of alternative options for energy savings, emission reductions, improved efficiency in the raw materials usage;
- identification of the best solution for emission trading;
- possibility of tracking past decisions for evaluating the results obtained.

The implementation of risk analysis and multi-criteria decision making techniques, using the information being provided by the rest of the LifeSaver building blocks, will enable the provision of the kind of decision support envisaged. Note that, since Decision Support will be "in the end" of the LifeSaver process, all information available may be valuable and useful for decision (context extracted/processed, energy models selected, prediction models selected, prediction results, energy consumed/predicted and emissions calculated/predicted).

3.3.1 Categorization of LifeSaver Decision Support

The first thing to do when analysing the characteristics of the LifeSaver process is understanding when and how decision support is needed, which determines the type of approach that must be followed. Generally speaking, LifeSaver will provide support in two different ways [17]:

- Support for immediate reaction—involves the definition of a strategy to respond to an abnormal situation that must be normalised. This approach implies the application of a corrective measure based on real-time measurements;
- Support for reconfiguration and ETS—involves the elaboration of scenarios which will then be evaluated in order to reach a decision about the best alternative. The approach is to be applied during operational phase but it is not directly interacting with production process in real time.

Due to their different characteristics the two approaches demand different methods:

- Immediate reaction approach: based on the paradigm of Intelligent Decision Support implemented through the use of Case-based Reasoning together with probabilistic analysis;
- Reconfiguration and ETS approach: Implemented through the use of Multi-Criteria Decision Analysis (MCDA), combined with simulation-based analysis when needed.

Figure 3.2 presents the categorization of LifeSaver decision support methodological approach according to their different characteristics.

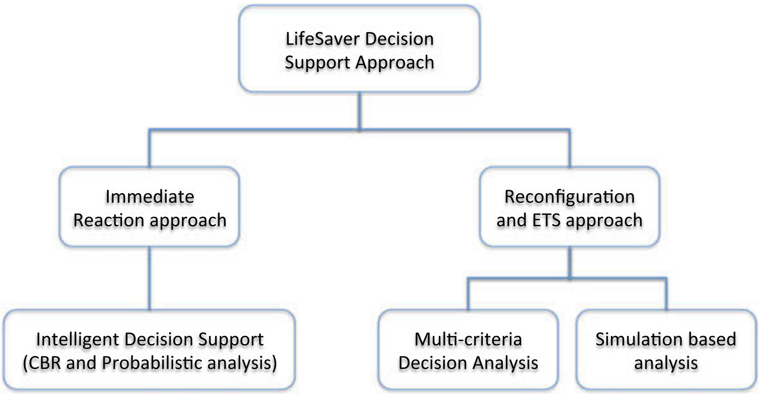


Fig. 3.2 Categorization of LifeSaver decision support methodological approaches

3.4 Methodology to Support Immediate Reaction

One of the main research lines on decision support systems has evolved towards including intelligent abilities in those systems. These systems are based on artificial intelligence or intelligent agent technologies and are commonly called Intelligent Decision Support Systems (IDSS). Their main objective is to realize decision making functions by gathering and analysing evidence, identifying and diagnosing problems, proposing possible courses of action and evaluating the proposed actions representing some human brain competences. A solution that demonstrated to be quite successful is Case-Based Reasoning (CBR). CBR systems are based on a starting set of cases that are structured in an appropriate format in order to constitute training examples. It is a problem solving approach that works by identifying commonalities between a retrieved case and the target problem.

3.4.1 Decision Model

The approach proposed in LifeSaver is to combine CBR with probabilistic analysis in order to provide the user with a brief idea of which might be the result of following a specific course of action [18]. This is made through the collection of information and data about the industrial plant along its operational phase. The idea is to collect information coming not only from the machines but also from experts that have deep knowledge on the specific production processes.

The following aspects must be covered:

- List of variables important to assess the overall energy consumption of the plant/machines;
- List of rules associated to the normal behaviour of the variables;
- List of common causes for rule violation (abnormal energy consumption);
- List of possible alternatives (actions) to deal with the causes (restore normal energy consumption).

To avoid critical energy consumption values it is considered the observation of an energy consumption threshold for each Energy Cost Center.[1] When the threshold is violated a decision point is achieved and a set of different alternatives must be considered: $\mathrm{A} = \{\mathrm{A}_1, \ldots, \mathrm{A}_\mathrm{n}\}$. The success of the applied alternative is measured in terms of time elapsed, Δt, between subsequent detections of abnormal energy consumptions for the same ECC. This time should vary accordingly with the alternative selected, and should be as long as possible to maintain the methodology objective.

In the proposed approach energy consumption is monitored and, in case there is an abnormal consumption, an event is generated. This event represents the need

[1]Energy Cost Centers (ECC) are business segments (i.e. departments, areas, units of equipment or single equipment) where activities or production volume are quantifiable and where a significant amount of energy is used.

for an action to restore the normal energy consumption level. The event triggers a search for similar cases, which will result in a set of cases, and some of them could even be caused by different causes. The selection and elimination of the cause is made by providing the user with information on possible outcomes depending on the different alternatives that may be followed. The suggested alternative is the one that presents better cost/efficiency relation, i.e. the cost associated to the number of similar decision processes that occurred along time each time a specific pair (Cause, Alternative) was selected. In the tree this cost is given by $\mathrm{Co}(\mathrm{C}_x, \mathrm{A}_x)$ and is computed in Eq. 3.1 using:

- Check how many times that alternative was selected together with that cause, $n\_DP(C_x, A_x)$:
- Compute the elapsed time between each situation; and
- Computed the medium cost.

$$\mathrm{Co}(C_x, A_x) = \frac{\mathrm{Co}(A_x) \times n\_DP(C_x, A_x)}{\Delta t(DP)} \tag{3.1}$$

where $\mathrm{Co}(A_x)$ is the cost of the alternative and $\Delta t(DP)$ is the accumulated time elapsed between decision processes that were solved with A_x.

In this approach the methodology tries to congregate the effect of each alternative in the future energy consumption of the plant by navigating in the probabilities attributed to each path. In the end, the solution will come in the form of an action to eliminate a cause, being that action the most cost-efficient for that cause, at that moment. Figure 3.3 shows the proposed decision model. The model was built following the logic of the problem, ensuring that at each probabilistic node probabilities along any outgoing branch sum one. The expected result is achieved by rolling the tree backward (i.e., starting at the bottom and working towards the root). On the tree, the value of a node can be calculated when the values for all the nodes following it are

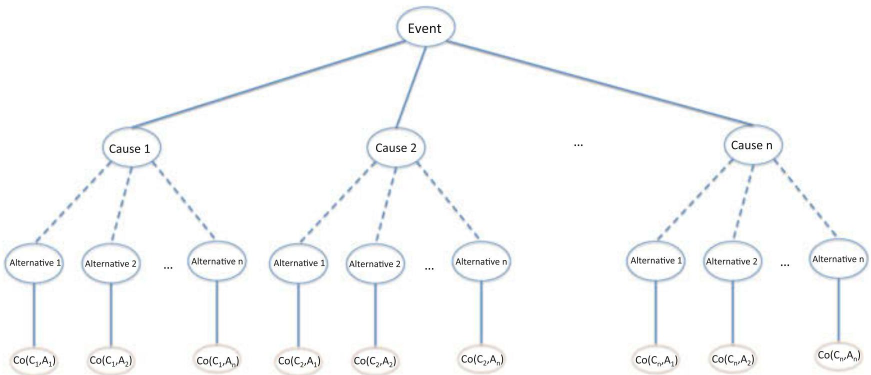


Fig. 3.3 Decision model for immediate reaction

available. The value of a node is the expected value of the nodes following it, using the probability of the arcs. The tree can grow in complexity considering the number of causes that might be involved as well as the number of possible alternatives that may be selected.

3.5 Methodology to Support Process Reconfiguration and ETS

Section 3.5.1 discusses the main criteria required to support the Reconfiguration Process, while Sect. 3.5.2 illustrates the Decision Algorithm.

3.5.1 Main Criteria for Process Reconfiguration and ETS

The criteria for energy efficiency and energy management in industry can be either quantitative or qualitative. In what concerns LifeSaver the objective of decision support for process reconfiguration is to understand how to save energy in current operations. In what regards support for ETS the idea is to support companies in taking advantage of the emission trading system. In both cases, the primary purpose of decision support is to validate that time, effort and money invested provides the expected results.

The envisaged results from the end-users point of view can be synthetized as *improving production planning and process control system by integrating energy and environmental management*. To cope with this objective a set of criteria were identified as being important for LifeSaver. Table 3.1 presents the main criteria identified for LifeSaver decision support for Process Reconfiguration and ETS as well as a number of associated sub-criteria. This list is not exhaustive and may be revised.

LifeSaver industry orientation drives into highly concrete and tangible scenarios. This results in having a set of mains criteria and sub-criteria that can be actually computed or measured. The possibility of including user preferences is considered an important feature. Thus, the proposed approach includes a mix of quantitative criteria expressed by indicators, which are weighted accordingly with user preferences. In addition to this, a methodology for quantifying attractiveness within a criteria based on the one proposed by MACBETH method [19–21] is also used. The use of the MACBETH methodology is justified by the nature of the problem, for example; very specific (quantifiable) criteria versus the need for having a straightforward method for defining (and identifying) user preferences.

Table 3.1 Main criteria for Process Reconfiguration and ETS and associated sub-criteria

	Criteria	Sub-criteria	
Criteria main categories	Energy use	EU_1	Energy consumption by time unit
		EU_2	Energy use by process
		EU_3	Energy savings by retrofitting
	Emissions	E_1	Emissions by time unit
		E_2	Emissions reduction potential
		E_3	Environmental impact
	Production	P_1	Production capacity
		P_2	Expected production
		P_3	Utilized capacity
	Cost	C_1	Direct cost and investment
		C_2	Maintenance costs
		C_3	Life span
		C_4	Emission allowance cost
		C_5	Net present value (NPV)
		C_6	Internal rate of return (IRR)

3.5.2 Decision Algorithm for Process Reconfiguration and ETS

The approach proposed for LifeSaver can be broken down into several steps.

Step 1—Scenario collection
In this step an inventory of the testing scenarios is collected based on the inputs from plant expert users. In what regards Support for Process Reconfiguration and ETS, and in the most simple case, the analysis will be made considering the "AS-IS" situation with one "TO-BE" situation.

Step 2—Definition of criteria
In order to be as exhaustive as possible and to define the question properly, particular attention must be given to the definition of criteria. On the other hand, the number of criteria must not exceed a reasonable limit. The list proposed in Sect. 4.1.1 may be used as a starting point and other criteria may be added to it.

Step 3—Analysis of the scenarios
Once the scenarios and criteria have been defined, a quantitative estimation must be made. Based on the criteria and scenarios to be evaluated, a multi-criteria evaluation matrix is build. This matrix is a table with as many columns as there are criteria and as many lines as there are scenarios to be compared (see Table 3.2).

Table 3.2 LifeSaver evaluation matrix

	Criteria	Weight (w)	Sub-criteria	Sub-weight (sw)	Scenarios			
					S_1	S_2	...	S_n
Criteria main categories	Energy use	w_{eu}	EU_1	w_{eu1}	$V1_{eu1}$	$V2_{eu1}$		Vn_{eu1}
			EU_2	w_{eu2}	$V1_{eu2}$	$V2_{eu2}$		Vn_{eu2}
			EU_3	w_{eu3}	$V1_{eu3}$	$V2_{eu3}$		Vn_{eu3}
	Emissions	w_e	E_1	w_{e1}	$V1_{e1}$	$V2_{e1}$		Vn_{e1}
			E_2	w_{e2}	$V1_{e2}$	$V2_{e2}$		Vn_{e2}
			E_3	w_{e3}	$V1_{e3}$	$V2_{e3}$		Vn_{e3}
	Production	w_p	P_1	w_{p1}	$V1_{p1}$	$V2_{p1}$		Vn_{e1}
			P_2	w_{p2}	$V1_{p2}$	$V2_{p2}$		Vn_{p2}
			P_3	w_{p3}	$V1_{p3}$	$V2_{p3}$		Vn_{p3}
	Cost	w_e	C_1	w_{c1}	$V1_{c1}$	$V2_{c1}$		Vn_{c1}
			C_2	w_{c2}	$V1_{c2}$	$V2_{c2}$		Vn_{c2}
			C_3	w_{c3}	$V1_{c3}$	$V2_{c3}$		Vn_{c3}
			C_4	w_{c4}	$V1_{c4}$	$V2_{c4}$		Vn_{c4}
			C_5	w_{c5}	$V1_{c5}$	$V2_{c5}$		Vn_{c5}
			C_6	w_{c6}	$V1_{c6}$	$V2_{c6}$		Vn_{c6}

where Eq. 3.2 provides the weightings.

$$w = w_{eu} + w_e + w_p + w_c = 1 \tag{3.2}$$

and the sum of each sub-weighting, associated to each criteria, must also be one.

Step 4—Evaluation of scenarios in terms of each of the selected criteria

The scenarios are scored for each criteria. The values obtained by each scenario regarding each criteria are mapped into an interval from [0,1] (see Eq. 3.3).

$$V_{S_{SW}} = \frac{ActualValue - MinimumValue}{MaximumValue - MinimumValue} \tag{3.3}$$

where ActualValue—represents the absolute value of the scenario for that criteria; MinimumValue—represents the minimum (worst) accepted value for that criteria; MaximumValue—represents the maximum (best) accepted value for that criteria.

This way the authors avoid regular normalization which is totally blind to user preferences.

Note that the methodology also accepts qualitative description of the impact of each scenario, in terms of criteria, (impact descriptors that include: neutral and significant). In these cases an additional mapping between the descriptors and the values. As this case is not foreseen within LifeSaver it is not developed here.

The overall value obtained by each scenario is calculated using Eq. 3.4.

$$OV_n = \sum_{i=1}^{m} w_i \left(\sum_{j=1}^{k} sw_j \times V_{n_{sw_j}} \right) \tag{3.4}$$

with n—the number of scenarios; m—the number of criteria; k—the number of sub-criteria inside each criteria. The overall values obtained for each scenario will be in [0,1] with the best scenario being the one presenting higher score i.e. closer to 1.

Step 5—Sensitivity analysis

It is also important to understand how variations on parameters or variations on the criteria weights affect the results obtained. Basically this corresponds to applying some sort of sensitivity analysis on the final result, for example a "what-if" question to see what happens when some modification is introduced in any of these inputs and to what extent they affect the output. This will contribute for an increased understanding of the relationships between input and output variables in the decision process.

Sensitivity analysis in LifeSaver relies on running the energy models developed, taking into account the level of uncertainty measured in the different parameters of the model, and considering the "what-if" question introduced.

Step 6—Store results

In the end of the process, results are stored in such way that enables the identification of the alternative selected. The alternative is selected by the user(s), for this authorised users will have to be identified within each company. Storing the results will enable traceability of the decision process as well as analysis of the outcome of choosing a specific path.

3.6 Test Cases

Section 3.6.1 discusses the Decision Support for immediate reactions, while Sect. 3.6.2 explores the Process for Reconfiguration of the ETS during Paint Production.

3.6.1 Test Case 1: Decision Support for Immediate Reaction at Cement Production

Cement production is one of the most energy intensive processes in industry, causing large amounts of CO_2, dust and other emissions, noise (quarry). According to [22], global cement industry accounts for about 70–80 % of the energy use in the non-metallic minerals sub-sector, consuming 8.2 exajoules (EJ) of energy a year (7 % of total industrial fuel use—year 2005) and accounts for almost 25 % of total direct CO_2 emissions in industry. LifeSaver platform is being tested via a prototype information system in the largest cement-producer in Slovenia, Salonit Anhovo (Salonit). Salonit has provided a real testing environment for the validation of the proposed concept of decision support for immediate reaction.

Table 3.3 Data collection format for Salonit test scenario

PS	T	X1	X2	X3	X4	X5	X6	X7	X8	AS1	AS2
ps_a	t_a	x_{1aa}	x_{2aa}	x_{3aa}	x_{4aa}	x_{5aa}	x_{6aa}	x_{7aa}	x_{8aa}	as_{1aa}	as_{2aa}
	t_b	x_{1ab}	x_{2ab}	x_{3ab}	x_{4ab}	x_{5ab}	x_{6ab}	x_{7ab}	x_{8ab}	as_{1ab}	as_{2ab}
	t_c	x_{1ac}	x_{2ac}	x_{3ac}	x_{4ac}	x_{5ac}	x_{6ac}	x_{7ac}	x_{8ac}	as_{1ac}	as_{2ac}
Factory benchmark		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	as_1	as_2

For testing purposes Salonit provided data related to rotary clinker furnace. Note that, since the alternatives provided by Salonit did not include associated costs, the algorithm was tested without that perspective. For the initial testing the data samples were collected on a one minute interval for the period of seven days, which resulted in 10080 samples of each input signal. Testing days were carefully selected assuring that all used fuels were covered, meaning that each individual fuel has been used at least once. Data was collected using the format presented in Table 3.3 where (PS) represents the production stage, (T) represents the control room team, (X1, X2, X3, X4, X5, X6, X7, X8) represent the energy carriers flow from which fuel mix is derived, and (AS1, AS2) represent auxiliary systems.

LifeSaver compares the actual energy consumption data with the predicted values and benchmark values for the team and factory. Also, the system compares the predicted energy consumption data with the benchmark values for the team and factory. In case a deviation is detected (value higher than the benchmark value for the team and factory) the system generates Event 1 "Deviations on Fuel Mix" and a Decision Process with status "CREATED" is stored at the database.

The Decision Support Services BB initiates the support for immediate reaction(s) and finds 50 Decision Processes stored at the database with status "CLOSED". All these Decision Processes are related with situations of abnormal energy consumption. The degree of similarity between cases is established through the Event detected (which congregates the information presented in Table 3.3) and is computed in Eq. 3.5.

$$Sim(E_1, E_2) = \sum_{i=1}^{n} w_i e^{-(\gamma|\alpha_{1i} - \alpha_{2i}|)} \tag{3.5}$$

where α_{1i} and α_{2i} are the parameters of each Event, w_i is the weight of each parameter and γ is a scaling factor. The computation of similarity, and the subsequent selection of cases, is performed using a similarity threshold of 90 % (all the cases with similarity level below 90 % are discarded). The set of similar cases includes 23 Decision Processes. From these, six are associated to Cause 1 "Problems with fuel sensor" and 17 are associated to Cause 2 "Unbalanced fuel dosing", thus: $p(C_1|E_1) = 0.26$ and $p(C_2|E_1) = 0.74$.

All the cases associated to Cause 1 were solved using Alternative 1 "Reset or Replace sensor", whereas half cases associated to Cause 2 were solved using Alternative 2 "Reduce dose rate of X1, confirm dosing of X2 and check the status of the auxiliary systems" and the other half using Alternative 3 "Confirm dosing of X4, X5, X6 and check the status of the auxiliary systems", thus: $p(A_1|(C_1|E_1)) = 1$, $p(A_2|(C_2|E_1)) = 0.5$ and $p(A_3|(C_2|E_1)) = 0.5$.

The cases were grouped considering the Cause and the Alternative and the mean similarity was computed for each sub-set of cases, resulting in: $\overline{Sim_1} = 0.9308$, $\overline{Sim_2} = 0.9180$ and $\overline{Sim_3} = 0.9702$.

The final score for each is then computed using (1): $Score_1 = 0.2428$, $Score_2 = 0.3392$ and $Score_3 = 0.3585$.

After this process the alternative recommended to the user is Alternative 3 (due to the higher score obtained). Nonetheless, all the alternatives considered are presented to the user. The user selects and implements one of the possible alternatives and closely monitors the clinker burning process. The system stores the user selection and changss the status of the Decision Process from "CREATED" to "SELECTED". In case no new deviations are detected in a specific time frame the user informs the system that Alternative 3 was successful and the status of the Decision Process is changed to "CLOSED". From now on, this Decision Process will be usable in future situations. This action serves to measure the success of the suggestion provided since.

3.6.2 Test Case 2: Decision Support for Process Reconfiguration and ETS at Paint Production

LifeSaver prototype is also being tested at J.W. Ostendorf GmbH & Co. KG (JWO), a large producer and supplier of paints and varnishes. JWO has provided a real testing environment for the validation of the proposed concept of decision support for process reconfiguration and ETS.

For testing purposes JWO provided data related to production of white dispersion paint, which can be produced in 10 different production lines. Each production line is equipped with different dispersers, resulting in varying energy consumption for the production of the same product. Hence the selection of the most energy efficient production line will represent an appropriate means to reduce energy consumption in the process. Nonetheless, and due to process restrictions, there are other types of paints that can only be produced in specific lines. Note that, in this test scenario, JWO is not considering Emissions or Costs. In fact the scenarios are scored considering the utilization of the installed capacity, using only electric energy, which will be, in the end, the main selection point.

Table 3.4 Data collection for JWO test case

PL Scenario	1	2	3	4	5	6	7	8	9	10
Scenario 1	pt_1	pt_2	pt_3	pt_4	pt_5	pt_6	pt_7	pt_8	pt_9	pt_{10}
Scenario 2	pt_2	pt_7	pt_8	pt_{10}	pt_3	pt_5	pt_4	pt_1	pt_9	pt_6
Scenario 3	pt_3	pt_9	pt_7	pt_6	pt_{10}	pt_5	pt_1	pt_8	pt_4	pt_2

The testing data included the elaboration of three different scenarios, representing 10 different orders (including white dispersion paint and other 9 types of paint) and aiming at finding the best selection of production lines to produce the orders. The scenarios were elaborated with the support of JWO experts resulting in Table 3.4: where PL represents the production line, and pt_i represents the type of paint to be produced (including the viscosity and the recipe, with impact in the energy consumption). The scenarios are simulated using the LifeSaver prediction capabilities and the results are used for scoring. Table 3.5 presents the options made in terms of criteria weight, acceptable intervals as well as the normalised results obtained for each scenario in each category and the overall score.

Once the process is completed the system presents the results ranking the scenarios by the score obtained. In this case Scenario 3 is considered the most adequate one, followed by Scenario 2 and, in the last position, Scenario 1. In case the user wants to start a production order this information supports the selection of the most energy efficient production lines combination.

Table 3.5 LifeSaver matrix for JWO test scenario

Criteria	Energy use		Production	
Weight	0.8		0.2	
Sub-criteria	EU_1	EU_2	P_3	
Sub-weight	0.7	0.3	1	
Acceptance interval	$[minEU_1; maxEU_1]$	$[minEU_2; maxEU_2]$	$[minP_3;\ maxP_3]$	
				OV
Scenario 1	$V_1(eu_{11}) = 0.5$	$V_1(eu_{21}) = 0.6$	$V_1(p_{31}) = 1$	0.624
Scenario 2	$V_2(eu_{12}) = 0.6$	$V_2(eu_{22}) = 0.6$	$V_2(p_{32}) = 1$	0.68
Scenario 3	$V_3(eu_{13}) = 0.5$	$V_3(eu_{23}) = 0.5$	$V_3(p_{33}) = 1$	0.7

where:

$$V_i(eu_{1i}) = \frac{eu_{1i} - minEU_1}{maxEU_1 - minEU_1}, i = 1, 2, 3$$
$$V_i(eu_{2i}) = \frac{eu_{2i} - minEU_2}{maxEU_2 - minEU_2}, i = 1, 2, 3$$
$$V_i(p_{3i}) = \frac{p_{3i} - minp_3}{maxp_3 - minp_3}, i = 1, 2, 3$$

The scenarios and related score are stored in the knowledge repository associated to a Decision Process that can, at any time, be accessed. The quality of the result is measured by the feedback of the user in terms of “degree of satisfaction” with the suggestion provided.

3.7 Discussion

For Test Case 1 the main objective is overall performance improvement by eliminating, or at least reducing, behaviours that lead to increased energy consumption. The proposed approach combined with the new hardware and software and introduction of postulates of energy and environmental management into Salonit’s daily activities represents introduction of change in people’s attitude about energy use in daily operational practices and routines.

The results of the decision model have confirmed the possibility of using such a model for responding to energy consumption deviations that require an immediate reaction to restore normal situation. Due to availability of the reliable historical/past data the results obtained with the decision model were appropriate for the situations detected and considered excellent.

During the testing phase predicted values of the energy consumption (for each particular fuel and electricity) were compared with the context specific benchmark values (for each team and fuel mix) to facilitate understanding of energy use patterns and trigger early warning and reaction if needed. Initial testing results have confirmed the potential of energy savings enabled by proper and tailor made consumption feedback through decision support services which successfully influenced on the established behavioural patterns of less efficient process operators.

In what concerns Test Case 2 the main objective is to support JWO operators in finding strategies to enhance energy efficiency. The results obtained have confirmed the possibility of the proposed approach to provide information on the best scenario production line selection. During the testing phase predicted values of the energy consumption were compared and information regarding contextual variables was considered (specific viscosity of the paint). Initial testing results have confirmed the potential of energy savings enabled by suggesting the most energy efficient production line combination for each production order.

In both test cases, limitations of the proposed concept are related with the requirements for expert knowledge during modelling period and definition of the initial set of context sensitive information. Also, context extraction from the available history data requires a significant amount of time, efforts and experience, especially when there is an intention to involve variables that will contextualize complex industrial operations.

3.8 Conclusions

The work presented here is adequate for LifeSaver objectives in the field of decision support. More specifically, the decision support approach proposes two different perspectives: *support for immediate reaction* involving the definition of a strategy to respond to an abnormal situation that must be normalised, and *support for process reconfiguration and Emission Trading System (ETS)* involving the elaboration of scenarios which will then be evaluated in order to reach a decision about the best alternative.

This categorization served as base for defining the methods to be applied. The *support for immediate reaction* is based on the paradigm of intelligent decision support implemented through the use of Case-based Reasoning together with probabilistic analysis. *Process reconfiguration and ETS* is implemented through the use of multi-criteria decision analysis using MACBETH. The identification of main criteria and sub-criteria is provided as well as the high level algorithms for both approaches.

The results obtained within the test cases are in line with the envisaged ones and provide excellent perspectives for the future utilization of LifeSaver in diverse applications.

Acknowledgments Authors express their acknowledgement to the consortium of LifeSaver project, *Context sensitive monitoring of energy consumption to support energy savings and emissions trading in industry* [http://www.lifesaver-fp7.eu] funded under the 7th Research FP of the European Union (contract FP7-ICT-7-287652).

References

1. European Community: Directive 2002/91/EC of the European Parliament and of the Council of 16 December 2002 on the Energy Performance of Buildings, L1/65, Official Journal of the European Communities, 2003
2. Kolokotsa, D., Diakaki, C., Grigoroudis, E., Stavrakakis, G., Kalaitzakis, K.: Decision support methodologies on the energy efficiency and energy management in buildings. Adv. Build. Energy Res. **3**, 121–146 (2009)
3. Rutman, E., Inard, C., Bailly, A., Allard, F.: A global approach of indoor environment in an air-conditioned office room. Build. Environ. **40**(1), 29–37 (2005)
4. Martinaitis, V., Rogoz ∨ a A., Bikmaniene, I.: Criterion to evaluate the "twofold benefit" of the renovation of buildings and their elements. Energy Build. **36**(1), 3–8 (2004)

5. Martinaitis, V., Kazakevicius, E., Vitkauskas, A.: A two-factor method for appraising building renovation and energy efficiency improvement projects. Energy Policy **35**(1), 192–201 (2007)
6. Ma, Z., Wang, S., Xu, X., Xiao, F.: A supervisory control strategy for building cooling water systems for practical and real time applications. Energy Convers. Manage. **49**(8), 2324–2336 (2008)
7. Kolokotsa, D.: Artificial intelligence in buildings: a review on the application of fuzzy logic. Adv. Build. Energy Res. **1**, 29–54 (2007)
8. Kaklauskas, A., Zavadskas, E.K., Raslanas, S.: Multivariant design and multiple criteria analysis of building Refurbishments. Energy Build. **37**(4), 361–372 (2005)
9. Doukas, H., Patlitzianas, K.D., Iatropoulos, K., Psarras, J.: Intelligent building energy management system using rule sets. Build. Environ. **42**(10), 3562–3569 (2007)
10. Chen, Z., Clements-Croome, D., Hong, J., Li, H., Xu, Q.: A multi-criteria lifespan energy efficiency approach to intelligent building assessment. Energy Build. **38**(5), 393–409 (2006)
11. Wong, J.K.W., Li, H.: Application of the analytic hierarchy process (AHP) in multi-criteria analysis of the selection of intelligent building systems. Build. Environ. **43**(1), 108–125 (2008)
12. Campos, A., Neves-Silva, R.: Decision on the best retrofit scenario to maximize energy efficiency in a building, IDT—Smart Innovation Syst. Technol. **10**(Part IV), 853–862 (2011)
13. De Benedetto, L., Klemeš, J.: The environmental performance strategy map: an integrated LCA approach to support the strategic decision-making process. J. Clean. Prod. **17**(10), 900–906 (2009)
14. Khalili, N., Duecker, S.: Application of multi-criteria decision analysis in design of sustainable environmental management system framework. J. Clean. Prod. **47**, 188–198 (2013)
15. Moya, J., Pardo, N.: The potential for improvements in energy efficiency and CO_2 emissions in the EU27 iron and steel industry under different payback periods. J. Clean. Prod. **52**, 71–83 (2013)
16. LifeSaver Consortium: *LifeSaver Concept*, LifeSaver project: Context sensitive monitoring of energy consumption to support energy savings and emissions trading in industry, Gran Agreement no: FP7-ICT-2011-7-287652, 2012
17. Marques, M., Neves-Silva, R.: Approach for decision support for energy savings and emissions trading based on industrial requirements. IDT—Front. Artif. Intell. Appl. **255**, 74–83 (2013)
18. Marques, M., Neves-Silva, R.: Risk Based decision support system for life cycle management of industrial plants. In: Proceedings of IEEE 9th International Conference on Industrial Informatics, Caparica, Portugal (2011)
19. Bana e Costa, C., Vansnick, J., : General overview of the MACBETH approach. In: Pardalos, P.S. (ed.) Advances in Multicriteria Analysis, pp. 93–100. Kluwer Academic Publishers, Dordrecht (1995)
20. Bana e Costa, C., Vansnick, J.: MACBETH—an interactive path towards the construction of cardinal value functions. Int. Trans. Oper. Res. **1**(4), 489–500 (1994)
21. Bana e Costa, C.A., De Corte, J.-M., Vansnick, J.-C.: On The Mathematical Foundations of Macbeth, The London School of Economics and Political Science, ISBN 0-7530-1525-0, 2004
22. Harris, J., Anderson, J., Shafron, W.: Investment in energy efficiency: a survey of Australian firms. Energy Policy. **28**(12), 867–876 (2000)

Chapter 4
A Parsimonious Radial Basis Function-Based Neural Network for Data Classification

Shing Chiang Tan, Chee Peng Lim and Junzo Watada

Abstract The radial basis function neural network trained with a dynamic decay adjustment (known as RBFNDDA) algorithm exhibits a greedy insertion behavior as a result of recruiting many hidden nodes for encoding information during its training process. In this chapter, a new variant RBFNDDA is proposed to rectify such deficiency. Specifically, the hidden nodes of RBFNDDA are re-organized through the supervised Fuzzy ARTMAP (FAM) classifier, and the parameters of these nodes are adapted using the Harmonic Means (HM) algorithm. The performance of the proposed model is evaluated empirically using three benchmark data sets. The results indicate that the proposed model is able to produce a compact network structure and, at the same time, to provide high classification performances.

Keywords Radial basis function neural network · Adaptive resonance theory · Harmonic mean algorithm · Classification

4.1 Introduction

Artificial Neural Networks (ANNs) are one of the popular machine-learning approaches for learning information from data samples and for performing data classification. Over the years, a variety of ANN models have been introduced. A number of improvements to conventional ANN models have also been proposed, which include variants of the multilayer perception networks (Widrow et al. [20]), Radial Basis Function (RBF) networks (Fernández-Navarro et al. [8]), and Adaptive Resonance Theory (ART) networks (Xu and Wunsch [21]).

S.C. Tan (✉)
Multimedia University, Cyberjaya, Malaysia
e-mail: sctan@mmu.edu.my

C.P. Lim
Centre for Intelligent Systems Research, Deakin University, Geelong, Australia

J. Watada
Waseda University, Tokyo, Japan

J.W. Tweedale et al. (eds.), *Intelligent Decision Technology Support in Practice*, Smart Innovation, Systems and Technologies 42,
DOI 10.1007/978-3-319-21209-8_4

In general, the learning process of an ANN is driven by training data samples, whereas its network parameters (weights) and structure can be adjusted to absorb and generalize the learned information for providing useful solutions to a problem. In general, the learning process in an ANN can follow in a number of approaches, constructive Martínez-Rego et al. [13], destructive (Ni and Song [14]; Reed [17]), or a combination of both (Yang and Chen [22]; Han and Qiao [9]). When an ANN undergoes a constructive learning process, the hidden nodes of its hidden layer are increased from scratch, and this leads to the growth of its network structure incrementally. However, the nodes added into an ANN with constructive learning may not be trained properly [3]. Conversely, when an ANN employs a destructive learning approach [14, 17], less useful nodes can be pruned from its structure, which is initially set to a large size. However, as stated in Han and Qiao [9] and Yang et al. [23], the use of a pruning algorithm could be computational expensive because less useful network parameters need to be removed iteratively from a large ANN structure. The combination of constructive-destructive approaches [9, 22], therefore, provide a compromise to determine a parsimonious ANN structure in a flexible manner.

In this study, the parsimonious structure of an ANN is achieved by an alternative idea to that of the constructive-destructive approach. For this, a type of RBF network trained by a dynamic decay adjustment algorithm (known as RBFNDDA) is selected. On one hand, the RBFNDDA network can undergo fast and constructive learning, and can perform with good generalization. However, its learning algorithm is prone to a greedy insertion behavior. During the training process, the network encodes information of all data samples, which include noisy samples, as prototypes (hidden nodes) in its structure. Therefore, the resulting RBFNDDA network is large in size, with its hidden layer contains many "small" prototypes. To reduce the number of prototypes in RBFNDDA, they can be grouped using a classifier, such as; the Fuzzy ARTMAP (FAM) [4] network, and the corresponding parameters can be adapted with a harmonic mean algorithm. FAM is chosen because it is an incremental learning ANN that overcomes the *plasticity-stability* dilemma [4]. FAM provides a high compression capability pertaining to information that it has learned from data samples [4]. Note that FAM organizes information in terms of hyper-rectangles while information encoded in RBFNDDA prototypes is in the form of radial basis functions. In the authors proposed model, FAM is not used to update the information of these prototypes directly. Instead, it is employed to classify all centroids of the RBFNDDA prototypes into different labeled clusters in the data space. Then, a k-harmonic mean (k-HM) algorithm [24] is applied to estimate a representative centroid for a group of RBFNDDA prototypes in each labeled cluster. The resulting model is known as RBFNDDA-FAMHM. For clarity, RBFNDDA-FAMHM does not remove explicitly information encoded in the initial RBFNDDA prototypes from its network. Instead, it separates and adapts all RBFNDDA prototypes into a smaller group of clusters, which can lead to formation of a parsimonious network structure.

The remaining sections of this chapter are as follows. In Sect. 4.2, the proposed RBFNDDA-FAMHM model and its constituents are explained. In Sect. 4.3, three benchmark data sets from the UCI Machine Learning Repository [1] are used to evaluate the performance of RBFNDDA-FAMHM, with the results analyzed, compared, and discussed. Concluding remarks are presented in Sect. 4.4.

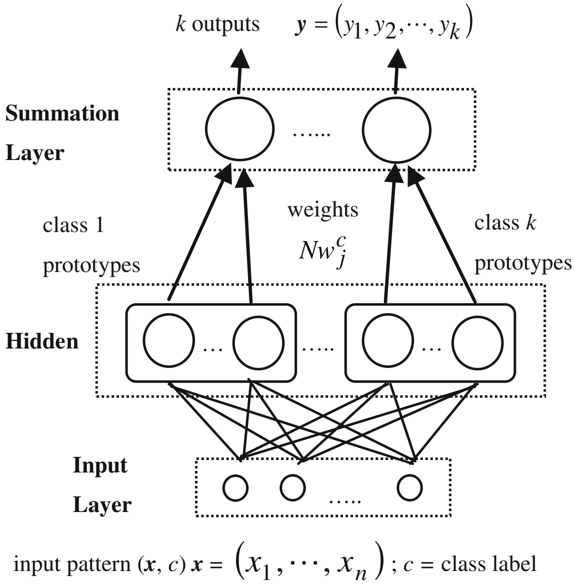
4.2 The Methodology

In this section, the algorithms of RBFNDDA, FAM, and HM are explained first before describing the overall proposed RBFNDDA-FAMHM model.

4.2.1 Radial Basis Function Network with the Dynamic Decay Adjustment Algorithm (RBFNDDA)

RBFNDDA combines salient features of the Probabilistic Neural Network (PNN) [19] and the Restricted Coulomb Energy Network (RCEN) [10, 18] to produce an adaptive ANN model. It integrates the probabilistic characteristics of the static PNN model and the ability of the RCEN in adjusting the prototype (hidden neuron) width to formulate an incremental network. Figure 4.1 shows the RBFNDDA structure.

Fig. 4.1 The RBFNDDA network structure



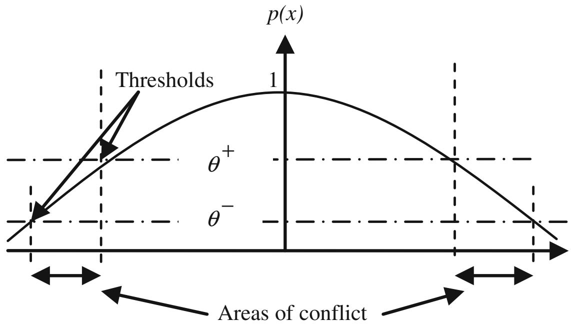


Fig. 4.2 The two user-defined thresholds of a radial basis function node [2]

The learning process of RBFNDDA is governed by two user-defined parameters, i.e., θ^+ and θ^- (see Fig. 4.2). These two thresholds are required to adjust the width of a prototype, and distinguish the influence of the prototype from its neighbours (prototypes) of other classes. On presentation of a training instance to RBFNDDA, θ^+ imposes the minimum correct-classification probability for the correct class, whereas θ^- imposes the highest probability tolerable with respect to an incorrect class. As can be seen in Fig. 4.2, the intermediate region between the two thresholds is an area of conflict which is inhabitable by any training instances. In other words, θ^+ and θ^- control the size of the overlapping region of each prototype in RBFNDDA. Berthold and Diamond [2] and Paetz [15] suggested that $\theta^+ = 0.40$ and $\theta^- = 0.20$ could serve as the default setting for the RBFNDDA network to achieve satisfactory classification results.

RBFNDDA employs the DDA algorithm to perform constructive learning, whereby new prototypes are introduced to encode new information from the data samples. Figure 4.3 shows the learning procedure of RBFNDDA in a single epoch. All the prototype weights are initialized to zero at the beginning of the training session. The rationale is to prevent accumulation of duplicated information related to the training samples. When a set of training samples is provided, they are presented to the network one by one. If the sample is correctly classified by a few existing prototypes, the weight of the largest prototype is increased. However, if the training sample is incorrectly classified, a new prototype is introduced to encode the sample in the network. In this case, the training sample is assigned as the centroid of the new prototype and its weight is set to one. The initial width of the prototype is determined in such a way to avoid overlapping with prototypes of other classes nearby. Then, width shrinking for all prototypes of different classes will be executed if their activations caused by the training sample are above θ^-. For additional details pertaining to RBFNDDA, readers can refer to Berthold and Diamond [2] and Paetz [15].

Fig. 4.3 The learning algorithm of RBFNDDA in one epoch [2]

// reset weights
FORALL prototypes p_j^c **DO**
$\quad w_j^c = 0$
ENDFOR
// a training procedure in one epoch
FORALL training patterns $(\boldsymbol{x}, c)$ **DO**:
$\quad$**IF** $\exists p_j^k : p_j^k(\boldsymbol{x}) \geq \theta^+$ **THEN**
$\quad\quad Nw_j^c = Nw_j^c + 1$
$\quad$**ELSE**
$\quad$// introduce a new prototype (hidden neuron)
$\quad m_c = m_c + 1$
$\quad Nw_{m_c}^c = 1$
$\quad z_{m_c}^c = x$
$\quad$// adapt radii

$$r_{m_c}^c = \min_{\substack{s \neq c \\ 1 \leq j \leq m_s}} \left\{ \sqrt{-\frac{\left\| z_j^s - z_{m_c}^c \right\|^2}{\ln \theta^-}} \right\}$$

$\quad$**END**
// shrink radii of conflicting prototypes
$\quad$**FORALL** $s \neq c,\ 1 \leq i \leq m_s$ **DO**

$$r_i^s = \min \left\{ r_i^s, \sqrt{-\frac{\left\| z_j^s - z_{m_c}^c \right\|^2}{\ln \theta^-}} \right\}$$

$\quad$**END**
END

4.2.2 Fuzzy ARTMAP (FAM)

FAM is an ANN model comprising two Fuzzy ART modules being connected through a mapping field [4]. The two Fuzzy ART modules respectively referred to as an input module (labeled as Fuzzy ART$_a$) and an output module (labeled as Fuzzy ART$_b$).

FAM undergoes a supervised incremental training session whereby it self-organizes and self-stabilizes information from a data set in its structure.

To embark on a supervised learning process, Fuzzy ART$_a$ receives the input pattern whereas the output module receives the corresponding output class. The input pattern is complement-coded [4] to avoid the category proliferation problem. The activation of each prototype (node) j is measured using Eq. 4.1.

$$T_j = \frac{\left|\boldsymbol{A} \wedge \boldsymbol{w}_j^a\right|}{\chi_a + \left|\boldsymbol{w}_j^a\right|} \tag{4.1}$$

where $\boldsymbol{A}$ is the complement-coded input pattern, χ_a is the choice parameter in a small positive setting close to zero [5]; $\boldsymbol{w}_j^a$ stands for the weight of node j. FAM adopts a winner-take-all scheme for node selection. In this regard, the node with the highest activation (represented as J) is picked out as the winner. Then, a vigilance test is carried out to compare the similarity degree between the winning prototype, $\boldsymbol{w}_J^a$, and $\boldsymbol{A}$ against a threshold (vigilance parameter, ρ_a) [4] (Eq. 4.2).

$$\frac{\left|\boldsymbol{A} \wedge \boldsymbol{w}_J^a\right|}{|\boldsymbol{A}|} \geq \rho_a \tag{4.2}$$

If the vigilance test fails, a new search cycle is triggered to find another winning node. The process of finding a winning node is started again until the selected winning node could pass in the vigilance test. However, when no such node exists, a new node is created to encode the input pattern. The output module of FAM (i.e., Fuzzy ART$_b$) also undergoes the same pattern-matching procedure to select a winning node that represents the target class. After both modules have identified their winning nodes, a map-field vigilance test [4] is carried out using Eq. 4.3.

$$\frac{\left|\boldsymbol{y}^b \wedge \boldsymbol{w}_J^{ab}\right|}{\left|\boldsymbol{y}^b\right|} \geq \rho_{ab} \tag{4.3}$$

where $\boldsymbol{y}^b$ is the weight vector of the winning node from the Fuzzy ART$_b$ to the map field; $\boldsymbol{w}_J^{ab}$ stands for the weight vector of the winning node from the Fuzzy ART$_a$ to the map field; and ρ_{ab} indicates the map-field vigilance parameter. If the map-field vigilance test fails, it indicates the winning node from the input module could not manage to predict for the target class correctly. Under this circumstance, a matching-tracking process [4] is carried out to increase the vigilance parameter, as shown in Eq. 4.4.

$$\rho_a = \frac{\left|\boldsymbol{A} \wedge \boldsymbol{w}_J^a\right|}{|\boldsymbol{A}|} + \delta \tag{4.4}$$

where δ is a small positive value. The rationale of increasing ρ_a is to disqualify the current winning node through the vigilance test. Consequently, a new round of search for another winning node in the input module is commenced with a new setting of ρ_a. This process is repeated until the winning nodes from both the input and output modules has made a correct prediction. Then, FAM is in a resonance state when the weight of the winning node is adjusted [4], as shown in Eq. 4.5.

$$\boldsymbol{w}_J^{a(new)} = \beta_a \left(\boldsymbol{A} \wedge \boldsymbol{w}_J^{a(old)} \right) + (1 - \beta_a)\, \boldsymbol{w}_J^{a(old)} \tag{4.5}$$

where β_a indicates the learning rate of the input module. The learning process in both the input and the output modules is the same. Therefore, all equations of the input module are also relevant to the output module by replacing superscript or subscript a with b.

4.2.3 k-Harmonic Mean (k-HM) Algorithm

The k-HM algorithm [24] is one of the clustering methods to find a set of cluster centroids in the data space. The main objective of the HM algorithm, like any other clustering algorithms, is to find a group of clusters that are homogeneous within the entities (data samples) in the same cluster and also that are highly separable from one cluster to another.

The k-HM algorithm is formulated as a local search optimization algorithm [24]. Its objective function is defined in Eq. 4.6.

$$f(k, p) = \min \sum_{i=1}^{N} \frac{k}{\sum_{j=1}^{k} \frac{1}{\|\boldsymbol{x}_i - \boldsymbol{c}_j\|^p}} \tag{4.6}$$

where k denotes the number of clusters; $\boldsymbol{x}_i$ denotes the i-th data sample; $\boldsymbol{c}$ denotes the cluster centroids. Initially, k-centroids are generated randomly. The k-HM algorithm assigns the data samples (entities) to the nearest cluster centroids and updates the centroids iteratively until a terminating criterion (the maximum number of iteration or a small tolerance threshold) is reached. The cluster centroid c_j is updated using the Eq. 4.7 [24].

$$\boldsymbol{c}_j^{(new)} = \frac{\sum_{i=1}^{N} m_{kHM}\left(\boldsymbol{c}_j/\boldsymbol{x}_i\right) \cdot w_{kHM}\left(\boldsymbol{x}_i\right) \cdot \boldsymbol{x}_i}{\sum_{i=1}^{N} m_{kHM}\left(\boldsymbol{c}_j/\boldsymbol{x}_i\right) \cdot w_{kHM}\left(\boldsymbol{x}_i\right)}, \forall j = 1, \ldots, k. \tag{4.7}$$

where the membership function $m_{kHM}(\cdot)$ and the weight function $w_{kHM}(\cdot)$ are defined in Eqs. 4.8 and 4.9.

$$m_{kHM}\left(\boldsymbol{c}_j/\boldsymbol{x}_i\right) = \frac{\left\|\boldsymbol{x}_i - \boldsymbol{c}_j\right\|^{-p-2}}{\sum_{j=1}^{k}\left\|\boldsymbol{x}_i - \boldsymbol{c}_j\right\|^{-p-2}}, \forall i = 1, \ldots, N,\ \forall j = 1, \ldots, k \tag{4.8}$$

$$w_{kHM}\left(\boldsymbol{x}_i\right) = \frac{\sum_{j=1}^{k}\left\|\boldsymbol{x}_i - \boldsymbol{c}_j\right\|^{-p-2}}{\left(\sum_{j=1}^{k}\left\|\boldsymbol{x}_i - \boldsymbol{c}_j\right\|^{-p}\right)^2}, \forall i = 1, \ldots, N \tag{4.9}$$

4.2.4 The Proposed RBFNDDA-FAMHM Model

The RBFNDDA-FAMHM model incorporates the salient features of FAM and k-HM to classify and adapts all RBFNDDA prototypes into a smaller group of information clusters. The training procedure of the proposed model in one epoch can be described as follows. The RBFNDDA algorithm is first applied to extract information from a training data set. Such information is a collection of prototypes $\boldsymbol{pr}_{rbf}$ for all P classes represented by centroids (or references), weights, and widths, respectively (see Eq. 4.10).

$$\boldsymbol{pr}_{rbf} = \left\langle \boldsymbol{r}_{1,\ldots,m_p}, Nw_{1,\ldots,m_p}, wd_{1,\ldots,m_p}\right\rangle, p = 1, 2, \ldots, P \tag{4.10}$$

Then, each reference vector of $\boldsymbol{pr}_{rbf}$ is complement-coded as $\boldsymbol{A}$, and the learning algorithm of FAM is initiated. Note that the FAM prototypes are in the form of hyperboxes whereas those of RBFNDDA are in the form of radial basis functions. In the proposed model, FAM is employed to assign the reference vector of an RBF prototype to one of its prototypes during supervised training. For this, the boundary of the selected winning FAM prototype is expanded in the data space to accommodate new information provided by an RBF prototype. Such boundary expansion of an FAM prototype allows inclusion of individual RBF prototypes of a class in the same data region, and this helps ensure homogeneity within an FAM prototype. Upon completion of an FAM training session, all individual RBF prototypes pertaining to each FAM prototype are grouped. Information of the first RBF prototype in terms of reference vector $\boldsymbol{c}_j$, weight Nc_j, and width wdc_j are initialized in an FAM prototype j. Then, the k-HM algorithm is executed to estimate the centroid of each FAM prototype. In this work, $k = 1$, and Eqs. (4.7)–(4.9) are applied in a single iteration of k-HM to update centroid $\boldsymbol{c}_j$ for each FAM prototype j based on a collection of RBF prototypes. The proposed RBFNDDA-FAMHM model does not undergo the local optimization process in multiple iterations to refine the FAM centroids (see Eq. 4.11).

$$\boldsymbol{pr}_{FAMHM} = \left\langle \boldsymbol{c}_{1,\ldots,m_p}, Nc_{1,\ldots,m_p}, wdc_{1,\ldots,m_p}\right\rangle, p = 1, 2, \ldots, P \tag{4.11}$$

Now, the RBFNDDA-FAMHM model incorporates the information obtained from $\boldsymbol{pr}_{FAMHM}$, with all previous information in $\boldsymbol{pr}_{rbf}$ removed.

4.3 The Experimental Study

The classification performance of the proposed RBFNDDA-FAMHM model was evaluated using three benchmark data sets from the UCI Machine Learning Repository [1]. In this chapter, the benchmark study covered comparisons between RBFNDDA-FAMHM and RBFNDDA as well as other state-of-art machine learning models.

4.3.1 Performance Comparison with RBFNDDA

Two benchmark data sets, Wisconsion Breast Cancer (WBC) and Heart, were utilized to compare the classification performance between RBFNDDA-FAMHM and RBFNDDA. The WBC data set contained 699 clinical records of patients who suffered from either malignant or benign breast cancer. On the other hand, the Heart data set contained 270 medical records, each indicated the presence or absence of heart disease.

The authors applied the same data partition scheme as employed in Paetz [15] to divide both WBC and Heart data sets equally into a training set and a test set. RBFNDDA-FAMHM underwent a single-epoch training session with the following parameter settings: $\theta^+ = 0.40$, $\theta^- = 0.20$, and $\bar{\rho}_a = 0.98$. The experiment was carried out for 30 runs, and the average classification results in terms of accuracy and number of nodes were estimated using the bootstrapped method [7].

Table 4.1 shows the classification results of RBFNDDA-FAMHM from this experiment and those of RBFNDDA reported in Paetz [15]. The test accuracy rates of RBFNDDA-FAMHM are slightly inferior to those of RBFNDDA. However, the significant advantage of RBFNDDA-FAMHM is its smaller network size as compared with that of RBNFDDA. It is worth-noting that RBFNDDA-FAMHM does not delete any information from the (redundant) RBF nodes from its structure. Instead, the hidden nodes are re-organized to produce a more compact network structure, without affecting classification accuracy seriously.

Table 4.1 Performance comparison between RBFNDDA-FAMHM and RBFNDDA (standard deviations are typed in round brackets)

	RBFNDDA-FAMHM		RBFNDDA [15]	
	Accuracy (%)	#Nodes	Accuracy (%)	#Nodes
WBC	96.52 (0.74)	55.0 (13.3)	96.86 (0.99)	70.6 (13.4)
Heart	77.87 (3.23)	66.6 (7.5)	79.26 (2.83)	83.6 (3.3)

4.3.2 Performance Comparison with Other Classifiers

The thyroid data set was utilized to benchmark the performance of RBFNDDA-FAMHM against those from other classifiers. The thyroid data set contained 215 patient records, each with five laboratory test results. Each record showed one of three diagnostic outcomes, namely, class-1 (normal function of the thyroid gland), class-2 (hyperthyroidism), or class-3 (hypothyroidism). Hyperthyroidism is referred to an over-active gland that produces an excessive amount of thyroid hormones; conversely, hypothyroidism is referred to an under-active gland that supplies an insufficient amount of thyroid hormones.

RBFNDDA-FAMHM was trained in a single epoch with the following parameter settings: $\theta^+ = 0.40$, $\theta^- = 0.20$, and $\bar{\rho}_a = 0.95$. In accordance with Kohavi [12], the 10-fold cross-validation method was used for performance evaluation. As such, the data set was divided (almost) equally into ten subsets; nine subsets were selected as the training set whereas the remaining was used as the test set. The classifier was trained and evaluated repeatedly for ten rounds with ten different combinations of the training and test sets, and the results were averaged. The same cross-validation method process was also applied to RBFNDDA with $\theta^+ = 0.40$ and $\theta^- = 0.20$.

Table 4.2 shows the results of RBFNDDA-FAMHM, RBFNDDA, and FAM. The results indicate the overall accuracy rate of RBFNDDA-FAMHM (i.e., 95.71 %) is higher than those of RBFNDDA (94.95 %) and FAM (95.24 %). Specifically, RBFNDDA-FAMHM is more accurate than RBFNDDA and FAM in classifying Class-2 and Class-3 samples. In other words, RBFNDDA-FAMHM can predict both abnormal conditions (hyperthyroidism and hypothyroidism) more accurately. Another finding is that RBFNDDA-FAMHM exhibits a smaller number of prototypes than that in RBFNDDA, but a higher number of prototypes than that in FAM. The results show that RBFNDDA-FAM is a more compact (as compared with RBFNDDA) and yet more accurate (as compared with both RBFNDDA and FAM) model for undertaking this thyroid diagnosis problem.

The performance of RBFNDDA-FAMHM is also compared with those from other classification methods. The results are listed in Table 4.3. The methods used for comparison include Expert System for Thyroid Disease Diagnosis with Neuro Fuzzy Classification (ESTDD with NEFCLASS-J) [11], the Artificial Immune Recognition System (AIRS) [16], the Artificial Immune Recognition System (AIRS) with fuzzy

Table 4.2 The 10-fold CV results using the thyroid data set

Network	Overall accuracy (%)	Accuracy (%)			#Nodes
		Class-1	Class-2	Class-3	
FAM	95.24	96.68	95.50	83.00	6.2
RBFNDDA	94.95	100.00	85.17	86.30	61.7
RBFNDDA-FAMHM	95.71	96.30	98.57	94.07	37.5

Table 4.3 Comparison of the 10-fold CV accuracy between RBFNDDA-FAMHM and other methods

Classification method	Accuracy (%)
ESTDD with NEFCLASS-J [11]	95.33
AIRS [16]	81.00
AIRS with fuzzy weighted preprocessing [16]	85.00
ADSTG [6]	93.77
RBFNDDA-FAMHM	95.71

weighted pre-processing [16], and the automatic diagnosis system based on thyroid gland (ADSTG) method [6] The outcome clearly indicates that the accuracy rate of RBFNDDA is the highest among a variety of classification methods.

4.4 Summary

In this chapter, an ANN classification model that comprises an integration of RBFNDDA, FAM, and k-HM has been introduced to handle data classification problems. The proposed RBFNDDA-FAMHM model mitigates the effects of the greedy insertion behavior of RBFNDDA, which tends to generate many prototypes during its training process. In this regards, the network compactness of RBFNDDA-FAMHM is achieved by re-organizing and adapting the RBF prototypes into a smaller number of clusters through FAM and k-HM classification processes. To evaluate the usefulness of RBFNDDA-FAMHM, three benchmark data sets have been used. Its classification performances have been compared with those of RBFNDDA and other machine learning models. The experimental results positively indicate that RBFNDDA-FAMHM, which is a more compact model than RBFNDDA, is able to produce high accuracy rates in undertaking data classification problems.

For further work, additional experiments will be carried out to investigate the effectiveness of RBFNDDA-FAMHM in dealing with data classification problems in different domains. In addition, methods to further enhance the classification performance of RBFNDDA-FAMHM will be studied.

References

1. Bache, K., Lichman, M.: UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. University of California, School of Information and Computer Science, Irvine (2013)
2. Berthold, M.R., Diamond, J.: Constructive training of probabilistic neural networks. Neurocomputing **19**(1–3), 167–183 (1998)
3. Bortman, M., Aladjem, M.: A growing and pruning method for radial basis function networks. IEEE Trans. Neural Netw. **20**(6), 1039–1045 (2009)

4. Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J.H., Rosen, D.B.: Fuzzy artmap: a neural network architecture for incremental supervised learning of analog multidimensional maps. IEEE Trans. Neural Netw. **3**(5), 698–713 (1992)
5. Carpenter, G.A., Milenova, B.L., Noeske, B.W.: Distributed artmap: a neural network for fast distributed supervised learning. Neural Netw. **11**(5), 793–813 (1998)
6. Dogantekin, E., Dogantekin, A., Avci, D.: An automatic diagnosis system based on thyroid gland: ADSTG. Expert Syst. Appl. **37**(9), 6368–6372 (2010)
7. Efron, B.: Bootstrap methods: another look at the jackknife. Ann. Stat. **7**(1), 1–26 (1979)
8. Fernández-Navarro, F., Hervás-Martínez, C., Gutiérrez, P.A., Peña-Barragán, J.M., López-Granados, F.: Parameter estimation of q-gaussian radial basis functions neural networks with a hybrid algorithm for binary classification. Neurocomputing **75**(1), 123–134 (2012)
9. Han, H.-G., Qiao, J.-F.: A structure optimisation algorithm for feedforward neural network construction. Neurocomputing **99**, 347–357 (2013)
10. Hudak, M.H.: RCE classifiers: theory and practice. Cybern. Syst. **23**(5), 483–515 (1992)
11. Keleş, A., Keleş, A.: Expert system for thyroid disease diagnosis. Expert Syst. Appl. **34**(1), 242–246 (2008)
12. Kohavi, R.: A study of cross validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the 14th International Joint Conference Artificial Intelligence (IJCAI), pp. 1137–1145. Morgan Kaufmann (1995)
13. Martínez-Rego, D., Pérez-Sánchez, B., Fontenla-Romero, O., Alonso-Betanzos, A.: A robust incremental learning method for non-stationary environments. Neurocomputing **74**(11), 1800–1808 (2011)
14. Ni, J., Song, Q.: Dynamic pruning algorithm for multilayer perceptron based neural control systems. Neurocomputing **69**(16–18), 2097–2111 (2006)
15. Paetz, J.: Reducing the number of neurons in radial basis function networks with dynamic decay adjustment. Neurocomputing **62**, 79–91 (2004)
16. Polat, K., Gunes, S.: An expert system approach based on principal component analysis and adaptive neuro-fuzzy inference system to diagnosis of diabetes disease. Digit. Signal Proc. **17**(4), 702–710 (2007)
17. Reed, R.: Pruning algorithms—a survey. IEEE Trans. Neural Netw. **4**(5), 740–747 (1993)
18. Reilly, D.L., Cooper, L.N., Elbaum, C.: A neural model for category learning. Biol. Cybern. **45**(1), 35–41 (1982)
19. Specht, D.F.: Probabilistic neural networks. Neural Netw. **3**(1), 109–118 (1990)
20. Widrow, B., Greenblatt, A., Kim, Y., Park, D.: The no-prop algorithm: a new learning algorithm for multilayer neural networks. Neural Netw. **37**, 182–188 (2013)
21. Xu, R., Wunsch II, D.C.: BARTMAP: a viable structure for biclustering. Neural Netw. **24**(7), 709–716 (2011)
22. Yang, S.-H., Chen, Y.-P.: An evolutionary constructive and pruning algorithm for artificial neural networks and its prediction applications. Neurocomputing **86**, 140–149 (2012)
23. Yang, L., Yang, S., Zhang, R., Jin, H.: Sparse least square support vector machine via coupled compressive pruning. Neurocomputing **131**, 77–86 (2014)
24. Zhang, B.: Generalized k-harmonic mean—boosting in unsupervised learning. Technical Report HPL-2000-137, Hewlett-Packard Laboratories (2000)

Chapter 5
Personalized Intelligent Mobility Platform: An Enrichment Approach Using Social Media

Ruben Costa, Paulo Figueiras, Carlos Gutierrez and Luka Bradesko

Abstract This chapter aims to present a technical approach for developing a personalized mobility knowledge base supported by mechanisms for extracting and processing tweets related with traffic events, in order to support highly specific assistance and recommendations to urban commuters. In order to address a personalized mobility knowledge base, a step-wise approach is presented with the purpose of construction and enriching a knowledge model from heterogeneous data sources providing real-time information via Personal Digital Assistants (PDAs). The approach presented is decomposed into several steps, starting from data collection and knowledge base formalization targeting the development of a personalized intelligent route planner, enabling a more efficient decision support to urban commuters. The work presented here, is still part of ongoing work currently addressed under the EU FP7 MobiS project. Results achieved so far do not address the final conclusions of the project but form the basis for the formalization of the domain knowledge do be acquired.

Keywords Intelligent transportation systems · Knowledge acquisition · Social media

---

R. Costa (✉) · P. Figueiras · C. Gutierrez
UNINOVA, Centre of Technology and Systems, Monte de Caparica,
2829-516 Caparica, Portugal
e-mail: rddc@uninova.pt

P. Figueiras
e-mail: paf@uninova.pt

C. Gutierrez
e-mail: csg@uninova.pt

L. Bradesko
Jozef Stefan Institute, Jubljana, Slovenia
e-mail: luka.bradesko@ijs.si

J.W. Tweedale et al. (eds.), *Intelligent Decision Technology Support in Practice*, Smart Innovation, Systems and Technologies 42,
DOI 10.1007/978-3-319-21209-8_5

5.1 Introduction

Intelligent Transport Systems were envisaged to bring about optimization of travel through synergies of various technologies in particular information and communication technologies.[1] The main philosophy has been employing a wide range of devices which were to provide system knowledge of the status and whereabouts of components in a transportation system in order to control the various parts including vehicles, people. Towards an optimum versus a limited set of criteria, interurban traffic management systems. Existing intelligent transport systems however are not able to handle multiple complex situations in which mass of various agents (subjects, objects) and influences (information from the environment, society, time and place contexts) are causally interlinked. Moreover, existing systems usually optimise the shortest route, shortest time criteria function not taking into consideration predicted complex events and the minimisation of the energy consumption and pollution.

Nowadays with the proliferation of a panoply of smartphones and tablets on the market, almost everyone has access to mobile devices, which offers better processing capabilities and access to new information and services. Taking this into account the demand for new and more personalized services which aid users in their daily tasks is increasing. Current navigation systems lack in several ways in order to satisfy such demand, namely, accurate information about urban traffic in real-time, possibility to personalize the information used by such systems and also the need to have a more dynamic and user friendly interaction between device and traveller. In short, the current navigation systems were not designed to be intelligent enough to communicate their needs effectively to humans, and they are not intelligent enough to understand humans attempts to satisfy those needs. The authors recognize that the proliferation of social media, access to transport and traffic system information via mobile broadband services and availability of information of smartphones as sensors have changed the landscape in terms of the ability to deploy cooperative systems. Google was early to catch on this trend with their traffic information system which built on principles of crowd-sourcing [1] enabling participant in the system to contribute and benefit from shared information. However according to the authors knowledge, none of the existing systems uses the information stored in the existing social networks for the purpose of deciding the best possible route (real-time detection of traffic accidents from social networks).

In the scope of this work, the focus is on the analysis and detection of traffic events, where information being disseminated by social networks can have an important role. Social networks allow the detection of very small events, such as a damaged car in a side street. In addition to that, the interval between the beginning of a traffic incident and the publication of a tweet about this incident usually tends to be much less, when compared to the time required for a news agency to broadcast about such particular event.

Extracting useful information from tweets presents some challenges, as described by the authors (Bifet and Frank and Gimpel et al. [2, 3]). Taking into account that

[1] See http://sites.ieee.org/itss/.

the information is completely unstructured, tweets can contain grammatical errors and abbreviations, each user has its own way of writing, the information can be incomplete (For example, a street name with no more information about city), false or not credible. From the authors point of view, the information contained in a tweet message, can only be considered relevant, if the same topic was referred by users in a short period of time. Moreover, automatically recognize tweets about traffic events can be a difficult task (the tweet Excessive speed is the main cause of car accidents in Liverpool is not relevant but contains car accident and Liverpool). In fact, one of the most important limitations in extracting useful information from a tweet is its 140-character limit.

The multi-variant criteria and inherent distributed nature of a solution in a collaborative system based on crowd-sourcing, in which local optimization gains must be balanced against the cost of sharing global knowledge via mobile systems, mandate research towards a distributed communication solution for more personalized and intelligent mobility solutions. MobiS project [4], is approaching this complexity in a holistic way by introducing the concept of a mutual symbiotic relation in its core. This is a single, omnipresent relation type where all components that are involved or influenced by the multimodal mobility complex environment like objects (machines, computers, transport devices …), people (citizens, social networks …), businesses and organisations, legal environment (formal rules, standards, regulations, …) and natural and social environment that benefit from interacting with each other towards efficient and energy-positive mobility.

This chapter presents an approach for developing a personalized mobility knowledge base to be used by a mobile travelling companion application, whose conversational interactions crowdsource information about the world for the purpose of providing highly specific assistance and recommendations to the user. Into this extend, the notion of real-time information on social networks is also taken into account within this approach. The objective is to enrich a personalized mobility plans from urban commuters with traffic events captured from social networks, as a way to provide more accurate information to travellers and support the decision-making in their daily mobility plans. As social network, this work is focusing in Twitter, a popular microblogging service which has received much attention recently. It only cost 140 characters to generate a message (called tweet) for any user, and the real time nature of the tweets may be used to detect an event quickly. This work adopts an event-driven approach, based on Twitter messages ("tweets"), with the objective of detecting and classifying traffic events in real time. The approach presented here, implements a computational framework able to learn to classify a set of traffic-related tweets adopting machine learning techniques, extract a set of contextual information such as: the location and type of event, geo locate the event on a map and the follow up of the incident i.e. monitoring incidents evolution.

The work presented here, is still part of ongoing work currently addressed under the EU FP7 MobiS project. Results achieved so far do not address the final conclusions of the project but form the basis for the formalization of the MobiS domain knowledge along with the MobiS data models.

This chapter is structured as follows: Sect. 5.2 describes the overall approach, where data is acquired from heterogeneous sources and reasoning is applied to each individual mobility plan. Section 5.3 described the related work, which is considered relevant for support the development of the work presented here. Section 5.4 presents the methodological approach for achieving personalized mobility. Section 5.5 presents the methodology for domain knowledge formalization and Sect. 5.6 defines the vocabulary used by the knowledge base. Section 5.7 presents the approach developed for extracting traffic events from tweeter. Section 5.8 presents the mobile traveling companion module and Sect. 5.9 presents the conceptual architecture, followed by the conclusion and future effort.

5.2 Conceptual Approach

Following the above hypothesis, the main goal of this work, is to create a new customized and intelligent mobility concept and solution by applying artificial intelligence concepts and methods that will monitor, model and manage the complex network of urban mobility subjects, objects which interacts with natural, social and business environment in a symbiotic way, where all different stake-holders in urban traffic will be able to cooperate, thus providing more efficient citizen mobility.

More particularly, the presented work aims to bring a substantial contribution for the development of a holistic infrastructure for collecting, analysing and processing information from different sources (to be addressed within MobiS project), being able to:

- Retrieve and analyse the mobility plan of each citizen, coming from different information sources (desktop calendar, Personal Digital Assistant (PDA), sensor tables).
- Allow the citizen to define his/her decision criteria in order to propose the best alternative route. Decision criteria could be cost, time, energy savings.
- Collect information about the possible transport means that could satisfy each citizens mobility plan. Here the authors include traditional (public transport, private transport, urban bicycle networks) and new and emerging mobility models (car sharing, tweeting vehicles).
- Reasoning, predicting and planning the traffic in urban areas based on the citizens mobility plans, existing transport network topology, decision criteria and other information sources coming from different public or private service providers (For example, Metro, tram, bus, schedules, parking areas), existing sensor network infrastructures (For example, sensor-based parking services, positioning of existing transport means) and social media. The idea is to develop a prediction model, which will be able to predict potential anomalies (traffic jams, pollution, saved energy) based on the models learned from the historical data, current information data and other individual predicted models.

- Continuously update the behaviour and simulation model and propose alternative plans to citizens considering possible changes, anomalies and trends. In order to realize this vision, the work which in this chapter is concerned, will create the necessary research and prepare the basis towards handling with different sources of information and services.
- Private-public services: Those concern traffic information systems, telematics services, time schedules, e-transactions (e-booking, e-payment). Those services are provides either by private or public bodies (urban authorities).
- Ambient data: Those are coming from various sensors and related infrastructures. They might also consist of a specific service (telematics) or work autonomously in other cases (sensors on parking slots providing real-time availability).
- Networking data, consisting of information coming from existing social media. This information could help in proposing car sharing alternatives with people knowing each other from existing social networks (Facebook, Twitter). Networking data can also have the form of existing applications supporting clustering, communities, crowdsourcing.
- Reasoning, prediction and Artificial Intelligence (AI) services. Existing infrastructures and project results will be used in order to provide reasoning, prediction and decision function services. The adoption of a federation approach will enable to collect the information from different providers and process them through those AI services, in order to serve peoples mobility needs in the most efficient and personalized way.

The conceptual approach driving this work (Fig. 5.1), adopts the Monitor, Analyse, Plan, Execute and Knowledge (MAPE-K) adaptive control loop in order to address symbiotic relations to be tackled by the project. MAPE-K was first introduced by IBM in their white paper “An architectural blueprint for autonomic computing” [5] and later introduced in the context of self-adaptive systems. MAPE-K is a construct for realising a computing environment with the ability to manage itself and dynamically adapt to change in accordance with given policies and objectives.

The first step in Monitoring is to detect any unexpected events of interest by very efficient processing of events coming from all possibly relevant sources (private-public services, ambient/mobility data, social networking data). Monitor collects the details from the managed resources, aggregates, correlates and filters these details until it determines a symptom that needs to be analysed. The challenge is to enable the identification of early warnings, especially in the case of a very changeable environment (For instance, where the patterns indicative of a type of relevant situation have been dynamically changing).

The next step is the Analysis of these early warnings using reasoning, prediction on the symptoms provided by the monitor function. This process introduces background information in order to support a correct (semantic) interpretation of discovered unexpected event. The outcome of integrating data coming from different sources into a detection pattern, and associating it with a semantic description, is a meaningfully identified situation of interest, to which real-time adaptive dynamic route planning should react upon. The challenge is to perform the necessary reasoning (simulation

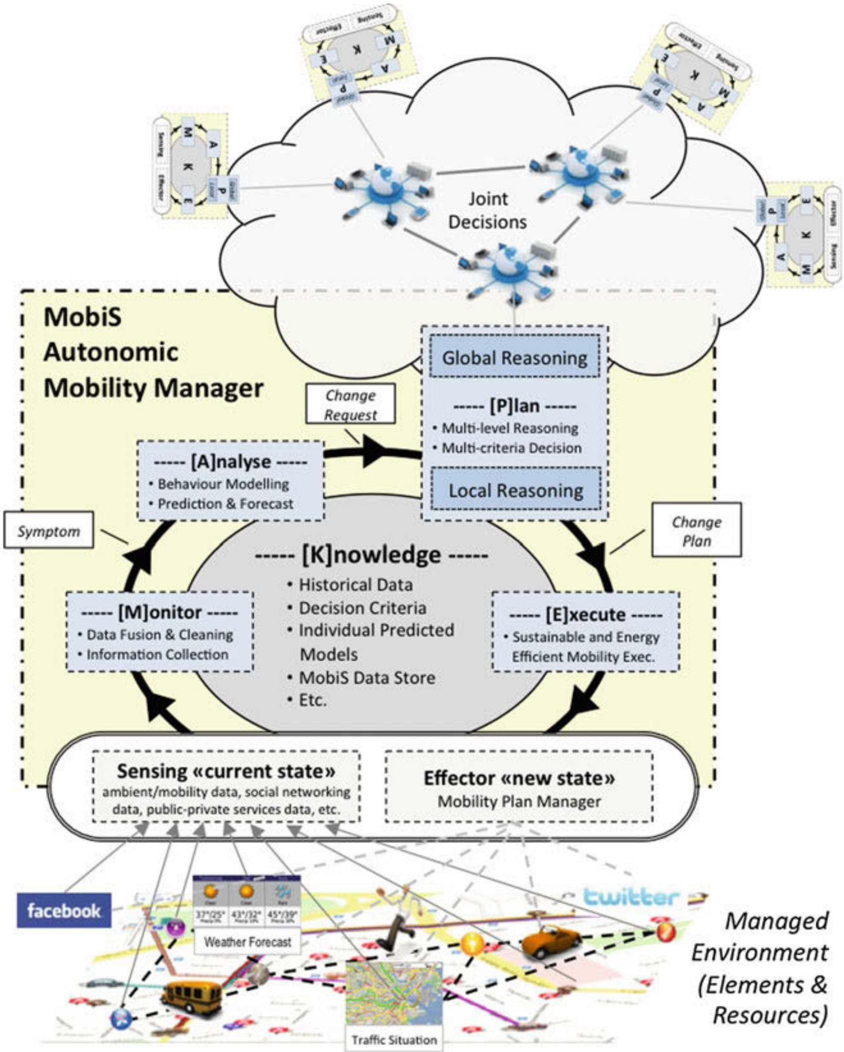


Fig. 5.1 Concept approach

and prediction) procedures on the stream data in real-time. If changes are required, a change request is logically passed to the plan function.

The Plan stage generates alternative plans/solutions (route (re)planning) for untapped situations. It structures the actions needed to achieve goals and objectives. The plan function creates or selects a procedure to enact a desired alteration in the managed resource (mobility plan manager). The challenge is to support a real-time feedback to the proposed reactions. Plan functions are performed at different levels: locally, by reasoning about optimisation at local context, and globally reasoning using large-scale knowledge and processing.

The Execute step, acts upon the managed resource (mobility plan manager) using effectors, based on the actions recommended by the plan function. The Knowledge aggregates data shared among the monitor, analyse, plan and execute functions. The shared knowledge includes historical data, decision criteria, individual predicted models. The execute step is responsible for updating the knowledge based, whenever a new individual mobility plan is executed.

5.2.1 Technological Objectives

The work proposed here provides some specific capabilities that derive from the three basic areas (federated architecture, multi-criteria decision function and reasoning/planning/prediction) and fulfil the expected functionality. Those capabilities are presented in Table 5.1.

5.3 Related Work

From a more general point of view, MobiS project encompasses several branches from different areas within Intelligent Transportation Systems (ITS) and Knowledge Acquisition (KA). In fact, the project aims to be an agglomerate enabler in terms of ITS technologies, integrating several research topics into one single system. The objective is to cover several technology branches which present great research efforts, as is the case of Traffic Prediction and Simulation [7, 8], Smart Parking [9], User Context based Navigation Systems [10]. The objective is simple: to improve traffic and road conditions and flow, while lowering carbon emissions and pollution within the road network. Some research regarding these topics is already taking place, as evidenced in Chueh et al. [11], and ITS generic platforms are already being developed [12]. Moreover, the above mentioned research endeavors use different types of data sources, from sensor data to floating car data, social networks, and infrastructure gathered data.

With respect to the work proposed here, some promising research has been developed in the area of Companion Cognitive Systems, according to Forbus and Hinrichs [13], companions will help their users work through complex arguments, automatically retrieving relevant precedents, providing cautions and counter indications as well as supporting evidence. Companions will be capable of effective operation for weeks and months at a time, assimilating new information, generating and maintaining scenarios and predictions. Companions will continually adapt and learn, about the domains they are working in, their users, and themselves. This work aims to bring this vision, into personalized urban mobility, where companions learn from urban commuters about their daily habits and make recommendations in order to make individual mobility plans more efficient and more personalized.

Table 5.1 Technological objectives

Technological Objectives	Description
Services federation/networking among objects, people, devices and organizations	Improving networking communication capabilities allowing collaborative processes among objects, people and other actors (authorities, service providers). This will be through open and federated architectures and generic interfaces that will be adapted by the Service providers in order to publish their services and make them identified, composed and consumed by the interested parties
Real-time situational awareness (planning, optimisation, prediction)	Implementation of technologies for real-time large scale data modelling, prediction and optimisation. Based on the multiple data and information sources coming from the mobility network, travellers intention, external monitoring networks and open access information sources, machine learning mechanisms will model the behaviour of entities involved, the complete mobility space and predict, detect and describe complex events that will influence the planning route functionality
Decision criteria function	New Decision Criteria Function as the basis for all decision, calculations and reasoning. Addressing environmental issues in a broader sense pursuing not only the pure reduction energy consumption and of CO_2 emissions, but also other negative influences n natural environment and society
Local reasoning capabilities	Extends the capabilities towards learning, awareness and reasoning on different levels
Large scale reasoning	Reasoning on a large scale knowledge through LarKC platform (Open source large scale reasoning platform developed under the FP7 IP LarKC [6], and data from urban transport systems, external data and news services (traffic, weather, pollution, news) and formalised knowledge descriptions (based on Cyc[a]—largest known common sense KB developed. For instance, the Euridice Knowledge Model—FP7 IP (EURIDICE) used in the mobile data transmission domain[b]

[a] See http://www.cyc.com/documentation
[b] See http://www.euridice-project.eu

In order to make personal companions, and into this respect, personal travel companions, more user friendly, conversational interaction between human and system is key. Real-world conversations between humans can contain context dependent terms or phrasing, rely on conversational memory, require common sense knowledge about the world, events, or facts, retain memory stretching back over a long history of interactions and shared experiences, and infer meaning from incomplete and partial statements. Perhaps the most successful system to date using these methods is Apple's Siri system, and it can only handle a limited number of predicted situations. One of the weaknesses of such systems is the lack of a discourse model that can support clarification and correction dialogues in any general way. Lasecki [14], Witbrock [15] and Kittur et al. [16] have all proposed important developments in conversational assistants capable of holding natural language dialog with users and knowledge acquisition from crowdsourcing. In this work, the authors propose a knowledge formalization methodology for supporting conversational interaction using natural language with urban commuters with the objective of providing them better recommendations to their individual mobility plans.

Recommendations systems to support driving behaviours have been proposed by Chandrasiri et al. [17] and Wu et al. [18]. The objective is to propose a driving the monitoring system to provide drivers an indicator of the danger level for driving safety. The authors advocate that the analysis of driving skill/driver state can be used in building driver support and infotainment systems that can be adapted to individual needs of a driver. Although, such work make the use of machine learning algorithms to capture driving behaviours, this research attempts to have a more comprehensive overview on capturing and learning mobility patterns of urban citizens, including car drivers and public transportation users.

Regarding the influence of social network data for enriching personal mobility plans, it can be considered that, event detection on social networks as a subdomain of web mining, so the publications on this subject are of various natures. Nevertheless, it will be described here some recently relevant research works, which have been done in the area of detecting events (not only about traffic) through twitter.

Concerning traffic events, Wanichayapong et al. [19], Schulz et al. [20], and Twitraffic[2] present different approaches to detect events from tweeter. Wanichayapong et al. [19] use an approach to extract parts of tweet messages as verbs and prepositions in order to parse tweets. Such approach enables to separate the tweets that refer to a traffic incident localized to a certain point and those located between two points. Unlike the authors system, tweets must have a certain structure for further processing. Schulz et al. [20] use several types of features are used to train a machine learning classifier: character n-grams, syntactic features, spatial and temporal unigram features, term frequency—inverse document frequency (tf-idf) [21] and linked open data features [22]. The classifier in this system uses only the tf-idf with very promising results, similar to those in Wanichayapong et al. [19] and Schulz et al. [20]. Twitraffic is a system to filter tweets about traffic in UK and extract the location as well as the sentiment of the tweet. None of them contain a component for monitoring incidents evolution.

Concerning event detection and management more generally, bibliography is more numerous. Abel et al. [23] propose an approach to detect any kind of incident from

[2] See https://twitter.com/twitraffic.

an emergency broadcaster. The main difference with this approach, except that it is not developed for traffic events, is that the data stem from a database of incidents and not from social networks. This has the advantage of easily structure the incidents because they all have the same format, which is not the case of incidents contained in the tweets. Social media serve only to enrich the information on events, while this research focuses directly on tweets. The authors use a Jaccard distance to compare tweets. A system for managing crisis through tweet analysis is presented in Rogstadius et al. [24]. Tweets analysed were previously filtered by the user of the system. It does not include algorithms to classify the content of a single tweet. A clustering algorithm for tweets is explained. Tedas [25] uses a system of propagation of rules for obtaining new tweets. In Twical [26] each event has four characteristics: event phrase, date, event type and named entity. TweetTracker [27] use keywords and geolocation as a system input. Other systems like Ushahidi [28] specialized in the analysis of mobile data and not just social networks. It is completely based on crowdsourcing. Yin et al. [29] present a crisis management system that incorporates tweet clustering and burst detection algorithms [30].

To the authors best knowledge, there are no publications on a system that integrates all stages of this system, namely classification, detection of traffic event type, information extraction (location entities, temporal information), geolocation and monitoring of traffic events.

5.4 Methodological Approach

The methodological approach to be tackled for reaching a personalized knowledge bases is presented here and is decomposed into several steps, starting from data collection and knowledge base formalization into an intelligent route planner supported by reasoning and prediction mechanisms. Figure 5.2 depicts the multilayer approach to be followed. This layered approach uses Graph Theory to represent the road and transport network.

5.4.1 Base Graph

A graph representing the entire transport network is built using Open-Street Maps (OSM) data.[3] OSM use a collaborative project to create a free editable map of the world. The two major motivations behind OSM are restrictions on use or availability of map information across the world and the advent of inexpensive portable satellite navigation devices.

[3]See http://wiki.openstreetmap.org/.

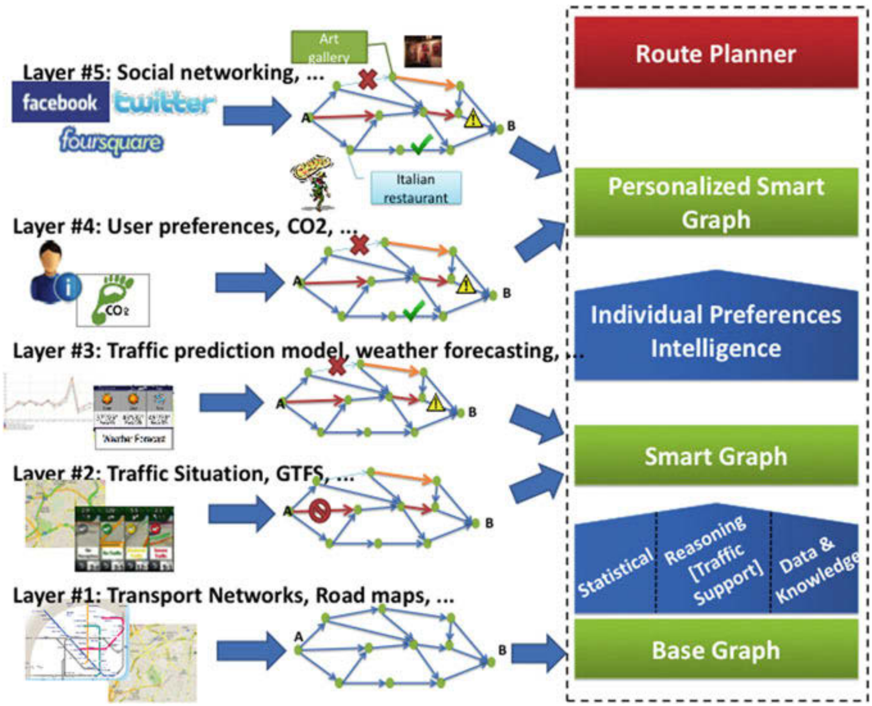


Fig. 5.2 Route planner methodological approach

OSM data is extracted from GeoFabrik,[4] a specialized website for downloading available OSM data, in the form of a graph. The graph contains information about roads, railways, footways, bicycle paths, along with speed limits, network barriers, public transportation stops and stations, and other data regarding the transport network. The OSM data is divided into nodes, ways and relations. This OSM data is then fed into a specialized OSM router, the Routino router.[5] Routino offers a series of routing functions and it easily integrated with other types of functionalities, such as the multimodal routing functions for public transport.

The Routino routing algorithm has suffered several improvements in terms of performance since its beginnings. The algorithm has its roots in the Dijkstra and A* algorithm families, adding constraints regarding particular transportation means, and the notion of super nodes, which are the most interesting nodes in the route. A node is deemed to be interesting if it is the junction of three or more segments or the junction of two segments with different properties or has a routing restriction different from the connecting segments. Instead of finding a long route among a data set of 8,000,000 nodes (number of highway nodes in UK at beginning of 2010) it

[4]See http://www.geofabrik.de/.

[5]See http://www.routino.org/.

finds one long route in a data set of 1,000,000 nodes and a few hundred very short routes in the full data set. Since the time taken to find a route is proportional to the number of nodes that need to be considered the main route takes 1/10th of the time and the very short routes take almost no time at all.

The next layer deals with Data Capture and Knowledge Capture: data and knowledge are captured from different sources such as social networks (Twitter posts about traffic), Geographic Information Systems (GIS) data, map and transport networks data other than OSM, sites and Application Programming Interfaces (APIs) about places and venues (For example, Google Places, Foursquare, Facebook Events), Googles Generic Transit Feed Specifications (GTFS).[6] Although Data Capture and Knowledge Capture are on the same level, they are different in the sense that data is thought of as being composed by unstructured bits and pieces of information, whereas knowledge has already some type of information structure. The next step is responsible for feeding the captured data and knowledge into a registry which has records for all events which matter to the creation of a mobility plan, from real-time traffic, accidents, road blocks or other traffic events, to particular social events such as sport or cultural events, which may have some influence in every individual mobility plan of urban commuters.

At this stage, captured data and knowledge is annotated using Cyc concepts (described in detail in further sections) in order to use it later in the reasoning process. The annotation process enables Cyc to reason about certain traffic aspects, such as users preferred places and paths, journey times and journey updates, optimized emissions and consumptions, among other. Some of this data will be used later by the reasoning and inference modules, but at this stage, this layer will use only data related with current traffic situation and events, public transportation paths and schedules (GTFS), car sharing services and data, and other data which does not need to be inferred or reasoned upon. This data may come from various different sources, such as social networks (feeds about traffic events), news, specialized traffic services and operators or simply from sensors scattered throughout the transport network. The combination of these information sources creates a graph with dynamic, built-in information about public transportation, car sharing, bike rental and the current status of traffic. Data regarding the road and transport network is aggregated so as to create a Traffic Model corresponding to the current transport network situation.

5.4.2 Smart Graph

Real-time traffic data is collected and stored in order to build a historical background about traffic. Historical data is mainly coming from sensor data collected throughout several years. The usage of historical data will enable the development of traffic forecasts, which depend on the hour of the day and the day of the week, traffic

[6] See https://developers.google.com/transit/gtfs/.

and weather patterns, regular and complex events and other information that can somehow aid in the creation of predictive traffic models.

This information is used to create Traffic Nowcasts, which are simulations based on the current traffic conditions. Depending on the type of events, it is expected to simulate how a real-time event can affect traffic conditions for a certain period of time. One example may be a football match. If a football match is planned for a certain period in city centre, there is a good chance that all roads near the stadium will be congested before the match. Hence, the use of simulation capabilities to predict the congestion around the stadium for a particular time.

Another use of the information is to create traffic forecasts or predictions based on the historical information about road network, traffic events and other situations that affect the road network in the near future. In most cases such situations occur periodically in constant intervals of time (For example, traffic congestions on normal days, sports matches). The idea is to combine information gathered from these two modules with relevant information coming from social networks, news or any other information source about future events, in order to simulate and predict the traffic conditions. By prediction and simulating, such results will enable reasoning engine based on Cyc, to better understand the nature of traffic and to suggest alternative routes to users, according to their preferences and needs.

At this stage, the graph contains real-time traffic data enriched with predictive and simulation capabilities from the traffic forecast and nowcast modules. This enable personal mobility services not only to create better mobility plans that are based on the current traffic information but also instantly recalculate such mobility plans.

5.4.3 Personalized Smart Graph

The next iteration is based on the addition of individual user preferences to the graph, such as the type of vehicle (car, bicycle, public transportation), the type of modality (shortest, quickest, environmentally friendly) or the type of route (from A to B, cultural path, time filling, personal). The information needed can be accessed directly through user preferences, from social networks or from historical records of past individual routes.

This layer already works with the knowledge base to find patterns on user routes and preferences, in order to find the best possible route for a particular user. The inference engine based on Cyc, will learn users preferences and habits, while providing suggestions on where to go, how and when to get there and advising about how to save fuel, contributing to lowering emissions.

For instance, one can imagine that a certain user likes Italian food and modern art. The engine can infer this fact from the data that it acquired through the user interaction i.e. asking mobile travelling companion about routes with these two types of places, namely Italian restaurants and modern art museums, on previous mobility plan requests.

Hence, the mobile travelling companion will search for possible places and routes which match the users preferences on its knowledge base, and will construct a new route and the travel plan that can even include these preferences. Places, venues and events are gathered from social networks and other online information sources, such as event sites, Google Places, Google Calendar,[7] TripIt,[8] Foursquare,[9] Facebook,[10] among others. Also, depending on the user, interaction with social networks is supported, meaning that the mobile travelling companion is capable of not only gather information from social networks, but also post relevant information on them, such as posting route information on the users Facebook page, or inviting other users to share a users car in some trip.

The preliminary tests regarding smart graph creation and update, where conducted on a dual core processor using 4 GB of RAM, running a Ubuntu virtual machine. For the test, it was used graph containing the traffic network for Sweden region (282 MB file) and it took 30 s to be processed. Although the figures dont really express the performance of the algorithm used, future test will be run on a server machine with more processing capabilities.

5.5 Methodology for Domain Knowledge Formalization

The way in which the personalized mobility concept is addressed is twofold: on one hand it focuses on the knowledge formalization that is necessary to support knowledge driven (top down) methods; on the other hand, it deals with data driven modelling, deep analytics, predictions, trend detection and knowledge extraction (bottom up) approaches. Since the authors are aiming to combine methods from both groups, there are strong dependencies and common grounds between these. The formalized knowledge model that will support the travelling companion must take into account two main features:

- Context calculation, i.e., identification of entities determining traffic current situation, by allowing services to access the related information sources. For example, by determining the traffic flow for the next hour in a certain area, the knowledge based allows to retrieve a wealth of formalized contextual information (For example, current traffic and weather information).
- Reasoning on live data, enriched with knowledge already available in Cyc[11] and with knowledge automatically detected by the proposed infrastructure. For example, patterns in traffic situation during a weekly day can be captured by observing

[7] See https://developers.google.com/.

[8] See https://www.tripit.com/developer.

[9] See https://developer.foursquare.com/docs/.

[10] See https://developers.facebook.com/docs/reference/api/.

[11] See http://www.cyc.com/documentation.

traffic flows over a period of time, formalized as rules in the knowledge base, and reused for the purposes of more efficient mobility.

There are several methodologies and tools for enabling domain knowledge formalization, namely OWL, PIDF or Cyc. The reason behind choosing Cyc as the most suitable context modelling ontology, is that fact that Cyc provides a unique and extensive commonsense knowledge base and inference capabilities therefore the key to a number of different methods for improving and enhancing the results of existing information extraction systems. In particular, Cyc can support:

- Strengthening of type identification,
- Detection of certain information extraction errors,
- Resolution of references to specific individuals; and
- Unification of references to the same entity.

What is more, Cyc as an integrated knowledge based system supports association of ontological concepts with real data through the Cyc API. This enables reasoning with live data and at the same time obtaining simple process of modification of the live knowledge model, with modelling time error detection.

The knowledge encoded in Cyc is very broad and already supports many of the concepts needed in the scope of this work. Due to a compact integration of different Cyc components the whole system presents a reliable, robust and compact tool for advanced real time reasoning with the mobility data. What is more, Cyc supports authentication of human modellers and knowledge base versioning. From this perspective, one of the disadvantages of the Cyc system is the lack of support for the contextualized modelling tool. The main aspects of methodology for domain knowledge formalization in Cyc are represented in Fig. 5.3.

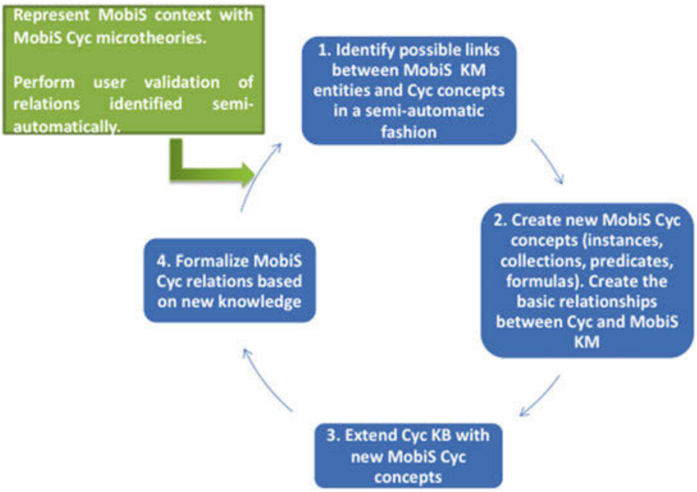


Fig. 5.3 Knowledge formalization methodology

Due to the fact that Cyc contains a broad commonsense knowledge base, allows reasoning and integration of live data, provides on-the-fly error detection, Cyc has been chosen as a principal tool for domain knowledge formalization. The methodology for domain knowledge formalization in Cyc uses the notion of Cyc microtheories, individuals, collections and predicates. Microtheories are used to represent thematic subsets or context of the ontology. Cyc collections are kinds of classes which instances have common attributes. Individuals in Cyc KB (knowledge base) represent single objects and predicates refer to relations.

5.6 Knowledge Base Vocabulary Definition

The Cyc knowledge base is a common sense ontology that will appropriately represent the vocabulary being reviewed in the mobile domain. Because it is common sense, it not only covers the concepts connected to the traffic, but human activities as well. For example, it has the concept of what is the driver (#$Driver-TransportationProfessional) and to humans obvious knowledge that vehicle needs a driver, which is not obvious for machine reasoning systems. The vocabulary part of the ontology is open source and can be downloaded as an RDF graph from the OpenCycs Sourceforge site.

The KB extension will also rely on the preexisting work already performed under the EURIDICE context model implementation of transportation concepts in Cyc. In EURIDICE, the context model is consisting of three different areas related to the main logistics stakeholders and they are called: Cargo Domain, Transport Domain and Environmental Domain. The various transportation modes within the EURIDICE taxonomy are grouped as follows:

1. Road Transport;
2. Air Transport;
3. Rail Transport; and
4. Maritime Transport.

As stated before, the starting point of this KB extensions is to reuse the existing Cycs Microtheory called TransportationMt. This is the Microtheory which subsumes all of the vocabulary regarding the transportation and traffic. Before the authors began working on the extensions the model contained 2179 assertions and 17 first level sub-Microtheories (Fig. 5.4) which encoded knowledge about transportation.

When using reasoning and not just the search in the KB, the number of Microtheories that contain the transportation relevant knowledge (because of the transitivity) goes up to 2000. Other Microthories inside the KB will be used to gather users information, regarding their choices and preferences as transportation network users and as individuals, in order to perform personalized reasoning. In terms of privacy, personal user data can only be access by each individual user itself and not available to others. This privacy method is based on the hierarchical connection between Microtheories. Microtheories can be hierarchically connected, where more specific

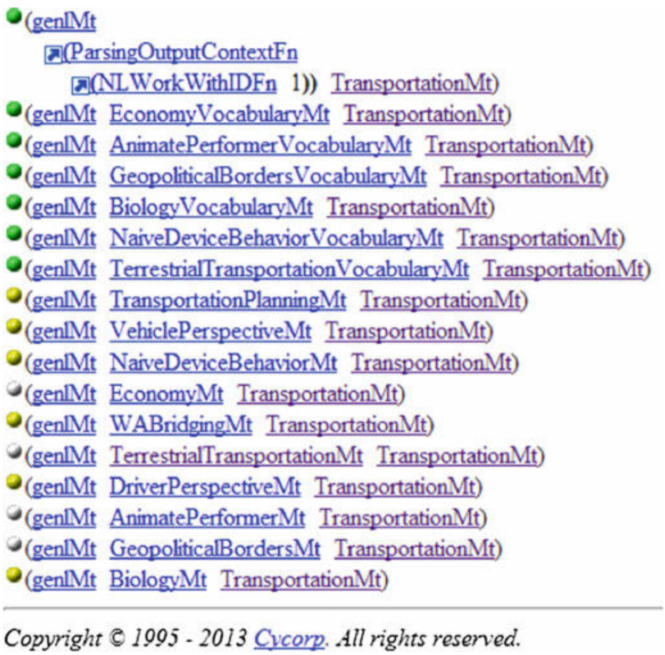


Fig. 5.4 First level sub Microtheories of the TransportationMt

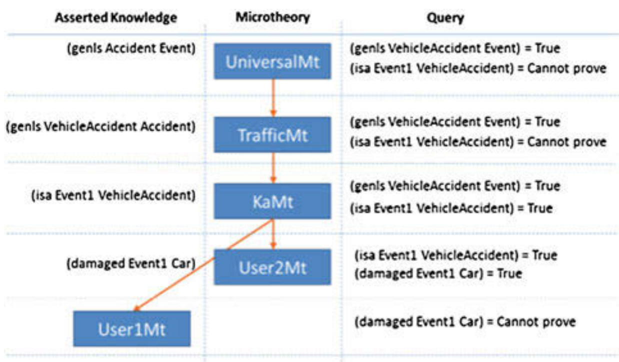


Fig. 5.5 Knowledge base microtheory structure example

Microtheories can access the knowledge in more general ones, but not in the other direction.

For deeper understanding please refer to Fig. 5.5, what it is asserted in the User2Mt cannot be accessed in User1Mt, whereas both Microtheories can accesses the assertions from TrafficMt and UniversalMt respectively.

Inside the Knowledge Acquisition module the authors use these Microtheories to solve different opinions of users and the changes through time. There are two approaches, which need to be further tested through the proceeding of this work. The first approach which and currently the default one is that the acquired knowledge (except the personal data), goes directly into the general KaMt. Each assertion from there is then presented to the other users as a checking question. For example: Is it true that this is a traffic accident?, Is it true that The Bar 5 is a type of Bar. The user then has to confirm or negate the statement. The system then counts the number of agrees and disagrees and if the disagreement comes over some threshold, the assertion is removed.

The second approach goes into another direction. Each answer is always answered to the User Microtheory (User1Mt, User2Mt …). The system then counts how many users gave the same answer to the same question. When this comes over some threshold, the assertion is moved into the more general KaMt, where it can be seen by other users of the system as well. The use of these two approaches also aid in any privacy issues relating knowledge gathering into the knowledge base.

5.7 Approach for Extracting Real-Time Traffic Events from Twitter

The approach presented by this work, adopts a step-wise approach, where several functionalities need to be performed for traffic event detection. Managing traffic events require several analysis components. The first one is related to the system input. Data is obtained from Twitter and has to be filtered in order to have pertinent information. Once a tweet about traffic is detected, information needed to best describe the incident is extracted. Figure 5.6 depicts the step-wise approach adopted in this research.

The objective of this first step is to classify the content of the tweets into two main classes: traffic related events and non-related traffic events. Since this system fits into a big data context, the authors analyse a huge quantity of tweets constantly and without any constraint about the content or structure of a tweet. Therefore there is a need to have a mechanism that automatically filters non-related traffic tweets. The classification adopts a supervised classification algorithm, although this case uses a Support Vector Machine (SVM) classifier. Relevant literature [19, 28], has shown that the adoption of SVMs in resolving problems of this nature, have bring significant improvements. SVM has shown to be a suitable model for this type of problem. The classification is performed in two different phases (learning and testing phase). The SVM start by separating the input data into two classes through a decision function that is learned during a learning process. Once the model has learned how to classify the data, it can be used for classification purposes. In the authors approach, the SVM will tell us if a tweet is referring to traffic or not.

The several steps which compose the proposed approach for extracting traffic events from social media can be described as follows:

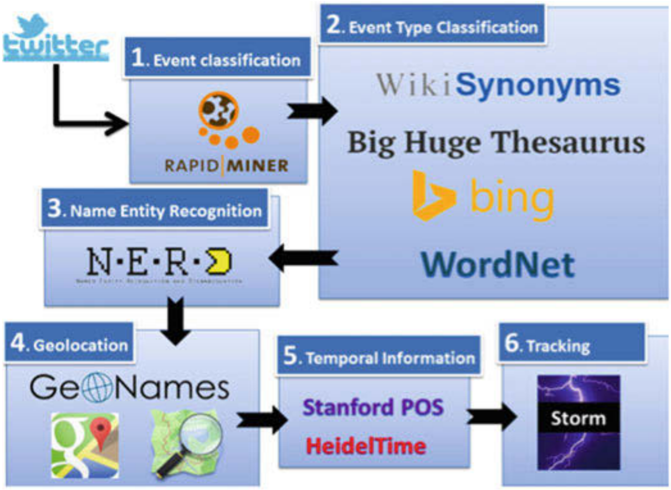


Fig. 5.6 Traffic events extraction from social media

Event type classification: After the filtering in the previous step, and eliminating the non-relevant tweets for this purpose, the event type classification, deals with detecting the type of event each tweet is related with. For this work, 8 different classes of events were considered: traffic jam, roadwork, freight traffic, road closure, ice, wind and snow, traffic accident and others. The objective here, is that each relevant tweet can be labelled with the event type.

Named Entity Recognition: The Named Entity Recognition (NER) step, involves processing a text and identifying occurrences of certain words or expressions which belong to a particular set of categories referred as Named Entity NE [31]. Categories are for example Locations, Products, Name of Persons, Organisms. With respect to the problem being tackled by this work, the most important is to detect the location of the event which the tweet message refers to. The adoption of a NER functionality within this framework, will aid such analysis, in the sense that the authors are particularly interesting in detecting where a traffic event took place.

Temporal Information: When analysing events contained in tweet messages, one particular aspect is to determine the where the event took place. Another important piece of information is also to detect the When the event took place. Also here in this step, several approaches can be followed. One approach could take into account, the time which the tweet message was posted. The authors are particular interesting in analyzing the tweet message, and detect whether the event is happening now, if it is already finished or if it will happen in the future. To do this it is necessary to extract and study temporal expressions and verbs used in the tweet message, supported by natural language processing techniques.

Geolocation: This task deals with, presenting the traffic events analysis in a map using Global Positioning System (GPS) coordinates. The idea is to georeference the locations detected by NER using a Geo-Reference engine (Google Geocoding). A particular aspect to take into account in this step, relates to the accuracy of the georeferencing. The traffic event can be confined to a particular place, or between two different places. For the last case, the presentation of the event in a map is confined into a region.

Real-time Clustering: In this step, the authors are particular interested in analysing the credibility of a tweet messages. The idea is not to analyse a tweet message separately, but in analysing a set of similar tweets. Considering a scenario where only one tweet was posted, which refers to a particular event, during a limited period of time, very little can be concluded about the credibility of the tweet message itself. But, if during that period of time, similar tweets keep being posted, the probability of such event becomes more real. There are other approaches for analysing the credibility of a tweet message, which take into account the historical behaviour of the authors of each tweet. For the moment, such approaches are not being taken into account and are part of future work. For this study not only a tweet but a series of tweets that talk about the same event. For this step, the authors propose the adoption of a clustering algorithm which is performed in real time within a short time window, simultaneously with the crawling (obtaining tweets from Twitter API). The result will be clusters of tweets, which relate to the same event. This will not only increase the credibility of traffic events, but on the same time, it will enrich the information about a particular event with more information and tracking the evolution of such event. Outputs from precedents components are used here. Thus, clustering uses both textual content of a tweet and other information like locations, event type and temporal information.

5.7.1 Preliminary Results

This section describes a preliminary analysis of the results provided the authors research. Classical performance metrics are used to assess the precision and recall highlighted through this research. In this case, precision is the ratio of the number of relevant tweets retrieved to the total number of irrelevant and relevant tweets retrieved. Recall is the percent of all relevant tweets that is returned by the classifier. The authors compute the accuracy and macro F1 score. Accuracy is calculated as the sum of correct classified tweets by the total number of classified tweets, and macro F1 is the average of precision and recall. Table 5.2 shows the testing and training performances, where the authors get very promising results both regarding training and testing phases, similar to performances shown in Schulz et al. [20] and Wanichayapong et al. [19].

The comparison of the different NER engines was performed using 50 tweet messages that were selected randomly from the training dataset. Each tweet is manually extracted to all locations, which make a total of 142 locations across all tweets. The

Table 5.2 Train and test performances in event classification

Train	*Accurracy: 0.893*		
	Negative	Positive	Precision
Prediction Negative	4326	390	0.917
Prediction Positive	674	4610	0.872
Recall	0.865	0.922	Macro F1: 0.894
Test	*Accurracy: 0.955*		
	Negative	Positive	Precision
Prediction Negative	993	83	0.922
Prediction Positive	7	917	0.992
Recall	0.993	0.917	Macro F1: 0.956

Table 5.3 Location extraction performances

	OpenCalais	Alchemy	Stanford	NERD
Detection %	50.7	39.43	35.21	80.98

50 tweets that were selected do not follow any particular structure, but contain a lot of information corresponding to one or more locations.

Table 5.3 shows the percentage of entities detected for each system. The differences in performances lead us to conclude that Named Entity Recognition and Disambiguation (NERD) engine present the best performances because it uses OpenCalais and Alchemy in addition to other NERD systems. Although the percentage of detection is high (over 80 %), there are also entities that are tagged incorrectly as location. For instance, common words like road, junction or eastern can be detected as locations by NERD. However, the precision will be improved in this case as in next step (geolocation) the authors perform a disambiguation of incorrect entities.

5.8 Mobile Traveling Companion

The main motivation guiding this work, relates also to the particularly interest in building AI systems which act as true companions [13], who care about the humans they work with, and who retain and use detailed information about those humans and their worlds to help them build lives that are more enjoyable and more enriching. The idea to retain about the mobile travelling companion is to take advantage of formalized knowledge stored in the KB modelled in the form of facts and rules, and interact with users proactively by suggesting/recommending a set of actions in a user friendly way. These set of suggestions can be exemplified as:

- To make a suggestion to a car driver during a journey, about a nearby gas station with lower gas prices, by inferring that the car was refuelled hundreds of kilometres ago and so needs to be refuelled again.
- suggest car driver to make a stop, because the driver is driving more than 2 h in a row and theirs a nice rest area on route or within that locality.
- assisting user in planning his/her mobility plan taking in consideration users personal agenda, planed meetings, locations and distance between such meetings by enriching such mobility plans with knowledge previously acquired by the KB.

In order to be able to cope with this, the system will have to be able to elicit detailed knowledge of the entities, processes and situations involved in these heterogeneous topics, and of the way they interact to affect a users mobility plans (before, during and after journeys), and be able to apply that knowledge appropriately. To enrich and improve the experience of information gathering and to connect the specific data provided with the needs of other users of the system, crowd-taught companion systems will need to have a good amount of prior knowledge and intelligence. Only then will they be able to ask the right questions at the right time, situate what is learned within existing knowledge, and combined that knowledge to deliver the right advice to the right persons when needed.

One tempting approach might be to provide this increased intelligence itself through human computation, achieving adaptability either by platform enhancements that support complex tasks [14] or by using human agents to decompose the task, but this is unlikely to scale to supporting systems that can cost effectively support millions of users, each of whom needs to be understood in detail.

Ultimately, the flexibility will have to come from the use of rich knowledge representations that can represent complex heterogeneous domains, powerful inference systems that can operate over them to combine knowledge to produce personalised content, and Natural Language systems that can render these representations into human comprehensible form. It is likely that, in the future, probabilistic logics, deep learning [32] and similar techniques will be able to support such operations, but at present, predicate calculus representations, supported by efficient forward inference and natural language generation provide a good balance of power and practicality.

The mobile travelling companion, attempts to integrate the route planner functionalities with Cyc KB. Figure 5.7 depicts a screenshot of the first mock-up where both approaches are integrated in a common environment. The ability to support human computation via rich interactions relies (inter alia) on three AI capabilities:

- Efficient inference from knowledge base state and user context to knowledge capture goals;
- Carefully designed interactions that enable the user who is providing the human computation to understand what is being asked of her or him; and
- Sufficiently natural generation of Natural Language text from logic.

The typical knowledge capture process using inference consists of at least four steps, which all depends on the specific logical vocabulary. These steps consist of:

Fig. 5.7 Screenshot of mobile interface

(1) decide when and what to ask,
(2) convert a logical question into a natural language,
(3) check the consistency and existence of the answer, and
(4) assert the answer into the ontology and/or creating new concepts supporting it.

Although computers have completely reliable means for keeping track of state, human computation providers do not. A human computation based system like the mobile traveling companion must, therefore accommodate stateloss and other features of human cognition. One form this accommodation can take is providing supplemental information, that is not directly relevant to a knowledge capture goal, but which provides a context that allows the person satisfying the goal to do so more accurately or with less effort.

For example: mobile travelling companion often wants to capture several similar kinds of knowledge about an entity in the world. The knowledge base contains complete knowledge of all previously captured facts expressed using a (for example)

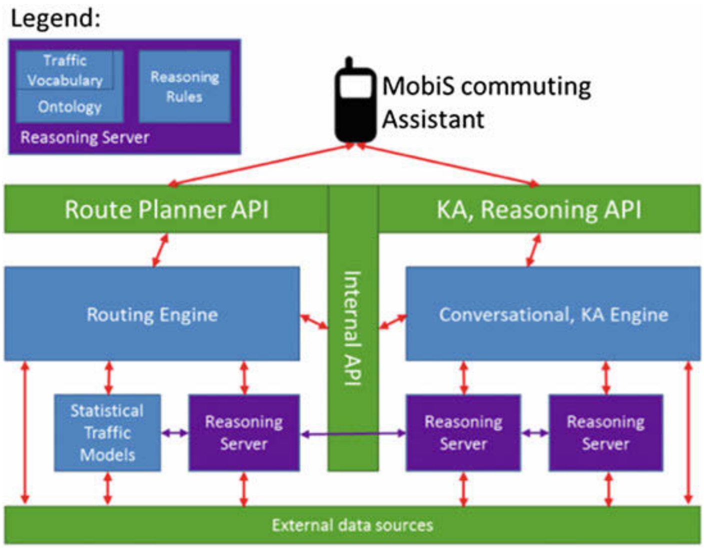


Fig. 5.8 Commuting assistant architecture

a particular predicate and complete knowledge of how the resulting knowledge is embedded in the rest of the KB. From the AI systems point of view, all that is needed to gather more facts using that predicate is to repeatedly ask the same question.

5.9 Technological Architecture

The mobile travelling companion architectural solution (Fig. 5.8) consists of four main building blocks: Mobile and web site client and server, which has two parts. One part is responsible for the statistical and routing requests as described in Sect. 5.4, where another part serves the natural language, knowledge acquisition and reasoning requests as described in Sect. 5.5.

The server parts, as well as their internal modules dont need to be on the same machine and communicate via internal API, where each part sends the relevant data to another side. For performance and scalability reasons conversational engine is communicating with multiple Reasoning modules, which all sync via sync and backup server (represented with arrows).

5.10 Conclusion

The objective of the work described within this chapter is to establish a framework for the personalized intelligent mobility and prepare the ground for the development of a mobile personalized travel companion application. The chapter presents the work that is being done in the MobiS project regarding knowledge formalization and implementation in Cyc.

The initial step regarding KB formalization started by the identification of available data sources and consequently, the description of the methods involved in the KB formalization in Cyc and also the extension of Cyc transportation knowledge with the definition of new vocabulary involving domain experts in the use cases. Logical representations, forward inference, and natural language term recognition and generation provide a practical means to describe interactions with users that allow them to provide knowledge and other computational output to a collaborative system, and provide a convenient mechanism for making use of that knowledge, as it is captured. An important component of making these interactions satisfactory to users is to design them so that machine computation accommodates human abilities.

Moreover, the impact of social media and how can it be applied into a mobile travelling companion, was also addressed here, the objective is to establish a computational framework, able to detect traffic related events in real-time, using social networks. The framework aims to be flexible enough not only to be applied into twitter, but also to other social networks. Semantic analyses were applied to tweets to classify them into a positive and a negative class. Named entity recognition was used to detect locations in a tweet message, and pinpoint those locations in a map. A temporal analysis was also performed, using NLP techniques in order to detect the timeline of the message references.

Despite the challenges associated with the real-time characteristics that distinguish twitter from other social networks and the extraction of information from twitter messages previously described, as future work, some improvements and addons need to be taken into account, namely the aspects concerning the credibility of a tweet message. The authors propose to adopt a real-time clustering approach which clusters messages from a stream of tweets. A tweet message tends to be more credible, if several users post similar messages in a very short period of time. Other factors may also influence the credibility of the messages being tweet, like the profile of each user and historical usage of twitter by each user. Such capability is per se a challenge.

The work presented here is still on an early stage, but some preliminary results have been achieved and lead us to conclude that results are promising. The authors have achieved to deploy the proposed approach from start to end and every component outperforms a very naive or baseline system.

Acknowledgments The authors acknowledge the European Commission for its support and partial funding and the partners of the research project: FP7-318452 MobiS.

References

1. Brabham, D.: Moving the crowd at istockphoto: the composition of the crowd and motivations for participation in a crowdsourcing application. First Monday **13**(6), 1–33 (2008)
2. Bifet, A., Frank, E.: Sentiment knowledge discovery in twitter streaming data. In: Proceedings of the 13th International Conference on Discovery Science, pp. 1–15. Springer, Berlin, Heidelberg (2010)
3. Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., Smith, N.A.: Part-of-speech tagging for twitter: annotation, features, and experiments. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 2., pp. 42–47 (2011)
4. Costa, R., Figueiras, P., MalÚ, P. Jermol, M., Kostas, K.: Mobis—personalized mobility services for energy efficiency and security through advanced artificial intelligence techniques. In: 5th KES International Conference on Intelligent Decision Technologies, pp. 296–306. Sesimbra, IOS Press, Amsterdam, The Netherlands (2013)
5. IBM: An architectural blueprint for autonomic computing. Autonomic computing white paper, IBM Corp., Hawthorne, NY, USA (2005)
6. Fensel, D., Harmelen, F., Andersson, B., Brennan, P., Cunningham, H., Valle, E.D., Fischer, F., Huang, Z., Kiryakov, A., Lee, T.K., Schooler, L., Tresp, V., Wesner, S., Witbrock, M., Zhong, N.: Towards LarKC: a platform forweb-scale reasoning. In: International Conference on Semantic Computing, pp. 524–529. IEEE Press, Santa Clara, New York (2008)
7. Lee, W.H., Tseng, S.S., Tsai, S.H.: A knowledge based real-time travel time prediction system for urban network. Expert Syst. Appl. **36**, 4239–4247 (2009)
8. Tseng, P.J., Hung, C.C., Chang, T.H., Chuang, Y.H.: Real-time urban traffic sensing with GPS equipped probe vehicles. In: International Conference on ITS Telecommunications, pp. 306–310. IEEE Press, New York, Taipei, Taiwan (2012)
9. Chen, C.H., Hsu, C.W., Yao, C.C.: A novel design for full automatic parking system. In: 12th International Conference on ITS Telecommunications, pp. 175–179. IEEE, New York, Taipei, Taiwan (2012)
10. Hung, J.C., Lee, A.M.C., Shih, T.K.: Customized navigation systems with the mobile devices of public transport. In: 12th International Conference on ITS Telecommunications, pp. 113–118. IEEE, New York, Taipei, Taiwan (2012)
11. Chueh, T.H., Chou, K.L., Liu, N., Tseng, H.R.: An analysis of energy saving and carbon reduction strategies in the transportation sector in taiwan. In: 12th International Conference on ITS Telecommunications, pp. 316–318. IEEE, New York, Taipei, Taiwan (2012)
12. Chen, I.X., Wu, Y.C., Liao, I.C., Hsu, Y.Y.: A high-scalable core telematics platform design for intelligent transport systems. In: 12th International Conference on ITS Telecommunications, pp. 412–417. IEEE, New York, Taipei, Taiwan (2012)
13. Forbus, K., Hinrichs, T.: Companion cognitive systems: a step towards human-level AI, pp. 83–95. AI Magazine (2006)
14. Lasecki, W.: Real-time conversational crowd assistants. In: Extended Abstracts on Human Factors in Computing Systems, pp. 2725 – 2730. ACM, New York, NY, USA (2013)
15. Witbrock, M.: Acquiring and using large scale knowledge. In: International Conference on Information Technology Interfaces, pp. 37–42. Cavtat, Dubrovnik, IEEE, New York (2010)
16. Kittur, A., Smus, B., Khamkar, S., Kraut, R.: Crowdforge: crowdsourcing complex work. In: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, pp. 43–52. St Andrews, UK, ACM, New York, NY, USA (2011)
17. Chandrasiri, N., Nawa, K., Ishii, A., Li, S., Yamabe, S., Hirasawa, T., Sato, Y., Suda, Y., Matsumura, T., Taguchi, K.: Driving skill analysis using machine learning: The full curve and curve segmented cases. In: 12th International Conference on ITS Telecommunications, pp. 542–547, IEEE, New York, Taipei (2012)
18. Wu, B.F., Chen, Y.H., Yeh, C.H.: Fuzzy logic based driving behavior monitoring using hidden markov models. In: 12th International Conference on ITS Telecommunications, pp. 447–451. IEEE, New York, Taipei (2012)

19. Wanichayapong, N., Pruthipunyaskul, W., Pattara-Atikom, W., Chaovalit, P.: Social-based traffic information extraction and classification. In: 12th International Conference on ITS Telecommunications, pp. 107–112. IEEE, New York, Taipei (2011)
20. Schulz, A., Ristoski, P., Paulheim, H.: I see a car crash: Real-time detection of small scale incidents in microblogs. In: ESWC 2013 Satellite Events, Montpellier, France, May 26–30. Lecture Notes in Computer Science, vol. 7955, pp. 22–33. Springer, Heidelberg (2013)
21. Singhal, A., Choi, J., Hindle, D., Lewis, D., Pereira, F.: At&t at trec-7. In: Proceedings of the Seventh Text Retrieval Conference, pp. 239–252. National Institute of Standards and Technology, Gaithersburg, Maryland, United States (1999)
22. Paulheim, H., Fümkranz, J.: Unsupervised generation of data mining features from linked open data. In: Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics, pp. 1–12. ACM, New York, NY, USA (2012)
23. Abel, F., Hauff, C., Houben, G.J., Stronkman, R., Tao, K.: Semantics + filtering + search = twitcident. exploring information in social web streams. In: Conference on Hypertext and Hypermedia, pp. 285–294. ACM (2012)
24. Rogstadius, J., Vukovic, M., Teixeira, C.A., Kostakos, V., Karapanos, E., Laredo, J.A.: Crisistracker: crowdsourced social media curation for disaster awareness. IBM J. Res. Dev. **57**(5), 1–4 (2013)
25. Li, R., Lei, K.H., Khadiwala, R., Chang, K.C.: Tedas: a twitter-based event detection and analysis system. In: 28th International Conference on Data Engineering, pp. 1273–1276. IEEE, Washington, DC, New York (2012)
26. Ritter, A., Etzioni, O., Clark, S.: Open domain event extraction from twitter. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1104–1112. ACM, New York, NY, USA (2012)
27. Kumar, S., Barbier, G., Abbasi, M.A., Liu, H.: Tweettracker: An analysis tool for humanitarian and disaster relief. In: Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media, pp. 661–662. (2011)
28. Okolloh, O.: Ushahidi, or 'testimony': Web 2.0 tools for crowdsourcing crisis information. Participatory Learn. Action **59**(1), 65–70 (2009)
29. Yin, J., Lampert, A., Cameron, M., Robinson, B., Power, R.: Using social media to enhance emergency situation awareness. IEEE Intell. Syst. **27**(6), 52–59 (2012)
30. Fung, G.P.C., Yu, J.X., Yu, P.S., Lu, H.: Parameter free bursty events detection in text streams. In: Proceedings of the 31st International Conference on Very Large Data Bases, pp. 181–192. ACM, New York, NY, USA (2005)
31. Mikheev, A., Moens, M., Grover, C.: Named entity recognition without gazetteers. In: Proceedings of the Ninth Conference on European chapter of the Association for Computational Linguistics, pp. 1–8. Association for Computational Linguistics, Stroudsburg, PA, USA (1999)
32. Mirowski, P., Ranzato, M., Lecun, Y.: Dynamic auto-encoders for semantic indexing. In: Proceedings of the NIPS 2010 Workshop on Deep Learning, pp. 1–9. Whistler (2010)

Chapter 6
Exploiting Alternative Knowledge Visualizations and Reasoning Mechanisms to Enhance Collaborative Decision Making

Spyros Christodoulou, Nikos Karacapilidis and Manolis Tzagarakis

Abstract Collaborative decision making in today's knowledge intensive and multi-disciplinary environments is a challenging task. The diversity of these environments and the associated plurality of decision makers' perceptions of the issue under consideration require the exploitation of a variety of meaningful knowledge visualizations and reasoning mechanisms to effectively support the overall stakeholders' collaboration towards making a decision. This chapter reports on an innovative approach that offers a number of interrelated visualizations of the knowledge exchanged and shared during a collaborative decision making process. These visualizations incorporate suitable reasoning mechanisms that exploit human and machine understandable knowledge to facilitate the underlying what-if analysis and aid stakeholders towards reaching consensus and, ultimately, making a collective decision.

Keywords Collaboration · Multi-criteria decision making · Group decision making · Computer-supported cooperative work

6.1 Introduction

Knowledge intensive work is becoming increasingly collaborative in nature. In many settings, multi-disciplinary teams are formed to manage big amounts of data associated with their decision making tasks. Within such teams, decisions are usually collective; the decision making process involves a group of stakeholders, each one having his own perception for the context under consideration [1]. For instance, in the clinico-genomic domain, teams comprising statisticians, biologists and genomic

S. Christodoulou · N. Karacapilidis (✉) · M. Tzagarakis
Computer Technology Institute & Press "Diophantus" and University of Patras, 26504 Rio Patras, Greece
e-mail: nikos@mech.upatras.gr

S. Christodoulou
e-mail: shristod@cti.gr

M. Tzagarakis
e-mail: tzagara@upatras.gr

J.W. Tweedale et al. (eds.), *Intelligent Decision Technology Support in Practice*, Smart Innovation, Systems and Technologies 42,
DOI 10.1007/978-3-319-21209-8_6

researchers have to collaborate and decide how to structure an experiment, which of the available data to take into account and how to interpret the experimental results; in the medical domain, teams consisting of biologists, medical doctors and technicians engaged in clinical drug trials need to elaborate a number of medical treatment issues; in the marketing domain, professionals involved in web-based opinion mining tasks have to consider issues related to which sites to monitor and how to interpret the associated results.

In the aforementioned settings, the decision making process should take into consideration a variety of criteria. Reaching consensus [2] requires, apart from adequate decision making algorithms, diverse collaboration mechanisms for the expression of concerns, conflicting ideas (interpersonal conflicts) [3] and tacitknowledge (knowledge that the members do not know they possess or knowledge that members cannot express with the means provided). Such issues become even more important when decision making is conducted in data-intensive contexts, characterized by great volume, velocity and variety of data [4].

A number of Group Decision Support Systems (GDSS) have been proposed in the literature [5], each of them implementing an aggregation preference method to enhance reaching of consensus among a group of individuals. Among the various aggregation preference methods, Multiple Criteria Decision Making (MCDM) methods [6] were found to be particularly suitable for GDSS since they are interactive, permit multiple viewpoints of a problem and focus on the decision process rather than on its outcomes alone [7]. However, GDSS usually focus on the needs of specific communities and implement reasoning mechanisms suited to a specific type of problem. As a consequence, decision makers facing diverse problems are forced to use separate systems to meet their needs. In addition, GDSS lack media richness as most of the information has to be in textual form. Moreover, limited support is provided for 'what-if' analysis and expression of tacit knowledge. Current systems only inform decision makers about the optimal decision but this is rarely enough; stakeholders require additional information concerning how each decision was made, the parameters taken into account and the processes/data that led to these decisions (decision provenance).

This chapter presents an approach aiming to remedy the above problems by providing an integrated environment that facilitates and augments collaboration and decision making in diverse data-intensive and cognitively-complex settings. The proposed approach builds on the formalization of the collaboration space to provide alternative visualizations that enable both human and machine understandable argumentative discourses. A number of diverse reasoning mechanisms have been integrated to support multi-disciplinary decision makers reach a decision; stakeholders are able to focus on the multiple 'components' of the decision making process (including the mechanism parameters and related data) to realize why an alternative is preferred over another one. The proposed approach has been developed in the context of an FP7 EU research project, namely Dicode (http://dicode-project.eu/).

The remainder of this chapter is structured as follows: Sect. 6.2 presents requirements and challenges related to collaboration and decision making support in knowledge intensive environments; Sect. 6.3 discusses related work in the area of multi-criteria decision making; Sect. 6.4 describes the overall approach followed in the Dicode project, while Sect. 6.5 focuses on the mechanisms supporting collaborative decision making; Sect. 6.6 uses an example scenario to demonstrate how Dicode may be used to augment the quality of collective decision making; finally, Sect. 6.7 concludes the chapter.

6.2 Requirements and Challenges

To meet the challenges associated with supporting collaboration and decision making in diverse data-intensive and cognitively-complex settings, a series of interviews to identify the major issues that stakeholders face during their collaboration practices was performed. These were:

- **Information overload**. This is primarily due to the extensive and uncontrolled exchange of diverse types of data and knowledge resources. For instance, such a situation may appear during the exchange of numerous ideas about the solution of a public issue, which is accompanied by the exchange of big volumes of positions and arguments in favor or against each solution.
- **Difficulty in monitoring social behavior**. The representation and visualization of social structures, relationships and interactions taking place in a collaborative environment with multiple stakeholders are also of major importance. This is associated to the perception and modeling of actors, groups and organizations and their behaviors in the diversity of collaborative contexts. A problem to be addressed is to provide the means to appropriately represent and manage user and group profiles, as well as social relationships given that they are not static but changing over time.
- **Diversity of collaboration modes**. Interviews indicated that the evolution of a collaboration session proceeds incrementally; ideas, comments, or any other type of collaboration objects are exchanged and elaborated, and new knowledge emerges slowly. When members of a community participate in a collaborative session, enforced formality may require them to specify their knowledge before it is fully formed. Such emergence cannot be attained when the collaborative environment enforces a formal model from the beginning. On the other hand, formalization is required in order to ensure the environment's capability to support decision making or estimate the present state of the collaboration
- **Expression of tacit knowledge**. A group of people is actually an environment where tacit knowledge predominantly exists and dynamically evolves.
- **Difficulty in exploiting and integrating legacy resources**. Many resources required during a collaborative session have either been used in previous sessions or reside outside the members' working environment such as e-mails and results from the execution of various data processing algorithms. Moreover, outcomes of past collaboration activities should be able to be reused as input in subsequent collaborative sessions. Such functionality must be provided in ways that do not disrupt or impede an ongoing collaboration.

- **Data processing and decision making support**. In the settings under consideration, timely processing of data related to both the social context and social behavior is required. Such processing will significantly aid the members of a community to conclude the issue at hand (by extracting meaningful knowledge and reaching a decision). This means that their environment needs to interpret the knowledge item types and their interrelationships in order to proactively suggest trends or even aggregate data and calculate the outcome of a collaborative session.

The above issues delineated some categories of crucial requirements to be met during the development of Dicode's collaborative decision making support services.

6.3 Multi-criteria Decision Making

MCDM concerns the evaluation of a number of *alternatives* on the basis of a number of *criteria* (attributes) [6]. Alternatives refer to the different (usually finite) choices available to the decision maker for the problem under consideration, while criteria correspond to the different dimensions from which the alternatives may be viewed. A number of different MCDM methodologies have been proposed, each one suited to address a different type of problem. Major MCDM categories include *elementary methods*, methods based on *Multi-Attribute Utility Theory* (MAUT) and *outranking* approaches [8].

The elementary methods are rather simple; they require small computational effort for the analysis and are more suited to problems with a relatively small number of alternatives and criteria. Methods of this category include: (i) the *maximin* method (the best alternative is considered to be the one with the highest score concerning the weakest criterion) and *maximax* method (the best alternative is the one with the best score concerning the criterion with the highest performance), (ii) the *conjunctive* method (calculates a set of acceptable alternatives, where an acceptable alternative is defined as one which performs above a predefined threshold for all the criteria) and *disjunctive* method (an acceptable alternative should perform above a predefined threshold for at least one criterion), and (iii) the *Lexicographic Decision Making* (LDM) rule (the best alternative is the one with the best performance relatively to the most important criterion).

Approaches based on MAUT are compensatory as they permit trade-offs among the attributes of an alternative (a good performance concerning one attribute may compensate for a bad performance concerning another attribute). Each criterion is associated with a weight to balance the performance of the alternative on the basis of the specific criterion (computing the corresponding alternative's subscore). The total score of an alternative is calculated by aggregating its partial subscores. The simplest method of this category is the Weighted Sum Model (WSM) method, where the alternative's score is calculated as the weighted sum of its subscores. The Analytic Hierarchy Process (AHP) [9] uses a linear additive model to calculate the alternatives' scores, based on pairwise comparisons among the criteria (the relative importance of criterion A to criterion B) and the alternatives (the relative importance of alternative A to alternative B with respect to each criterion).

Finally, the outranking methods are based on the concept of *outranking* to eliminate alternatives that are "dominated". An alternative $A$is considered to outrank an alternative B if there are enough criteria of sufficient importance such that A outperforms B (with respect to these criteria) and there are not any criteria such that A has significantly inferior performance with respect to B. Outranking methods allow two alternatives to be noted as "incomparable". Among the most popular outranking methods are those of the ELECTRE family [10]. ELECTRE I is based on the calculation of the concordance and discordance indices to calculate a partial ranking and choose a set of promising alternatives. ELECTRE I evolved in ELECTRE II, ELECTRE III, ELECTRE IV, ELECTRE IS and ELECTRE TRI. The family of PROMETHEE methods [11] has been designed to help a decision-maker rank partially (PROMETHEE I) or completely (PROMETHEE II) a set of alternatives evaluated on k criteria. The steps of the basic PROMETHEE method include the definition of a preference function for each criterion, a multi-criteria preference index and the preference flows (normed flows), as well as the computation of a complete or partial ranking of alternatives based on the defined preference structure.

6.4 The Dicode Approach

The overall goal of the Dicode project is to facilitate and augment collaboration and decision making in diverse data-intensive and cognitively-complex settings [12]. To do so, it builds on prominent high-performance computing paradigms and large scale data processing technologies to meaningfully search, analyze and aggregate data existing in diverse, extremely large, and rapidly evolving sources. At the same time, particular emphasis is given to collaboration and sense making support issues. The Dicode approach brings together the reasoning capabilities of the machine and the humans and enables the meaningful incorporation and orchestration of a set of interoperable web services to reduce the data-intensiveness and complexity overload in collaborative decision making settings.

Services developed and integrated in the context of the Dicode project are released under an open source license. Services already provided for the context under consideration include (this chapter focuses on the last category of services):

- **Data acquisition services**: They enable the purposeful capturing of tractable information that exists in diverse data sources and formats. Much attention is given to issues such as exploitation of new data sources, augmentation of the data volume, and data cleansing.
- **Data mining services**: These services provide functionality such as looking for subgroups in any user provided data (by searching the rules that cover many target value examples and few non-target values) and recommending similar users or documents from log file data (based on similarity models examples).

- **Collaborative decision making support services**: They facilitate the synchronous and asynchronous collaboration of stakeholders through adaptive workspaces, efficiently handle the representation and visualization of the outcomes of the data mining services (through alternative and dedicated data visualization schemas) and enable the orchestration of a series of actions for the appropriate handling of data. These services provide an interactive search and analysis mechanism for indexing and searching of standard documents. In addition, they aim to enhance (both individual and group) sense- and decision-making by supporting stakeholders in locating, retrieving and arguing about relevant information and knowledge, as well as by providing them with appropriate notifications and recommendations (taking into account parameters such as preferences, competences, and expertise). Services of this category build on an appropriate formalization of the collaboration and exploit a series of reasoning mechanisms to support stakeholders in their daily decision making processes.

Central to the proposed approach is the concept of the Dicode Workbench [13], a web-based application that follows a widget-based approach [14] to enable the seamless integration of heterogeneous services and ensure their interoperability from both a technical and a conceptual point of view. In this regard, semantics techniques have been exploited to define an ontological framework for capturing and representing the diverse stakeholder and services perspectives.

Technically speaking, the Dicode Workbench uses *iframe* elements to display the services (one iframe element per service is used). The service displayed in the iframe may use any of the state-of-the-art web technologies such as HTML5, CSS3, JavaScript, AJAX or jQuery. To integrate a service in the Dicode Workbench, service providers have to follow a number of necessary steps: develop the service (including the implementation of the service logic and the necessary public interface for invoking the service—usually, the exchange of structured information is based on RESTful calls or WS-* (SOAP) [15]), develop the web interface of the service (to allow user interaction with the service), deploy the service and the web interface (both accessible through an URL/URI to the web server hosting the service), and finally register/publish the service in Dicode (service registration includes providing metadata for the service, annotations contained in the Dicode ONtology (DON) [16] and the URI of the service). The Dicode Workbench enables integration of services in two distinct types (it is up to each service's developer to select the most appropriate integration type): at the user interface level (called *light integration*), and at a deeper, semantic level (called *full integration*).

6.5 Collaborative Decision Making in Dicode

Support for collaboration and decision making in Dicode brings together two paradigms: the Web 2.0 paradigm, which builds on flexible rules favoring ease-of-use and human interpretable semantics, and the traditional decision support paradigm, which requires rigid rules that reduce ease-of-use but render machine interpretable semantics. To achieve this, the approach adopted builds on a conceptual framework, where formality and the level of knowledge structuring during collaboration is not considered as a predefined and rigid property, but rather as an adaptable aspect that can be modified to meet the needs of the tasks at hand. The term formality refers to the rules enforced by the system, with which all user actions must comply. Allowing formality to vary within the collaboration space, *incremental formalization*, a stepwise and controlled evolution from a mere collection of individual ideas and resources to the production of highly contextualized and interrelated knowledge artifacts and finally decisions, can be achieved [6].

Dicode offers alternative visualizations of the collaboration workspace (called Dicode *views*), which comply with the above mentioned incremental formalization concept. Each Dicode view provides the necessary mechanisms to support a particular level of formality. The more informal a view is, the greater easiness-of-use is implied. At the same time, the actions that users may perform are intuitive and not time consuming; however, the overall context is human (and not system) interpretable. On the other hand, the more formal a view is, the smaller easiness-of-use is rendered; the actions permitted are less and less intuitive and more time consuming. The overall context in this case is both human and system interpretable [7]. The views supported in the Dicode approach are:

- **Mind-map view**, where a collaboration workspace is displayed as a mind map that enables an informal representation and interrelation of collaboration items, while bearing a set of useful semantics.
- **Formal argumentation view**, which adheres to the IBIS argumentation model [17] and invokes a set of dedicated scoring and reasoning mechanisms to aid users conceive the outcome of a collaborative session and receive support towards reaching a decision.
- **Multi-criteria decision making view**, where a set of multi-criteria decision making algorithms can be executed to rank the alternative solutions.

During collaboration sessions, each user can individually choose the view with which he/she may want to conduct the collaboration. In the following, the three views are presented in more detail.

6.5.1 Mind-Map View

In this view, the collaboration workspace is displayed as a mind map (Fig. 6.1), where users can upload and interrelate diverse types of items. This view deploys a spatial metaphor permitting the easy movement, arrangement and structuring of items on the collaboration workspace. The aim of this view is to support *information triage* [18], the process of sorting and organizing through numerous relevant materials and organizing them to meet the task at hand.

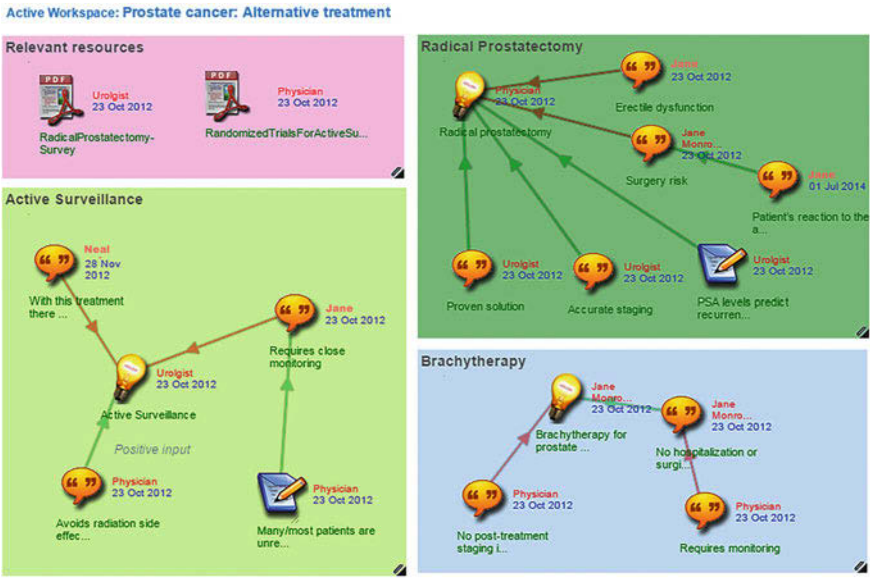


Fig. 6.1 An instance of the mind-map view of a collaboration space with various collaboration items and interrelations among them

While working in the Mind-map view of the collaboration workspace, stakeholders may organize their collaboration through dedicated item types such as ideas, notes, comments and services. Ideas stand for items that deserve further exploitation; they may correspond to an alternative solution to the issue under consideration and they usually trigger the evolution of the collaboration. *Notes* are generally considered as items expressing one's knowledge about the overall issue, an already asserted idea or note. *Comments* are items that usually express less strong statements and are uploaded to express some explanatory text or point to some potentially useful information. Finally, *service* items enable users to upload, configure, trigger and monitor the execution of external services from within the collaboration workspace, and allow the automatic upload of their results into the workspace (as soon as the execution of the service is completed). The service items as well as the results they produce are part of the discourse and can be handled like any other of the available items. Multimedia resources can also be uploaded into the Mind-map view (the content of which can be displayed upon request or can be directly embedded in the workspace). In any case, the set of available item types in the Mind-map view is not fixed; users may expand the existing set by creating new types to be used during their collaboration. This allows them to tailor the discourse to the needs of the problem at hand. Users may rate individual items on a 1–5 scale indicating the importance of each item.

All item types can be explicitly related to express agreement, disagreement, support, request for refinement, and contradiction. Visual cues are used to indicate the semantics of such relationships: for instance, a green-colored relationship indicates agreement, while a red-colored one indicates disagreement. Moreover, the thickness of a relationship may express how strongly an item agrees with or objects to another one. Finally, the Mind-map view provides abstraction mechanisms that enable items to be aggregated and be treated as a single entity within the workspace (see the colored rectangles in Fig. 6.1).

6.5.2 The Formal Argumentation View

The *formal argumentation view* of a collaboration workspace permits a limited set of discourse moves for a limited set of message types whose semantics is fixed and system defined. Following the IBIS argumentation model [17], items of this view include: (i) the *issue* (the problem under consideration), (ii) the *alternatives* (the different choices a decision maker has concerning the problem under consideration), (iii) the *positions* (positions are of two types: "in favor", for supporting, or "against", for refuting another position or alternative), and (iv) the *preferences* (to weigh the importance of two positions).

The formal argumentation view depicts the items created in the mind-map view of the collaboration workspace in a hierarchical way. Collaboration items are laid out in a tree-like structure, where the root (issue) is the title of the problem under consideration and alternatives are nodes appearing as children of the root issue.

Transformation rules allow items appearing on the mind-map view of collaboration workspaces to be transformed into the appropriate abstractions of the formal argumentation view. In particular, specific types in the mind-map view can be configured to be transformed into alternatives, when a transformation is requested. Currently, the default type "idea" is transformed into alternative when the workspace is operated in the formal argumentation view; however, this may vary according to the use case or workspace under consideration and, in general, any type can be specified to be transformed into an alternative. In the mind-map view, all collaboration items linked to items that will be transformed into alternatives, are transformed to form positions (arguments "in favor" or "against") in the tree structure of the formal view, taking into account the corresponding relations in the mind-map view (visual cues are used to specify the semantics of relationships; for instance, a green relation refers to a position "in favor", a red relation refers to a position "against"). Apart from the items earlier created in the mind-map view and depicted in the formal view, the user may use the provided functionality of this view to create new items and interrelations among them (add a new alternative, add a position to support or object to an alternative (or position), add a preference to express the relative importance of a position over another).

Each time an element is added on the formal collaboration workspace, an underlying reasoning mechanism is triggered and, based on the whole tree structure (alternatives, positions and preferences), calculates (and informs users about) the most prominent alternative. The reasoning algorithm of HERMES system [19] has been integrated to evaluate the alternatives. For each alternative, the corresponding alternative score is calculated as the algebraic sum of the weights of the active positions in favour of this alternative minus the weights of the active positions against this alternative as shown in Eq. 6.1.

$$score(e_i) = \sum\nolimits_{\text{in}\_favor_{p_j}} weight - \sum\nolimits_{against_{p_i}} weight \tag{6.1}$$

The formal argumentation view aims to make the collaboration space machine—understandable and exploit the reasoning capabilities of machine to support the decision making process.

6.5.3 *The Multi-criteria Decision Making View*

The *multi-criteria decision making view* (Fig. 6.2) of a collaboration workspace is a read-only view; its main purpose is to further support the decision making process by considering the attributes of the collaboration items appearing in the 'mind-map view'

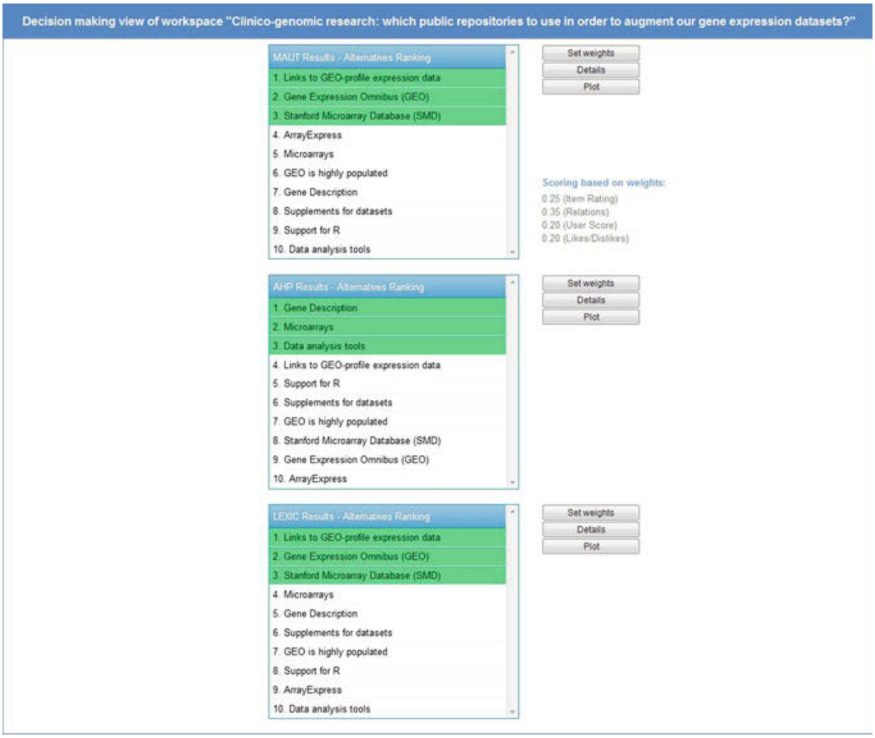


Fig. 6.2 An instance of the Multi-criteria decision making view

and exploiting diverse MCDM algorithms to indicate prevailing solutions. Based on the attributes of each alternative, each MCDM algorithm calculates a corresponding alternative score; the alternative with the highest score is considered to be the best solution to the problem at hand.

In Dicode, four attributes/criteria are used for the evaluation of each alternative:

- *Likes/Dislikes*. The algebraic sum of an item's number of 'Likes' minus its number of 'Dislikes'.
- *Creator Rating*. Calculated as the algebraic sum of all 'Likes' minus all 'Dislikes' corresponding to the items the creator has contributed on a workspace.
- *Relationships in-favor/against*. The algebraic sum of an item's number of 'in favor' relationships minus the item's number of 'against' relations
- *Item rating*. The total rating corresponding to the users' preferences (expressed through an 1–5 rating scale)

The selection of the MCDM algorithms to be implemented in the context of this view was based on a questionnaire filled in by senior decision makers, acting in diverse data-intensive settings. According to the results of this questionnaire, the best suited decision making methodology highly depends on the specific problem under consideration. Depending on the specific problem, decision makers would require support from methodologies that: (i) allow compensation among the attributes/criteria used for the evaluation of the alternatives (a good performance of an alternative concerning one attribute can compensate for a bad performance concerning another attribute), (ii) allow two or more alternatives to be incomparable, and (iii) do not allow compensation among criteria.

Three MCDM algorithms, fulfilling the aforementioned prerequisites, have been implemented in the context of this view: *the Weighted Sum Model (WSM), the Analytical Hierarchy Processing (AHP) and the Lexicographic Decision Making rule (LDM)*. For each algorithm, the user has to set the necessary parameters and, upon the execution of the algorithm, the calculated ranked list of the alternatives is returned. The user may then browse through the detailed results of the algorithm (to realize the reason why an alternative performs better than another one), view the plot with the scores of the alternatives or reset the algorithm's parameters to perform a 'what-if' (sensitivity) analysis [20]. The mechanisms developed in this view build on the reasoning capabilities of the machine to enhance decision making. In the next subsection, the three algorithms implemented in this Dicode view are briefly presented.

6.5.3.1 The Weighted Sum Model

The *Weighted Sum Model (WSM)* is the most popular and probably most used MCDM approach. For a number of M alternatives and N criteria, the best alternative is the one with the top score calculated in Eq. 6.2.

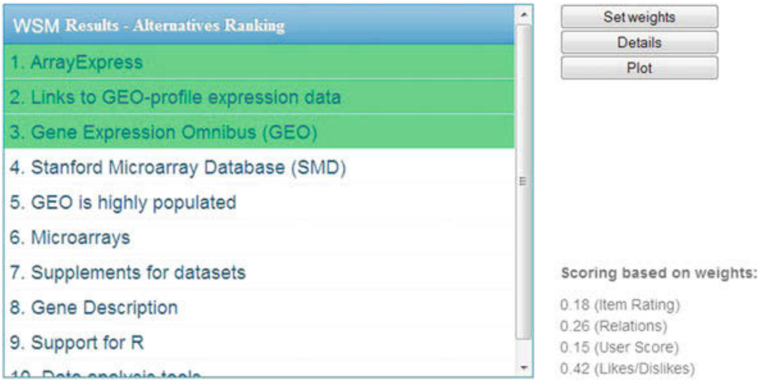


Fig. 6.3 Alternatives ranking based on WSM

$$A^*_{wsm} = max_i \sum_{j=1:N} q_{ij} w_j \quad \text{for } i = 1, 2, 3, \ldots M \tag{6.2}$$

where A^*_{wsm} is the score calculated for the best alternative, N is the number of criteria (for the Dicode case, $N = 4$), q_{ij} is the subscore of the i-th alternative with respect to the j-th factor and w_j is the factor weight (user-defined) reflecting the relative importance of the j-th factor. The output of the algorithm is a list of the alternatives in descending score order (Fig. 6.3).

The user may change the predefined weights of the four factors, browse through each alternative's score and sub-scores (each sub-score corresponds to one of the four factors) or view the plot of the results.

6.5.3.2 Analytical Hierarchy Processing

Analytical Hierarchy processing (AHP) is based on decomposing a problem into a system of hierarchies. Its first step includes constructing a *NxN* matrix (N is the number of attributes) expressing the relative values of a set of attributes. Setting value x to the a_{ij} element of this matrix states that attribute i is x times more important than attribute j. The next step includes constructing N matrices of dimension *MxM* (M is the number of alternatives), where setting the value y in the element b_{ij} of the matrix states that alternative i is y times more important than alternative j (with respect to a specific criteria). Values of x and y are taken from a common scale (the Saaty rating scale—see Table 6.1) used to declare the rel ative importance of an attribute (or an alternative) over another.

Based on the previously described matrices, the *Relative Value Vectors* (RVV) and the *Option Performance Matrix* (OPM) are calculated using the eigenvectors of each table. The vector (VFM) including the corresponding alternatives scores is calculated in Eq. 6.3.

$$VFM = OPM * RVV \tag{6.3}$$

Table 6.1 The saaty rating scale

Intensity of importance	Definition	Explanation
1	Equal importance	Two factors contribute equally to the objective
3	Somewhat more important	Experince and judgement slightly favour one over the other
5	Much more important	Experince and judgement strongly favour one over the other
7	Very much more important	Experince and judgement strongly favour one over the other. Its importance is demonstrated in practice
9	Absolutely more important	The evidence favoring one over the other is of the highest possible validity
2, 4, 6, 8	Intermediate values	When compromise is needed

In the context of Dicode, an open source library (AHP.NET—http://www.kniaz.net/software/ahp.aspx) has been used to conduct all the matrix calculations needed for the AHP algorithm. Concerning the implementation of the algorithm in Dicode, a wizard (Fig. 6.4) is used to perform all the basic steps of the AHP (that include

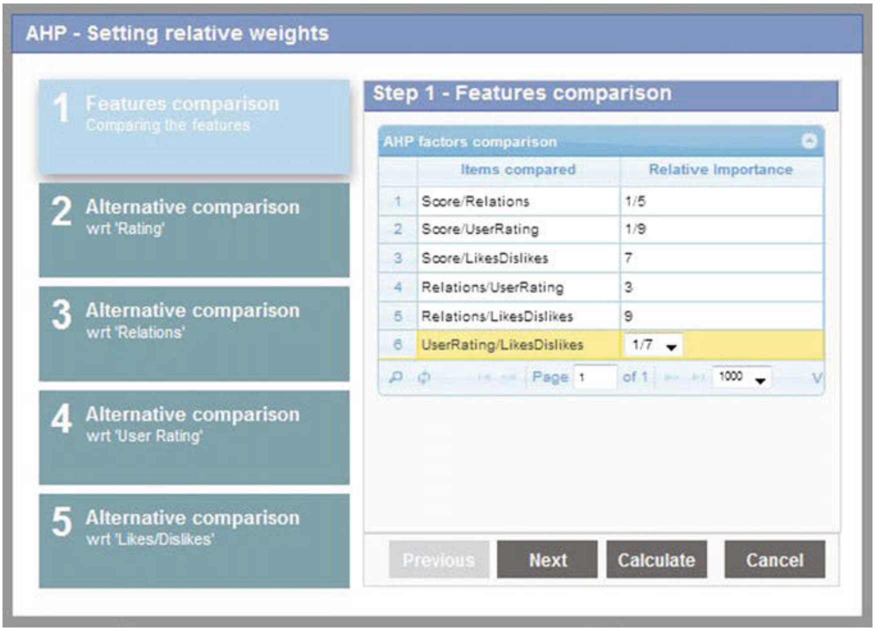


Fig. 6.4 Setting the relative weights in AHP

Fig. 6.5 Sorting attributes with respect to their importance in the Lexicographic DM rule

inputting all the necessary values stating the relative importance among all pairs of criteria and alternatives).

6.5.3.3 The Lexicographic Decision Making Rule

The *Lexicographic Decision Making (LDM) rule* is a decision rule based on ranking the attributes of the decision making process on terms of their importance. No compensation is allowed between the attributes. In the context of Dicode, the user has to rank the four attributes based on their importance. The calculation of the rank of alternatives is based on the partial score (the performance) of each alternative with respect to the most important attribute.

The Dicode user has to rank the four attributes based on their importance (Fig. 6.5). Calculating the rank of alternatives is based on the partial score of the most important attribute. If there are two or more equal sub-scores with respect to the most important attribute, the algorithm moves to the next more important attribute, compares the respective sub-scores and the procedure is repeated until all alternatives are distinguished and ranked (or the attributes are finished, in the case of alternatives with identical sub-scores).

6.6 Scenario of Use

To better illustrate the proposed approach, this section presents an illustrative real-world scenario from the area of prostate cancer research. A physician (George), an urologist (John) and a biomedical researcher (Jane) aim to investigate which is the best alternative treatment for the prostate cancer. Initially, they set up a Dicode collaboration workspace and start using it in the formal argumentation view (Fig. 6.6).

John suggests that one of the best and most popular treatments for the prostate cancer (Fig. 6.6a) is the “active surveillance”. He adds an alternative to make his statement (Fig. 6.6b). Jane is not in favor of this option, because it requires close monitoring (regular digital rectal exams, PSA tests, and prostate biopsy) to monitor for signs of progression, so she adds her ‘against’ position on the collaboration workspace (Fig. 6.6d). Contrary to Jane, George supports the John’s opinion (‘in favor’ position supporting the alternative suggested by John (Fig. 6.6c), in the sense that active surveillance avoids site effects from radiation therapy or prostatectomy. He contradicts to Jane’s opinion (‘against’ position) because, according to his experience, most patients are unreliable as many, or most of them, neglect to visit doctors. On the other hand, he is skeptical as with Active Surveillance there is no post-treatment staging information (‘against position’—Fig. 6.6e).

Jane argues that “Brachytherapy” has been also used to treat tumors in many body sites and this could be one option (alternative). One of its major advantages is that this procedure does not need hospitalization (‘in favor’ position) and, furthermore, there

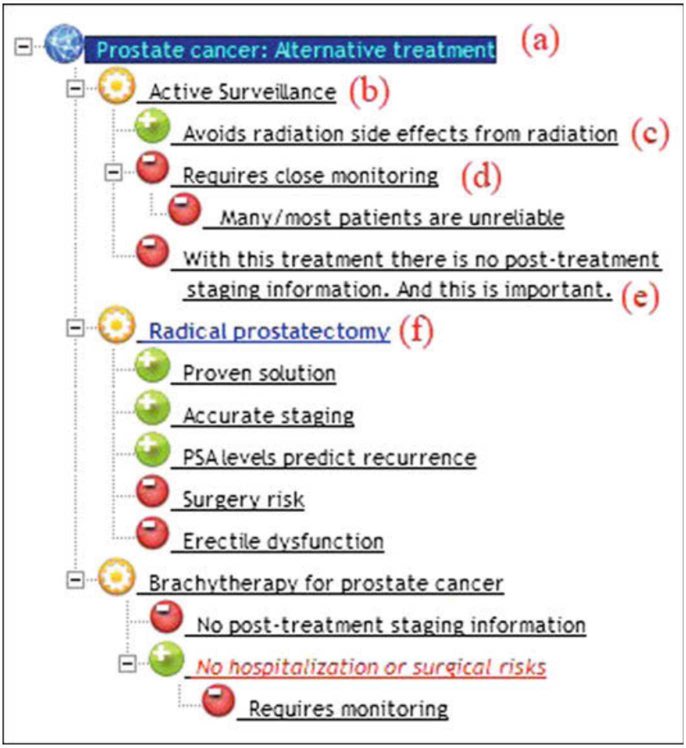


Fig. 6.6 An instance of the formal argumentation view of the collaboration space

are no surgical risks involved. John is not convinced by her arguments as Brachytherapy requires close monitoring ('against' position), which may even include hospital visits. He is so convinced that he adds a preference stating that his position is more important than the one Jane posted before (preference of type 'more important than'). To support his consideration against the Brachytherapy, John denotes that there is no post-treatment staging information which is also an important factor ('against' position).

George argues that the best alternative, in his opinion, is "radical prostatectomy" as it is quite common with very good results. John is in favor of this option ('in favor' position) as this solution is proven to reduce prostate cancer death rates. Moreover, the removed tissue allows accurate stating ('in favor' position), which is very important and the PSA levels may reliably predict the recurrence ('in favor' position). Jane does not share their enthusiasm as, due to surgery, a certain amount of risk is involved ('against' position). Apart from this, an erectile dysfunction is expected at the level of 30–50 % in 5 years. According to the input provided so far by the three collaborators, the underlying reasoning mechanism calculates that the alternative "Radical prostatectomy" is the best argumented/winning one (Fig. 6.6f). It is also noted that the three collaborators, instead of only using the formal argumentation view, could have also used the provided functionality of the mind-map view to express their speculations for the problem at hand (and then move to the formal argumentation view to fire the reasoning mechanism).

Having exploited the functionalities of the formal argumentation view, the above stakeholders have not reached a final conclusion concerning the best treatment for the prostate cancer. Jane suggests using the multi-criteria decision making view of the collaboration workspace, where a number of MCDM algorithms may help them reach a more acceptable decision. They all agree to switch back to the mind-map view of the collaboration workspace to express their likeness/dislikeness and rate preferences on the collaboration items; then, they move to the multi-criteria decision making view.

Jane believes that among the three offered algorithms the one closest to their needs is the WSM, so she sets the respective parameters and browses through the detailed results of the algorithm, the graphical representation of the alternatives scores and the alternatives list. According to WSM results (depicted in Fig. 6.7a), the "Active surveillance" is the optimal alternative to be followed. George believes that the best algorithm to be used is the AHP as it is a very popular algorithm in the area of multi-criteria decision making and allows the pairwise comparison of both criteria and alternatives. He initiates the wizard to set the corresponding AHP weights and calculates the scoring of each alternative revealing that, unlike WSM, "Radical Prostatectomy" is the prevailing alternative (relative AHP scores in Fig. 6.7b). John is in favor of using the LDM rule as, according to his opinion, no compensation should be allowed among the four criteria (in other words, the alternative with the best partial score for the most important criterion should be the winning one). He sets the order of the four criteria and calculates the score for each alternative; the results (Fig. 6.7c) certify that "Brachytherapy" is the best treatment for the prostate cancer. Having used the provided MCDM algorithms, stakeholders compare the respective

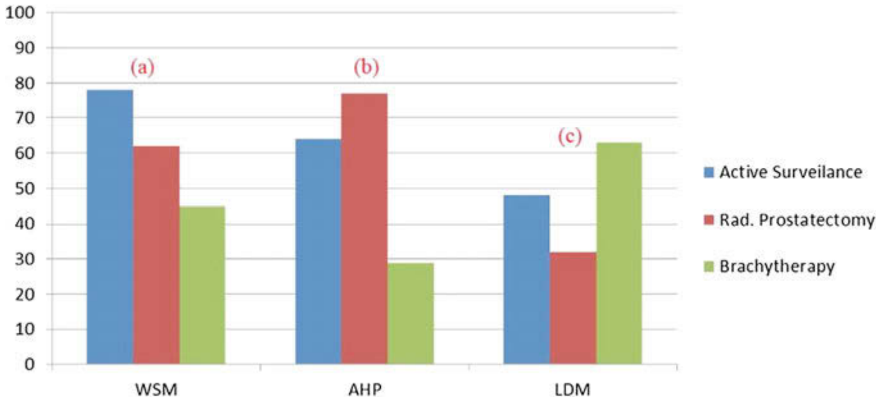


Fig. 6.7 Total scores of the three alternatives calculated after applying the WSM, AHP and LDM methods

results (alternatives' rankings). Through such a sensitivity ('what-if') analysis, they are in a better position to reach a final decision; for instance, they can continue their argumentation towards interpreting the outcomes of the MCDM algorithms provided and reaching consensus on the appropriate treatment to be followed.

6.7 Discussion and Conclusion

Making collaborative decisions in situations involving multi-disciplinary stakeholders with different perceptions remains a challenging task. Traditional decision making systems foster particular reasoning mechanisms aiming at meeting specific needs. On the contrary, the proposed approach integrates a number of reasoning mechanisms into the stakeholders' working context. The integrated collaboration and decision making mechanisms, ranging from simple to compensatory and non-compensatory methods, have been selected to apply to a wide range of decision making contexts. While traditional decision making systems provide limited visualization capabilities, the proposed approach provides a wide range of visualizations, each one offering a varying degree of formality to allow incremental formalization of the overall collaborative decision making context. Work reported in this chapter can be seen as complementary to research focusing on alternative visualizations of argumentation [21].

The major contribution of this work concerns the implementation of a number of mechanisms that are not only capable of displaying the result of the decision making process and enabling a user-friendly 'what-if' analysis, but also of providing additional information concerning how each decision was made and which are the respective processes/data that led to these decisions (decision provenance). Future work directions include the collaborative selection of the appropriate MCDM method to be used as well as the collaborative setting of the parameters that affect the decision making mechanisms; moreover, investigation of additional attributes/criteria to be taken into account in the evaluation of each alternative.

References

1. Jelassi, M.T., Kersten, G., Zionts, S.: An introduction to group decision and negotiation support, In: Bana e Costa, C.A. (ed.) Readings in Multiple Criteria Decision Aid. Springer, Berlin (1990)
2. Ness, J., Hoffman, C.: Putting Sense into Consensus: Solving the Puzzle of Making Team Decisions, Tacoma. VISTA Associates, Wash (1998)
3. Bogetoft, P., Pruzan, P.: Planning with Multiple Criteria: Investigation, Communication, Choice. North Holland, Amsterdam (1991)
4. Challenges and Opportunities with Big Data, White Paper, Computing Community Consortium, Spring (2012)
5. Hwang, C.L., Ling, M.J.: Group Decision Making Under Multiple Criteria. Lecture Notes in Economics and Mathematical Systems, vol. 281. Springer, Berlin (1987)
6. Figueira, J., Greco, S., Ehrogott, M.: Multiple Criteria Decision Analysis: State of the Art Surveys. International Series in Operations Research & Management Science, vol. 78, (2005)
7. Bui, T.X., Jarke, M.: Communications design for Co-oP: a group decision support system. ACM Trans. Off. Inf. Syst. **4**, 2 (1986)
8. Vincke, P.: Multi-criteria Decision-Aid. Wiley, Chichester (1992)
9. Saaty, T.L.: The Analytic Hierarchy Process. McGraw-Hill, New York (1980)
10. Roy, B.: The outranking approach and the foundations of ELECTRE methods. Theory and Decision **31**, 49–73 (1991)
11. Brans, J.P., Mareschal, B., Vincke, Ph.: PROMETHEE: a new family of outranking methods in multicriteria analysis. In: Brans, J.P. (ed.) Operational Research, IFORS 84, pp. 477–490. North Holland, Amsterdam (1984)
12. Karacapilidis, N., Rüping, S., Tzagarakis, M., Poigné, A., Christodoulou, S.: Building on the synergy of machine and human reasoning to tackle data-intensive collaboration and decision making. In: Proceedings of IDT 2011, pp. 113–122. Piraeus, Greece (2011)
13. de la Calle, G., Alonso-Martínez, E., Tzagarakis, M., Karacapilidis, N.: The dicode workbench: a flexible framework for the integration of information and web services. In: Proceedings of the 14th International Conference on Information Integration and Web-based Applications & Services (iiWAS2012), pp. 16–25. Bali, Indonesia, 3–5 Dec 2012
14. Swick, R.R., Ackerman, M.S.: The X toolkit: more bricks for building user interfaces, or widgets for hire. In: Usenix Winter 1988 Conference, pp. 221–228 (1988)
15. Pautasso, C., Zimmermann, O., Leymann, F.: Restful web services vs. "big" web services: making the right architectural decision. In: Proceeding of the 17th International Conference on World Wide Web, pp. 805–814. ACM, New York, NY, USA (2008)
16. Thakker, D., Yang-Turner, F., Lau, L., Dimitrova, V.: Socio-technical ontology development for modelling sensemaking in heterogeneous domains. In: Proceedings of OCAS 2011 Workshop at the 10th International Semantic Web Conference, pp. 60–71. Bonn, Germany, (Oct 2011)
17. Kunz, W., Rittel, H.W.J.: Issues as elements of information systems. Working paper, 1–9. University of California at Berkeley (1970)
18. Marshall, C.C., Shipman, F.M.: III: Spatial hypertext and the practice of information triage, In: Proceedings of the 8th ACM Conference on Hypertext, pp. 124–133. Southampton, United Kingdom, 06–11 (April 1997)
19. Karacapilidis, N., Papadias, D.: Computer supported argumentation and collaborative decision making: the HERMES system. Inf. Syst. **26**(4), 259–277 (2001)
20. Triantaphyllou, E., Sanchez, A.: A sensitivity analysis approach for some deterministic MCDM methods. Decis. Sci. **28**(1), 151–194 (1997)
21. Scheuer, O., Loll, F., Pinkwart, N., McLaren, B.M.: Computer-supported argumentation: a review of the state of the art. Int. J. Comput. Support. Collaborative Learn. **5**(1), 43–102 (2010)

Chapter 7
Decision-Making in a Distributed and Dynamically Scalable Environments

Jeffrey W. Tweedale

Abstract Experiments conducted by the Knowledge-Based Intelligent Information and Engineering Systems (KES) Centre use Java to gain its many advantages, especially in a distributed and dynamically scalable environment. Interoperability within and across ubiquitous computing operations has evolved to a level where plug 'n' play protocols can be used to invoke common interfaces. Designers are able to create dynamic interfaces using reflectance to effectively and efficiently conduct distributed decision-making. Many applications now use mobile agents to support web-centric activities. One example includes: dynamic agent functionality within simulations that automatically adapt to incoming data and/or languages via scripts or messaging. This has been shown using demonstrations described herein. Many are web centric and involve data mining or the use of other types of Intelligent Decision Support System (IDSS).

Keywords Agents · Decision-making · Decision support system · Intelligent agents · Intelligent decision support system · Multi-agent system · Teaming

7.1 Introduction

Intelligent Decision-Making systems provide industry with the potential of transforming digital information into usable knowledge. This domain integrates techniques from within the Artificial Intelligence (AI) domain [1, 2], using Advanced Information Processing (AIP) technology [3, 4] that is programmed based on systems engineering best practice. Information Systems were initially used to process

J.W. Tweedale (✉)
Aerospace Division, Defence Science and Technology Organization, Edinburgh, Adelaide, Australia
e-mail: jeffrey.tweedale@dsto.defence.gov.au

J.W. Tweedale
School of Engineering, University of South Australia, Adelaide, Australia
e-mail: jeffrey.tweedale@unisa.edu.au

J.W. Tweedale et al. (eds.), *Intelligent Decision Technology Support in Practice*, Smart Innovation, Systems and Technologies 42,
DOI 10.1007/978-3-319-21209-8_7

database queries in batch mode. These processes were used to implement repetitive functions associated with payroll and accounting applications. Initially this form of automation was limited to big industry, predominantly because main-frame computers were expensive and required specialist skills to operate. Over time, desktop computers stimulated the introduction of spreadsheets and word processing. Both facilitated the growth of Information Management Systems (IMS) [5] aimed at providing analytical feedback at all levels of industry (including the bank Automatic Teller Machine (ATM) network). As the volume of data grew, corporate logic, data-mining and warehousing techniques were packaged as a single Expert System (ES) that could be deployed using customised databases [6]. Eventually Knowledge Based Systems (KBS) [7] were added to enable organisations to automatically employ corporate rules and knowledge in their Decision Support System (DSS) [8].

The growing *density* of data influenced the efficiency of these systems. Dhar and Stein introduced a series of measures to report on the performance of a DSS. They monitored system accuracy, response time and level of explanations provided. After considering these constraints they were able to specify recommended *courses of action* [8]. Other heuristic measures could be used to measure the quality of utility within a DSS and provide feedback to balance efficiency within specified solutions.

Blackboard system are thought of as a component-based system, where each box can function as a database, series of *pigeon holes* or behave with an unknown behaviour that represents a specific aspect of a system or specific sub-systems. This needs to occur in an environment where experts and modular subsystems, are capable of representing different points of view, strategies, and knowledge formats. These problem-solving paradigms may include, but are not limited to [9]:

- Rule-Based Systems;
- Case-Based Systems;
- Bayesian Networks;
- Neural Networks;
- Fuzzy Logic Systems; and
- Legacy software systems (traditional or formal procedural).

As suggested, experiments conducted by the Knowledge-Based Intelligent Information and Engineering Systems (KES) Centre use Java to gain its many advantages. Over the past decade, KES has created a number of applications using agents, Fuzzy Logic (FL) and Neural Networks (NNs) within a structured Multi- Agent System (MAS) framework. Several examples include: the Cognitive Hybrid Reasoning Intelligent Agent System (CHRIS) [10], Web-basedMulti-Agent System Test-bed (Web-MAT) [11] and Intelligent Multi-Agent Decision Analyser (IMADA) [12]. AI techniques have also made a great deal of progress in other fields. Some include: knowledge representation, inference, machine learning, vision and robotics [2, 13]. Minsky suggested that AI is the science of making machines do things that would require intelligence if done by man [14], however other researchers regard AI as more than applied engineering. The study of AI employs multiple disciplines that require study associated with computers and the sciences relating to human and animal intelligence. Hence, although the notion of AI was initially conceived during the Dartmouth conference, the following teams pursued divergent directions:

John McCarthy and Nil Nillson:	using formal logic as a central tool to achieving AI (considered the *Neats*) [15].
Minsky and Schanks:	using a psychological approach to AI (the *scruffs*) [16].
Newell and Simon:	using production systems [17].
Russel and Norvig:	entered the argument more recently by describing an *environment* as something that provides input and receives output, using *sensors* as inputs to a program, producing outputs as a result of *acting* on something within that program [18].

The AI community now uses the later to create an environment that reacts to sensing (inputs) and acting (outputs). Simply speaking, an AI application is a system that possesses knowledge about an application domain that takes in data from its environment, and reasons about that data to derive information. This information is then used to make decisions to be acted upon [19]. That system may be a combination of algorithms used for reasoning, learning, planning, speech recognition, vision, and even language translation. For example, an ES used by an organisation or a robotic drone employed in autonomous activities. Data, information and knowledge can be acquired, stored, analysed and employed in almost any field. The technology used for data mining is still evolving, however the essential elements are well established. DSSs employ these techniques to manage information repositories. Researchers are then free to employ decision-making algorithms or other techniques to expand the intellectual capabilities of most traditional ESs. Here, intelligence is considers to be the cognitive aspects associated with human behaviour, such as; perceiving, reasoning, planning, learning and communication. The knowledge gathered can be represented as a set of "IF ... THEN" statements that are stored as rules in a database. In this case, the rules consists of a set of conditions and a conclusion. If all of the conditions are true, then the conclusion holds. When reasoning, an ES will be used to *infer* many rules. The conclusion from one rule can form part of the IF condition of successive rules sequentially in the process of *seeking* a prescribed goal.

Many aspects of DSS models involve web centric transactions and typically employ data mining or the use of more modern Intelligent Decision Support System (IDSS) techniques. This chapter discusses the use of decision-making within a distributed and dynamically scalable environment. Section 7.2 introduces the basic concepts of DSS, Sect. 7.3 discusses decision-making, while Sect. 7.4 explores IDSS enhancements. Section 7.5 explains how agents use a multi-lingual dynamic environment, while Sect. 7.6 highlights conclusions and future research direction.

7.2 Background

Prior to first World War (WW1), computers were introduced to extend the processing capabilities of humans. A combination of development in this domain facilitated the birth and/or expansion of a number of associated fields. Software examples include: LISt Processing (LISP) and Dendral, while Advanced Research Projects Agency Network (ARPANET) and Manufacturing Information and Control System (MIACS) represent hardware related enhancements [20].

LISP: McCarthy developed LISP—Inversion: A Programming Language Interface (APL). It can manipulate source code as a data structure, giving rise to heuristic macros that allow programmers to create new syntax that operate on parenthesized lists [21].

DENDRAL: Feighambaum designed the *DENDRitic ALgorithm* to topologically resolve distinct arrangements of any given set of atoms. This was considered the first large-scale expert system to provide automated responses based on heuristics measures [22, 23].

ARPANET: is an operational, resource sharing inter-computer network linking a wide variety of computers between the US, Hawaii, Norway, and England. [24, 25].

MIACS: Bachman invented MIACS while at General Electric. This led to the first *Integrated Data Store* which was able to store up to 40 kB of share files using a data-base management system [26].[1]

These early success stories stimulated the concept of data exchange and *transactional processing* techniques. Data-driven methodologies progressively evolved using lists, databases and eventually data warehousing. Data Mining, ESs and DSSs inputs are still based on a list of related facts that need to be tested or associated prior to being interpreted. These decisions are increasingly being made using AI techniques. For instance, organisational decision are typically derived from a vast amounts of data, information and/or knowledge that is related to the problem space prior to selecting an appropriate course of action. The appropriate mix of information depends on the characteristics of the decision making context [27], and particularly environmental aspects that influence value-based decisions [28]. The current approach of developing DSS to support a specific organisational decisions requires experts to tailor it corporate related information streams. These systems requires the developer to determine what priori information needs to be processed to perform the appropriate filtering a value-adding to assit decision-makers. A Knowledge Management Systems (KMS) can be introduced to manage large amounts of information that may or may not be related to traditional decision-making processes. KMSs assist the user

[1]The IDS provided application programmers with a set of powerful commands to manipulate data, using an early set of expressions that would soon be called a *Data Manipulation Language*. This was also used by SAGE and considered to be the first DSS.

Table 7.1 Perceived knowledge hierarchy [30]

Element	Description	Example
Data	Collection of facts	Number of units sold
Information	Converted or related data that have meaning in context	Average sales per month
Knowledge	Interpreted information based on expertise	Company may have future financial problem
Wisdom	Knowledge synthesized and applied to make a decision	Company should consider a merger

to systematically trawl though massive amount of data to determine the information, within the context, to make decisions without causing information overload [29].

A coherent system is required in order to integrate a KMS into the corporate DSS. Michalewicz et al. provides an mechanism to translate a general class of decision problem using promotional phases of information to processes intelligence prior to making decisions. As stated previously, this concept combines successive Knowledge Management (KM) processes to transform data into information, then knowledge and ultimately wisdom. Computers can be used to transform Data into Information and ultimately Knowledge into Wisdom. Technologies, such as; ESs, DSSs and IDSS can all be employed to transcend one or more of these categories. Table 7.1 describes the categories used in this chapter.

Knowledge is an elusive concept that generally encompasses both explicit know-how and tacit know-what [31, 32]. Explicit knowledge can be articulated and codified as, for example, a set of rules or guidelines, and it can be captured in KMS. Tacit knowledge is more difficult to quantify. It is based more on experience, culture and social norms [20]. Some authors have proposed that operations and management activities of a company are primarily task-based and knowledge-intensive, and that, in these cases, tacit knowledge can be codified if it is extracted from previous tasks [33]. The implication is that under some circumstances tacit knowledge as well as explicit knowledge can be stored and used for decision making [30].

Figure 7.1 introduced a mechanism to allow the user to digitally adapt specific problems based on their business systems [34]. Here, the adaption module is implemented with AI techniques. These techniques make it possible to envision a system that can place the decision into context. Some progress toward this goal has been made for applied problems. For instance, dynamically evolving disaster situations, such as a pandemic outbreak, were people need to interact and address the problem. Each of these external interactions may influence the next decision. This explains the need for an adaption module, which serves as an interface between the decision maker and the requisite model. This allows the model to define or redefine the data and processing needs to generate information. Information is then transformed into knowledge in the KMS by mining previous experience. Knowledge is then passed to the adaption module to assess and modify it for the specific decision problem. The system provides a proposed decision to the human user, who may accept, reject or

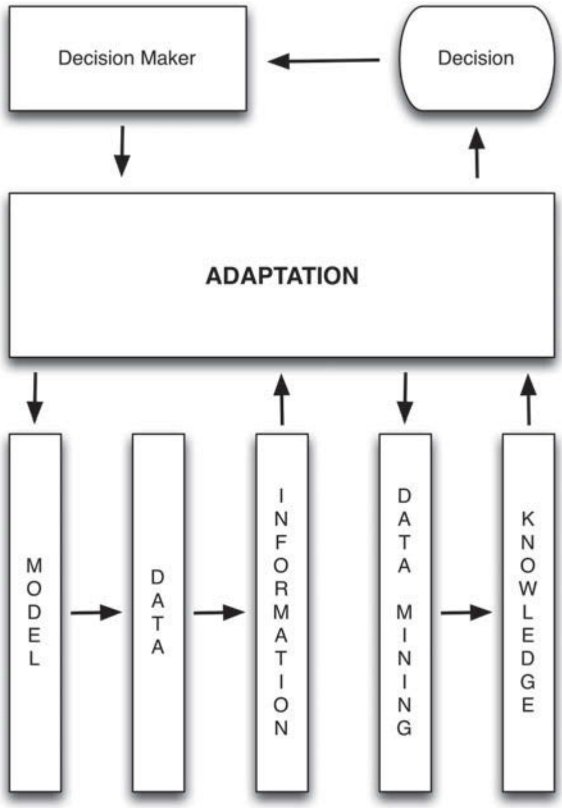


Fig. 7.1 Architecture of an adaptive decision support system [30]

modify the parameters. The user is central to the system and has the final decision on accepting or rejecting the recommended course of action prior to implementing. Users also interact with the system to provide user-specific domain knowledge or preferences [30].

In dynamic environments, systems require a capability that allows the decision-maker to adapt to change. Similarly the solution framework must be capable of solving a range of problems. Michalewicz et al. describes four characteristics he observed when dealing with complex problems [34]. These observations can be used by the adaptation module in the above example to influence decision-making, based on:

1. the number of possible solutions being so large that all possibilities cannot be examined;
2. the environment changes with time;
3. how heavily the problem is constrained; and
4. the number of conflicting objectives.

All adaptive business intelligence systems therefore require prediction and optimization processes to recommend near-optimal solutions. This could take the form of an adaptive learning module to improve future recommendations. More modern IDSS would either select the data analysis technique or assist the user with the selection. The resulting knowledge would consist of the results from one or more algorithms (typically hosted in a MAS micro-simulation system) [20, 35]. The adaptation module discussed could be used to reflect the most recent data associated with a dynamic environments and assess the best solution [30, 36].

7.3 Decision-Making Systems

A database becomes a support system when it can be used to help managers make useful decisions. A DSS must value add to the data being collected. This can be achieved by employing a knowledge-base designed to infer corporate minded responses. The most basic form of DSS would be a monitored environmental control system. One where data is collected and managed to maintain the environment of any given habitat. Traditionally a simple thermostat uses a temperature sensor to extract precise data about the environment, while its comparator[2] sets the threshold used to switch between heating or cooling modes. A series of these sensors can be deployed within a building to create a centrally managed system. The DSS simply collects data and waits for the temperature to exceed the limit set by the operator (rules). This would stimulate an output that can be used to mode switch. More complex environmental control systems may be required, where that system controls a building, a complete facility or even a complex variety of environments that are dispersed geographically. Regardless of the technology used to generate the decision, the outputs relate to a predefined set of judgements based upon readings that change within the environment being controlled.

A DSS often uses Computational Intelligence (CI) techniques to collect, fuse and analyze data in order to derive a decision or enhance a humans ability in a given environment.[3] An expansion of research into a variety of DSSs created greater confidence in the decision being generated by computers [37]. The context could be used to

[2]This is the logic center contains the actuators and reference limits which are used to decide to heat or cool.

[3]McCarthy developed LISP, enabling Bachman to create the first database management system which was converted into an Intelligent Decision Support (IDS) application called SAGE, while Feighambaum labelled his expert system DENDRAL.

Table 7.2 Decision support system domains categorised [38]

Category	Domain	Example
Data Mining	Data driven	Lists, data bases and data management systems
Evolutionary	Genetic algorithms	Optimisation and search spaces
Thought	Neural networks	Learning and pattern matching
Constraints	Rule based systems	Expert and knowledge based systems
Symbolic	Fuzzy logic	Transforming the ambiguity of crisp into fuzzy sets
Temporal	Case based Systems	Reasoning and analogy based systems
Inductive	Machine learning	Iterative creation of dynamic rule sets

determine the problem solving technique used. Examples include: communication-driven, data-driven, document-driven, knowledge-driven and model-driven architectures.

The growing volume of data had an overall effect on the efficiency of these systems. A series of tools where developed to measure the validity and results of many DSS. Factors such as; availability, accuracy, user response and operator ability, were raised as constraints to be considered before specifying courses of action [8]. The history of these systems has bee well documented. Power [38] for instance gives an excellent historical description of DSSs, which discusses many of the fundamental issues, however misses the vital human aspects. Keen reports on a study relating to "the theoretical aspects of organizational decision making". A preliminary study was done at Carnegie in the 1950s, while a more concise study from Massachusetts Institute of Technology (MIT) was completed in the early 1960s [39]. For instance FL can be used to determine confidence levels and subsequently specify the communication requirements for each human role involved in the interaction [40]. Similarly, multi-agent planning and coordination techniques have successfully been used to monitor resource management in command and control systems for several combat ships when defending against incoming threats [41].

The follow research continues to produce more refined DSSs within a number of categories. Table 7.2 lists seven categories (each has been interpreted by this author).

Obviously DSS cannot be used to solve all problems, but they have proven to be a versatile tool to be used when data can be relied upon to making sound decisions. The trend is moving towards value-adding, by making intelligent decisions, based on the concept of supervision. For instance, an IDSS used in machines that must react to changes in its environment based on planed activities or goals.

7.4 Intelligent Decision Support System

As suggested above, DSS are transactional systems that use databases. As these systems evolved they increasingly accessed data across networks, using decentralised databases. Similarly, IDSS now interrogate the Internet of Things (IoT) across dynamic connections to use a growing web of remote devices for enhanced decision-making. Interaction with specified nodes within that web can be achieved using Remote Procedure Calls (RPCs) where objects can be requested and accessed by many computers around the world. Some of the early standards used in distributed computing include: Distributed Computing Environment (DCE), Distributed Component Object Model (DCOM), Common Object Request Broker Architecture (CORBA) and Remote Method Invocation (RMI) in Java; which all provide an object-based RPC communication mechanism. CORBA enables the communication between software components and procedures that are written in different programming languages, across distributed platforms [42]. Unfortunately CORBA messages only include procedures and do not contain any semantics. Web-Services Description Language (WSDL) not only introduces the concept of semantics, it can be thought of as a dynamic Interface Description Language (IDL) because it can specify a *service endpoint* that can be bound during run-time and support *interface inheritance* directly.

Different strategies have been developed to enable the communication of software messages or the exchange of whole components/objects to exchange information and knowledge between applications [43]. For instance, the DARPA Agent Markup Language—Services (DAML-S) is another example of component software techniques maturing. Its heritage originated from procedural logic, OWL Services (OWL-S) and uses web services to communicate. As the name suggests, it is a service connected to a protocol. In this case, OWL-S and Universal Plug and Play (UPnP) assist in providing an ontological sound exchange of data across a web service. Over time OWL-S has metamorphosed into WSDL, but it has a limited form of expression that restricts it being mapped to DAML-S. Despite these restrictions, DAML-S is still superior to WSDL. The relationship between both is shown in Fig. 7.2.

The above methods facilitate seamless transaction process for any data source. These protocols allow designers to componentise DSS and IDSS using agent frameworks. Similarly these agents can be used to facilitate alternate decision-making processes. For instance may databases incorporate min/max attributes to monitor the need to re-supply and/or track patronage habits by counting transactions. Similarly multiple Intelligence Agents (IAs) could be incorporated to implement additional function independent to data collection and/or analysis. For instance providing customised reports for management, shareholders and/or independent stakeholders [45]. Wooldridge describes an IA as a computational systems that has properties associated with; autonomy, social ability, reactivity, and pro-activity [46]. Chira recently added mobility, with the ability to learn, reason, cooperate and coordinate [47]. All these characteristics promote the ability to incorporate component-based intelligence capabilities within IDSS.

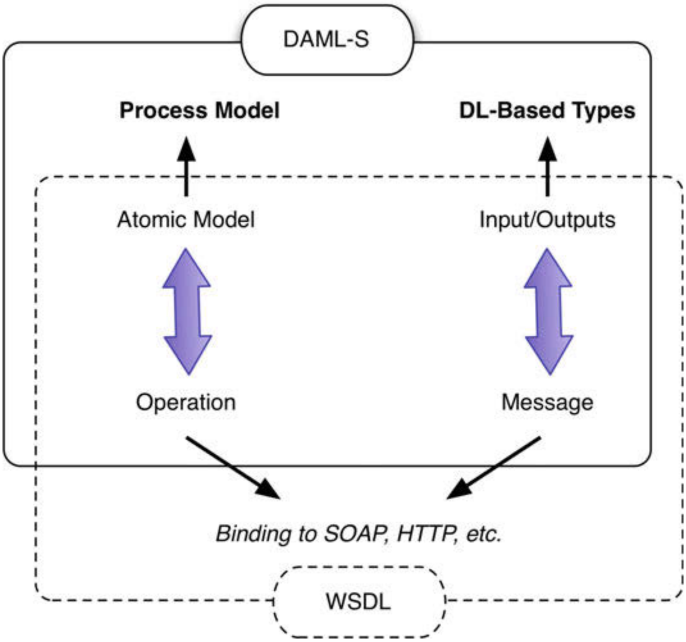


Fig. 7.2 Mappings from DAML-S to WSDL [44]

Figure 7.3 depicts one possible approach to incorporating agent technology within an IDSS. Here intelligence is described as a computational process that enables enhanced decision-making and interaction. The agents are required to bridge the gap between humans and machines. They also integrate information to form social structures, such as; teaming and learning. According to Tweedale et al. [48], new research and technologies are beginning to focus on developing human-centric MAS in order to achieve mutual goals. These agents must also react to changes in the environment (reactivity) and plan activities to solve their goals (pro-activity) [49].

This IDSS contains multiple agents to manage the DSS functions, Human Machine Interface (HMI) and intelligent decision-making capabilities. Agents are deployed to acquire data which a continually changing environment and reports this to the Inference Agent. This data is also processed into usable information and ultimately transformed into knowledge (using a set of beliefs about their perceived environment). The Inference Agent accesses the expert knowledge in terms of production rules and continually interprets the knowledge created and updated via the collection agents. A parent agent supervisors this process and coordinates their interact within the environment, which for this example, consists of pre-processed information about various platforms and information servers. In t his case three independent servers are used to highlight the distributed nature of the project. These include: My SQL, Oracle and ODBC servers [50]. The IDSS also contains a user interface to enable

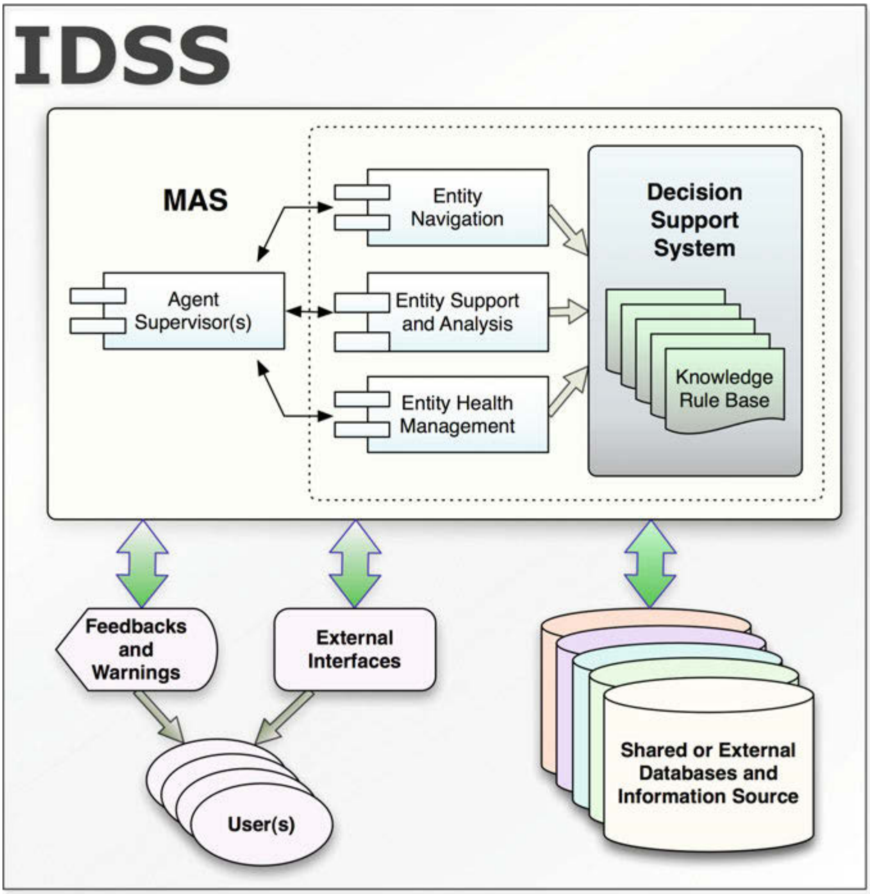


Fig. 7.3 The Principle of MAS within IDSS [20]

user-requested analysis. The outputs of the IDSS include reports, warnings and other feedback about the current state of the system [51].

While at KES, Haider developed a platform independent IDSS using Commercial Off-The-Shelf (COTS) Commercial Off-The-Shelf (COTSsoftware [52]. To fulfil these considerations, two software packages were selected. These included Java and JACK. Java was chosen as a development language for following reasons [53]:

- Portability;
- Platform Independence; and
- Extensive development support.

The JACK Intelligent Agent framework was used to implement the software agents, based on the following reasons [54]:

- Extension of Java;

- Graphical representation of agents; and
- Specifically written to implement software intelligent agents.

JACK is an event driven software development platform where software agents are developed to implement proactive and reactive strategies. In the developed IDSS, each agent is developed to continually monitor the assigned environment. Each agent consists of one or more plans to execute certain logical operation based on particular events or based on its own beliefs [51]. Prior to being populated with operational data, the example IDSS was successfully tested to provide alerts and warnings based on hand coded scenarios.

7.5 Dynamic Environments

Many applications now use Web-Service style component technology and type safe languages (like Simple Object Access Protocol (SOAP)), to preserve semantics, ontology and intent of the message transaction. Amazon, Yahoo and Google all use electronic kiosk style applications to transform requests from semi-intelligent information retrieval, into information rich responses, that are tailored using some form of server-side DSS technology. The relationship is implied through the complementary use of OWL-S and WSDL. Both languages are NOT covered in the same domain, however the overlap is obvious. WSDL uses Extensible Markup Language (XML) while Ontology Web Language (OWL) is logic based (classes) and the binding is specified in the form of WSDL (SOAP and HyperText Transfer Protocol (HTTP)). As discussed, the "resulting rich service descriptions provide a powerful way of assembling information resources in contexts that require the agile construction of virtual organizations" [55].

A series of follow-on developments resulted in a logic based protocol that could be catalogued into libraries and distributed in the format derived for WSDL using an XML protocol over HTTP. "This rich service provides a powerful mechanism of assembling information resources in context that require the agile construction of virtual organizations" [56].

A *Concept Demonstrator* called the Supervised Inter-operable Dynamic Agent Teaming Environment (SIDATE) was also developed at KES to investigate the simulated interactions between agents using real world parameters (Fig. 7.4) [20]. The system output is also shown in Fig. 7.5 [20]. The HMI reveals the realisation of loosely coupled agents that subscribe information, events and capabilities from an internal IDSS. An MAS with sensors and a navigation capability are simulated. This enables Agents to share knowledge and collaborate based on the context and their expertise to achieve specific tasks. The capabilities of agents can be extended by appending other agents' capabilities (like turn, hide, run, survive and assist)[4] until

[4] An Agent with the capacity to avoid collisions or optimize path finding would retain specific functions, however data and context used to achieve these would be derived and stored centrally.

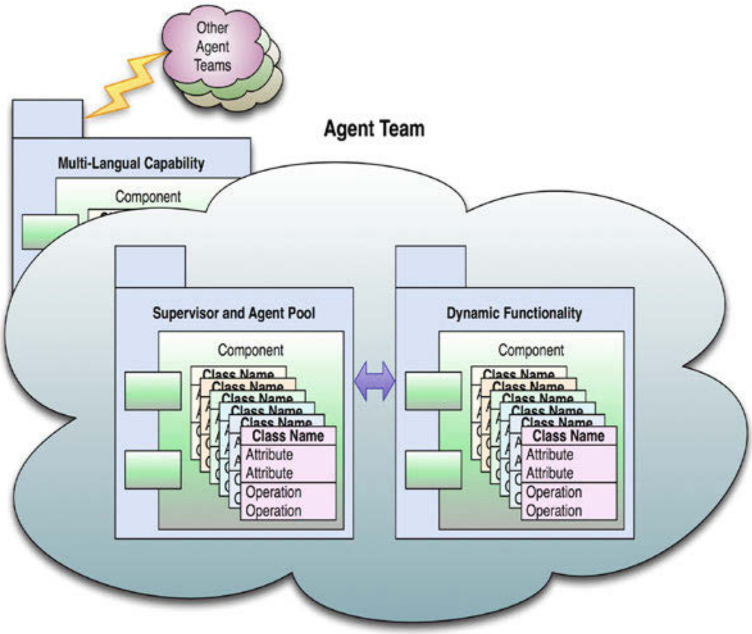


Fig. 7.4 Supervised inter-operable dynamic agent teaming environment [20]

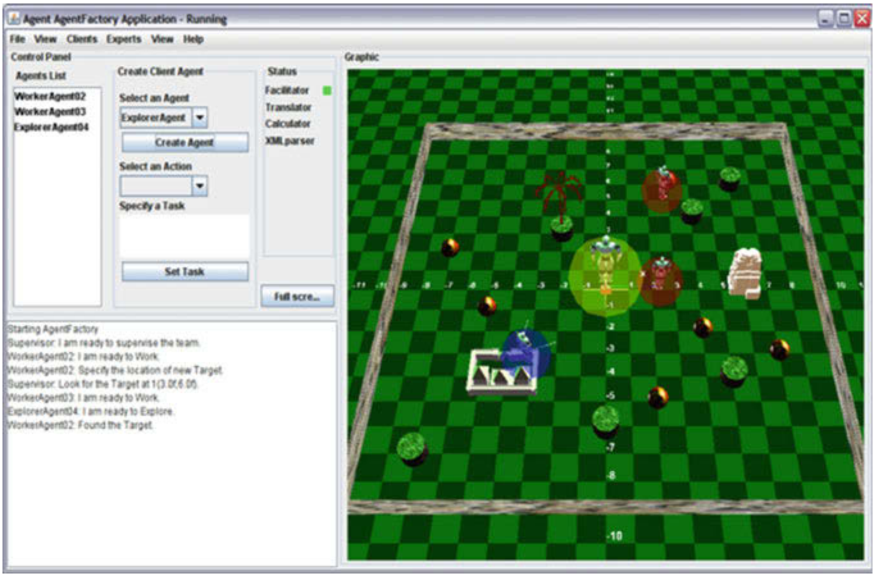


Fig. 7.5 Agent factory demonstrator [20]

the goal is achieved through cooperation. More detail about the model can be found in the original article [11].

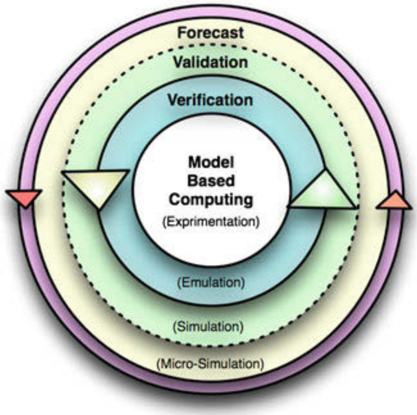
The dynamic environment consists of black-box system (see Fig. 7.4), which uses an MAS to realize many of the core capabilities [20]. Some include: a KBS, physics engine, navigation tactics, and mathematics models to enable planing and re-planning capabilities. The IDSS is used to store mission data and generate decisions based on the corporate knowledge loaded in the KBS. The MAS comprises of different layers that form a hierarchical structure based on the Trust, Negotiation, Communication (TNC) framework [57]. An entity can be a single agent or a complete subsystem in which capability can be dynamically linked during run-time to retain its autonomous ability.

The server is the center of system that translates and transmits data between the destination and source client via SOAP messages. The server can also run the source client on the same machine. The destination client runs a simulation environment where its entities are guided by the MAS in the source client to implement a distributed system over network.

The entities in the source client's simulation environment publish their geographical location to the destination client's entities, which move in sympathy with the source client when connected via the server's translation component. Every set amount of time the destination client requests the source client via the server for new position coordinates to navigate its entities based on agent's movement in the MAS. The coordination for this target point can be obtained from the source client via the server's translation component. Random events, entities and constraints can be introduced at will using the scenario script.

Today, researchers have the capacity to stimulate, emulate and even simulate constrained real-world problems. Fast hardware is required to achieve the number of iterations for real-time simulation. Most simulation system use a feedback/forward mechanisms to chain environmental and entity relationships. The fidelity and capacity to solve problems reduces as the number of variables or constraints increases.

Fig. 7.6 Model based computing reaches the boundary of micro simulation [20]



Using super computers, researchers are able to run some simulation faster the real-time. As the power of these machines increases, the number of successive simulations will increase. Eventually a series of micro-simulations could be used to determine simulated results for a variety of configurations. These results would then be used to determine more intelligent decisions in future IDSS. Figure 7.6 portrays the future concept of how experiments might interact to make decisions [20]. Supervised Interoperable Dynamic Agent Teams would form the basic nucleus of this system and must be capable of providing the process when the technology is ready. Further work on the black-box architecture, how its modules will communicate and the drive to create an efficient environment capable of dynamic functionality at run-time will continue together with components that *plug-n-play* that enable human interaction, learning, cooperation, planning and re-planning (this list is only limited to the students imagination and the practicality of their components functionality). Similarly increase processing capabilities would facilitate increased scale and felxibility within the simulated environment. For instance Moore created a more advanced decision-making capability called the Atomic Layer Deposition (ALD). This simulated the self-assembly of molecules, one mono-atomic layer at a time. To gain some perspective of these gains, 100 of these new transistor gates would fit inside the diameter of a single human red blood cell [58]. Eventually technology will surpass the needs for CI and fully scalable MAS capable of solving real world problems. Until then, we need to ensure our designs are tested and capable of such functionality to enable it to scale to realistic sizes capable of solving complex problem.

7.6 Conclusion

This chapter progressively discusses the use of decision-making techniques and KM techniques to enhance the functionality of existing DSSs. Several examples were discussed to highlight database management concepts associated with a distributed and dynamically scalable environment. It also indicates that DSS models involve data centric systems, were more modern IDSS use web centric transactions. The examples used reveal the basic concepts of DSS, decision-making and explores IDSS enhancements. Ultimately more work is required to further enhance decision-making processes, however massively parallel super computers are now facilitating constrained experiments. As discussed, model-based computing may provide one of those solutions using micro-simulations within an IDSS.

References

1. McCarthy, J.: Programs with common sense. In: Symposium of the National Physics Laboratory. Volume 1(2) of A Learning Machine, Part 1. Mechanisation of Thought Processes, IBM J. Res. Dev., London, pp. 77–84 (1959)

2. Grevier, D.: AI—The Tumultuous History of the Search for Artificial Intelligence. Basic Books, New York (1993)
3. America, P., Hulshof, B., Odijk, E., Sijstermans, F.: Parallel computers for advanced information processing. IEEE Micro **10**(6), 12–15 (1990)
4. Wickens, C., Carswell, C.: Information processing. In: Salvendy, G. (ed.) Handbook of Human Factors and Ergonomics, 2nd edn, pp. 89–129. Wiley Inter science Publication, Wiley and Son, New York (1997)
5. Garcia-Molina, H., Ullman, J., Widom, J.: Database Systems: The Complete Book, 2nd edn. Pearson Prentice Hall, Upper Saddle River (2009)
6. Hoffman, R.R.: Whom (or what) do you (mis)trust?: historical reflections on the psychology and sociology of information technology. In: Proceedings of the Fourth Annual Symposium on Human Interaction with Complex Systems, pp. 28–36 (1998)
7. Wooldridge, M.: A knowledge-theoretic approach to distributed problem solving. In: ECAI 98. 13th European Conference on Artificial Intelligence. Proceedings, Brighton, UK, Wiley, pp. 308–312 (1998)
8. Dhar, V., Stein, R.: Intelligent Decision Support Methods: The Science of Knowledge Work. Prentice-Hall Inc., Upper Saddle River (1997)
9. Tweedale, J., Bollenbeck, F., Jain, L., Urlings, P.: Agent transportation layer adaptation system. In: Jain, L., Nguyen, N. (eds.) Knowledge Processing and Decision Making in Agent-Based Systems. Studies in Computational Intelligence, vol. 170, pp. 247–273. Springer, Berlin (2009)
10. Sioutis, C., Ichalkaranje, N.: Cognitive hybrid reasoning intelligent agent system. In: Khosla, R., Howlett, R., Jain, L. (eds.) Knowledge-Based Intelligent Information and Engineering Systems. Lecture Notes in Computer Science, vol. 3682, pp. 838–843. Springer, Berlin (2005)
11. Khazab, M., Tweedale, J., Jain, L.: Web-based multi-agent system architecture in a dynamic environment. Int. J. Knowl.-Based Intell. Eng. Syst. **14**(4), 217–227 (2010)
12. Ghosh, A., Tweedale, J.W., Nafalski, A.: Modified hybridized multi-agent oriented approach to analyze work-stress data providing feedback in real time. Procedia Comput. Sci. **22**, 1092–1101 (2013)
13. Callan, R.: Artificial Intelligence. Palgrave MacMillan, Hampshire (2003)
14. Minsky, M.: Society of Mind. Simon and Schuster, Pymble (1985)
15. McCarthy, J.: Programs with common sense. In: Symposium on Mechanization of Thought Processes, National Physical Laboratory, Teddington, England (1958)
16. Minsky, M.: Heuristic aspects of the artificial intelligence problem. Lincoln Laboratory Report, Federal Scientific and Technical Information, Department of Commerce, Washington, pp. 34–55 (1956)
17. Newell, A.: Production systems: models of control structure. In: Chase, W.G. (ed.) Visual and Information Processing, pp. 463–526. Academic Press, San Diego (1973)
18. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd edn. Prentice Hall, New Jersey (2003)
19. Ichalkaranje, N., Sioutis, C., Urlings, P., Tweedale, J., Lui, F., Jain, L.: Intelligent agents for airborne mission systems and land operations environments: simulation environment report. Research Report EIE-KES-IAAMS-2002-LJNI-07 [WP9], University of South Australia, Maswson Lakes, Adelaide (2003)
20. Tweedale, J., Jain, L.: Multi-lingual agents in a dynamic environment. In: Nakamatsu, K., Phillips-Wren, G., Jain, L., Howlett, R. (eds.) New Advances in Intelligent Decision Technologies. Studies in Computational Intelligence, vol. 199, pp. 193–202. Springer, Berlin (2009)
21. McCarthy, J., Abrahams, P.W., Edwards, D.J., Hart, T.P., Levin, M.I.: LISP 1.5 Programmer's Manual, 2nd edn. MIT Press, Boston (1962)
22. Feigenbaum, E.A., Buchanan, B., Lederberg, J.: On generality and problem solving: a case study using dendral program. In: Edinburgh, B., Scottland, M., Michie, D. (eds.) Machine Intelligence, vol. 6, pp. 165–190. Edinburgh University Press, Edinburgh (1971)
23. Lindsay, R.K., Buchanan, B.G., Feigenbaum, E.A., Lederberg, J.: Dendral: a case study of the first expert system for scientific hypothesis formation. Artif. Intell., Elsevier **61**, 209–261 (1993)

24. Dobson, W.J.: Arpanet information brochure. Technical Report 535, Defense Communications Agency, Washington, DC (1978)
25. Griffin, S.: Internet pioneers: Marc andreesen. http://www.ibiblio.org/pioneers/index.html, University of North Carolina, USA, pp. 1–3 (2008)
26. Bachman, C.W.: The origin of the integrated data store (IDS): the first direct-access DBMS. IEEE Ann. Hist. Comput. **31**(4), 42–54 (2009)
27. Zack, M.: The role of decision support systems in an indeterminate world. Decis. Support Syst. **43**(3), 1664–1674 (2007)
28. Hosack, B.: The effect of system feedback and decision context on value-based decision-making behavior. Decis. Support Syst. **43**(4), 1605–1614 (2007)
29. Bobillo, F., Delgado, M., Gomez-Romero, J.: Representation of context-dependant knowledge in ontologies: a model and an application. Expert Syst. Appl. **35**(4), 1899–1908 (2008)
30. Phillips-Wren, G.: Adaptive decision support for dynamic environments. In: Nakamatsu, K., Phillips-Wren, G., Jain, L., Howlett, R. (eds.) New Advances in Intelligent Decision Technologies. Studies in Computational Intelligence, vol. 199, pp. 235–243. Springer, Berlin (2009)
31. Hedlund, G.: A model of knowledge management and the n-form corporation. Strateg. Manag. J. **15**, 73–91 (1994)
32. Bresman, H., Birkinshaw, J., Nobel, R.: Knowledge transfer in international acquisitions. J. Int. Bus. Stud. **30**(3), 439–462 (1999)
33. Liu, D., Wu, I.: Collaborative relevance assessment for task-based knowledge support. Decis. Support Syst. **44**(2), 524–543 (2008)
34. Michalewicz, Z., Schmidt, M., Michalewicz, M., Chiriac, C.: Adaptive Business Intelligence. Springer, Berlin (2007)
35. Tweedale, J. W., Jain, L. C.: Embedded Automation in Human-Agent Environment. Springer, Berlin, Heidelberg (2011)
36. Phillips-Wren, G., Sharkey, P., Dy, S.: Mining lung cancer patient data to assess healthcare resource utilization. Expert Syst. Appl. **35**(4), 1611–1619 (2008)
37. Yen, J., Fan, X., Sun, S., Hanratty, T., Dumer, J.: Agents with shared mental models for enhancing team decision makings. Decis. Support Syst. **41**(3), 634–653 (2006)
38. Power, D.J.: A Brief History of Decision Support Systems. Quorum Books. Greenwood Publishing, Orlando (2007)
39. Keen, P.G.W., Morton, M.S.S.: Decision Support Systems: An Organizational Perspective. Addison-Wesley, Reading (1978)
40. Ray, P., Chattopadhyay, S.: Fuzzy awareness model for disaster situations. Intell. Decis. Technol. **3**(1), 75–82 (2009)
41. Beaumont, P., Chaib-draa, B.: Multiagent coordination techniques for complex environments: the case of a fleet of combat ships. IEEE Trans. Syst. Man Cybern., Part C (Appl. Rev.) **37**(3), 373–385 (2007)
42. Labrou, Y., Finin, T., Peng, Y.: Agent communication languages: the current landscape. IEEE Intell. Syst. Appl. **14**(2), 45–52 (1999)
43. Seely, S., Sharkey, K.: SOAP: Cross Platform Web Services Development Using XML. Prentice Hall PTR, Upper Saddle River (2001)
44. Martin, D.: OWL-S: semantic markup for web services. ISWC (http://www.daml.org/services/ last accessed 2014)
45. Hayes, J.P.: Computer Architecture and Organization, 2nd edn. McGraw-Hill Inc., New York (1988)
46. Wooldridge, M., Jennings, N.: Intelligent agents: theory and practice. Knowl. Eng. Rev. **10**(2), 115–152 (1995)
47. Chira, O., Chira, C., Roche, T., Tormey, D., Brennan, A.: An agent-based approach to knowledge management in distributed design. J. Intell. Manuf. **17**(6), 737–750 (2006)
48. Tweedale, J., Ichalkaranje, N., Sioutis, C., Jarvis, B., Consoli, A., Phillips-Wren, G.: Innovations in multi-agent systems. J. Netw. Comput. Appl. **30**(3), 1089–1115 (2006)
49. Parsons, S., Pettersson, O., Saffiotti, A., Wooldridge, M.: Robots with the best of intentions. In: Robots, Towards Intelligent Mobile (ed.) Towards Intelligent Mobile, pp. 329–338. IEEE, Bristol (1999)

50. Harnandez, M.: Database Design for Mere Mortals: A Hands-on Guide to Relational Database Systems, 2nd edn. Addison-Wesley, New York (2003)
51. Haider, K., Tweedale, J., Jain, L.: An intelligent decision support system using expert systems in a mas. In: Nakamatsu, K., Phillips-Wren, G., Jain, L., Howlett, R. (eds.) New Advances in Intelligent Decision Technologies. Studies in Computational Intelligence, vol. 199, pp. 213–222. Springer, Berlin (2009)
52. Haider, K., Tweedale, J., Jain, L.: An intelligent decision support system using expert systems in a MAS. In K. Natamatsu et al. (eds.) SCI 199, pp. 213–222. Springer, Berlin, Heidelberg (2009)
53. Savitch, W.: Absolute Java, 2nd edn. Addison-Wesley, New York (2005)
54. Jarvis, J.: JACK Intelligent Agents: Teams Manual 3.0. Agent Oriented Systems, Melbourne (2004)
55. Gibbins, N., Harris, S., Shadbolt, N.: Agent-based semantic web services. In: WWW '03: Proceedings of the 12th international conference on World Wide Web, New York, USA, pp. 710–717. ACM (2003)
56. Spooner, J.G.: Intel breaks 3 GHz speed barrier, p. 1. ZDNET Australia, San Francisco, California (2002)
57. Tweedale, J., Cutler, P.: Trust in nulti-agent systems. In: Gabrys, B., Howlett, R., Jain, L. (eds.) Knowledge-Based Intelligent Information and Engineering Systems, vol. 4252, pp. 479–485. Springer, Berlin, Heidelberg (2006)
58. Moore, G.E.: Fueling new solutions for the digital planet electronics. Intel Silicon Innov. (2005)

Chapter 8
Enhancing the Tactical Data Link Decision Support System

Christos Sioutis

Abstract Coordinating a complex array of deployed assets is a difficult task. Defence employs Tactical Data Link (TDL)s to maintain situation awareness and convey the intent of command and control information. Connectivity is achieved using digital radio communication systems that conform to present military TDL standard waveforms. Managing a TDL is inherently complex and typically requires operators (called TDL Network Managers (TDLNMs)) to actively manage these networks. These TDLNMs are trained military personnel and, due to the transient nature of their careers, most are posted within a few years and are often replaced with less experienced operators. Given this fluctuating level of experience, a Decision Support System (DSS) is being considered to assist the operators in maintaining a network. This chapter describes the conceptual design of an agent-based DSS, that could be used to support an operator, who is responsible for monitoring and optimizing a TDL. This includes tasks such as monitoring, troubleshooting and modifying various aspects of the TDL in order to maintain the required levels of connectivity and performance. One of the key tools used to support this role is a computer-based Network Management System (NMS). The proposed DSS could be developed using an Agent Architecture Framework (AAF) and integrated with the DSS based on Distributed Object Computing (DOC) technology. This approach minimizes the risk of integration between the NMS and the DSS, whilst allowing flexibility for future integration requirements. Cognitive Work Analysis (CWA) techniques could also be employed to capture and encode the expertise required for the DSS to operate effectively.

Keywords Agent · Cognitive work analysis · Decision support system · Distributed object computing · Network manager · Tactical data link · Software oriented architecture

C. Sioutis (✉)
Defence Science and Technology Organisation, Edinburgh, SA 5111, Australia
e-mail: Christos.Sioutis@dsto.defence.gov.au

J.W. Tweedale et al. (eds.), *Intelligent Decision Technology Support in Practice*, Smart Innovation, Systems and Technologies 42,
DOI 10.1007/978-3-319-21209-8_8

8.1 Introduction

A Tactical Data Link (TDL) network employs a web of interconnected radios that can be configured to reliably achieve long range digital radio communication between military platforms, as shown in Fig. 8.1. The hardware and software characteristics of TDL technology have evolved throughout the years, as have the message and protocol standards that define their information exchange. The most common TDL employed currently is Link 16 (L16), whose messaging protocol is defined by military standard MIL-STD-6016 [1]. Under normal conditions a L16 network will remain operational for extended periods of time, requiring only occasional monitoring to ensure continued satisfactory operation. However, problems do arise, and therefore, a TDL Network Manager (TDLNM) is tasked with supervising the system to determine the most appropriate form of intervention to resolve the problem (or minimize its impact).

The TDLNM requires a level of training and experience in order to gain an understanding of the current state of the network and to detect and diagnose problems. Additionally, the TDLNM must be able to determine the appropriate corrective action required to bring the system back into a future desired state. This is made more difficult by the fact that new operators are typically rotated into this role every few years. Consequently, continuous training and support is required and a known loss of potential expertise and/or lessons learnt is accepted. Reduced operator proficiency also implies a possible loss of capability whilst building up experience.

A Decision Support System (DSS) is an expert system that could be designed to capture elements of the rules, skills and knowledge of the TDLNM and used to support effective decision making. This chapter describes the conceptual design of a DSS that is intended to work alongside the TDLNM and provide assistance in the

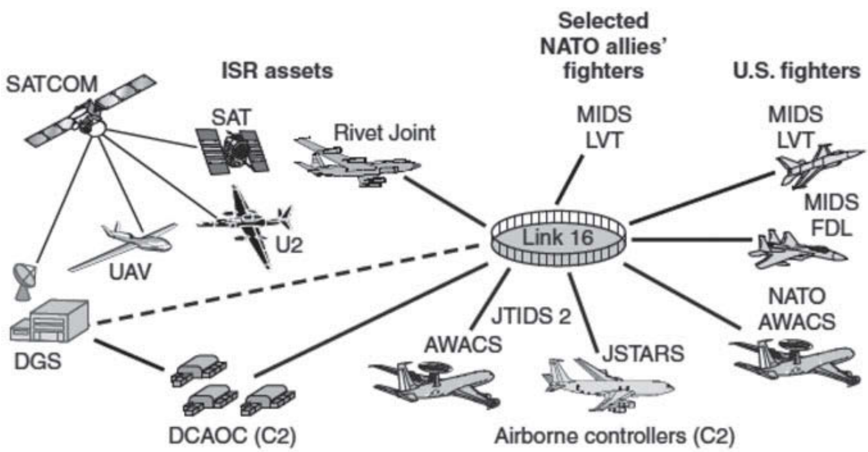


Fig. 8.1 MIDS in the future interoperable tactical communications architecture [2]

diagnosis of network issues as well as offering possible courses of action to resolve them.

The introduction of a DSS to support a Network Management System (NMS) is expected to reduce the burdens on the TDLNM and mitigate the effects of operators who may have reduced experience. These issues are discussed in this chapter using two themes. The first relates to modelling the TDLNM decision processes. This process will concentrate on enhancing the functional requirements of the DSS. The second is to design the DSS using a middleware based architecture. This approach uses software design patterns that realize agent-oriented behaviours.

To this end Sect. 8.2 describes the need to monitor and manage a TDL Network and Sect. 8.3 outlines the process used to model the TDLNM decision making framework. To assist the reader, Sect. 8.4 provides an introduction to the concepts associated with software agent design patterns, while Sect. 8.5 focuses on techniques that support Distributed Object Computing (DOC). Section 8.6 outlines the Agent Architecture Framework (AAF), and Sect. 8.7 describes the design of the proposed agent-based services prior to concluding remarks and a discussion on future effort.

8.2 Network Monitoring and Management

In order to employ a L16 communications network, platforms must be equipped with a specially designed radio terminal. These are called a Joint Tactical Information Distribution System (JTIDS) or a newer (smaller) variation called the Multifunctional Information Distribution System (MIDS). L16 operates in the UHF frequency spectrum which limits connectivity to direct Line of Sight (LOS). Consequently, the range of communication between two platforms using L16 is related to their respective range and altitude [3]. To overcome the LOS limitation, L16 employs a relay function that allows platforms to relay specified data to peers. This essentially achieves an interconnected mesh of nodes similar a Mobile Ad-hoc NET-work (MANET) network [4]. Figure 8.2 illustrates a L16 network in which the circles are surface platforms, the arches are aircraft and the squares are land platforms. L16 uses the communications principle of Time Division Multiple Access (TDMA) [5], which allows apparent simultaneous transfer of data by dividing transmissions into predetermined timeslots [6]. L16 information is transmitted in messages conforming to a fixed format, known as J-series which are grouped in general categories. For example:

- Network Management,
- Precise Participant and Identification,
- Surveillance, and
- Platform and System Status.

A L16 network needs to be designed, planned and initialized in order to achieve connectivity. The design process is operations agnostic and requires information such

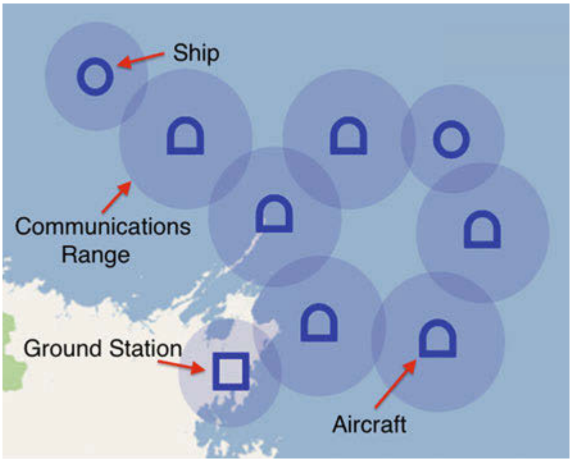


Fig. 8.2 Example link 16 network [3]

as the number and types of participants. It aims to split the total allocation of L16 timeslots into the different participant types as efficiently as possible whilst ensuring that each type has enough capacity to perform its mission requirements. The planning process in turn uses the network design and allocates participant types to specific platforms for a given operation. Platforms are issued with a unique L16 identifier and a specific configuration parameters to initialize their L16 data terminal [6].

L16 operates in the same Radio Frequency (RF) band as commercial aeronautical navigation aids. Therefore, in order to operate L16 there is a need to obtain a Frequency Clearance Agreement (FCA) [7]. The FCA stipulates a maximum allowable Time Slot Duty Factor (TSDF), which is a measure of the number of RF pulses transmitted within a given geographical area within a 12 second period. During operations the TDLNM is responsible for maintaining the health of the L16 network and also ensures that the TSDF being generated complies with the FCA. Additionally, L16 has the ability to alter the allocation of timeslots for platforms 'on the fly' using special management messages sent via a Network Management System (NMS) [8].

A NMS is employed to assist the TDLNM in the monitoring of a L16 network; both from a capacity and health perspective. Capacity is considered in terms of assigned operational roles, system capabilities, and the spectrum constraints imposed in terms of TSDF. The network health is influenced by the ability of individual platforms to maintain timing and position synchronization with the network. Platforms conducting these roles must adhere to strict standards of behaviour that govern network entry, time management and position determination. The specific role of the TDLNM includes (but is not limited to) the following activities [8]:

- Ensuring that the participants are operating as planned,
- Ensuring that connectivity is maintained throughout the network,
- Supporting the entry of new participants into the network, and
- Updating the network design to support operational needs.

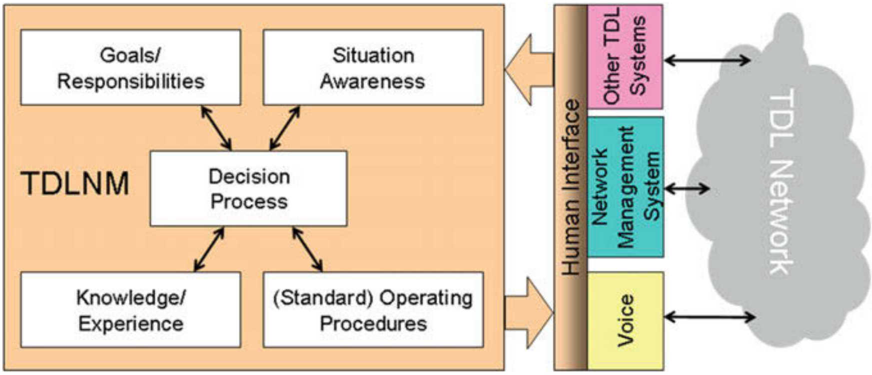


Fig. 8.3 TDLNM general decision process [9]

Figure 8.3 illustrates the basic premise of the TDLNM decision process. In simplified terms, a TDL network operates dynamically in the real world (illustrated as a cloud on the right) where the TDLNM is responsible for maintaining a required level of performance by managing the network. Whilst managing the network the TDLNM continually undertakes a decision making process. In favor of simplicity, the TDL is observed and manipulated through voice interfaces over other infrastructure (using radio, telephone or any other medium). The TDLNM decision process is influenced by four elements [9]:

Situation Awareness: This captures the TDLNM understanding of the TDL network. It can be modelled through a series of consistent beliefs. When new beliefs are asserted they are checked for consistency and, if a conflict arises, further investigation must be applied to determine which beliefs (old or new) should be removed to maintain consistency.

Goals/Responsibilities: This captures the TDLNM role (purpose of being). This is modelled through a hierarchy of goals with their priorities. The model can include many goals for many different situations. However, only a subset of these goals is active at any time, depending on the actual situation at hand.

Knowledge/Experience: This captures the TDLNM background knowledge and experience. This includes knowledge such as how a TDL network works and what happens when certain actions are taken.

Standard Operating Procedures: This captures the procedures and actions that the TDLNM takes. The simplest procedures to model are the Standard Operating Procedures (SOP) which are well defined and represent the standard practice. However, there may be procedures that are not specifically documented which could also be incorporated in the model.

8.3 Modelling the TDLNM Decision Process

The decision process modelling described in this section forms a subset of a large research area within the human factors research discipline that is known as Cognitive Work Analysis (CWA) [10]. The main focus of CWA is the analysis, design and evaluation of human-machine systems. In other words, understanding how humans use machines to achieve specific work objectives. It specifically recognizes that in order to achieve an acceptable level of performance in a given application one must consider the abilities and limitations of individual workers interacting with their supporting machines. The Computer Systems Engineering (CSE) [11] research area further extends upon this concept by designing human/machine systems based on similar principles.

CSE uses various models for representing the states of knowledge and information processes of human reasoning. One such model is known as the "decision ladder". The decision ladder models the human decision making process through a set of generic operations and standardized key nodes, or states of knowledge, about the environment. Figure 8.4 illustrates an example generic decision ladder from the TDLNM perspective. The decision process begins from the bottom left then works its way up the ladder and down again on the right; it is described in more detail below:

Observation: This is the default activity, where the TDLNM regularly updates their understanding of the state of the TDL network through inspection of the user interface. This is not necessarily a constant operation as the TDLNM may have other duties to perform.

Identification: In this activity, the TDLNM uses their understanding of the network state to make a diagnosis of the likely causes of a problem. This may include the further investigation of specific system parameters.

Decision: This activity makes a decision on what changes are necessary to bring the system back into the desired state based on the diagnosis of the problem.

Planning/Tasking: This activity implements the decided response by setting tasks and planning a set of actions to act upon the environment.

While these activities can be arranged in an easily understood sequence, it is important to recognize that experienced operators make intuitive leaps [12]. The expertise level is important when considering the design of the DSS, which should aim to raise the ability of less experienced operators. The ladder can be segmented into three levels of expertise that takes into account people making intuitive leaps:

- The skill (lowest) level represents very fast, automated sensory-motor performance and it is illustrated in the ladder via the heuristic shortcut links in the middle.
- The rule (medium) level represents the use of rules and/or procedures that have been predefined, derived empirically using experience, or communicated by others. It traverses the bottom half of the ladder.
- The knowledge (highest) level represents behaviours during less familiar situations when someone is faced with an environment where there are no rules or skills

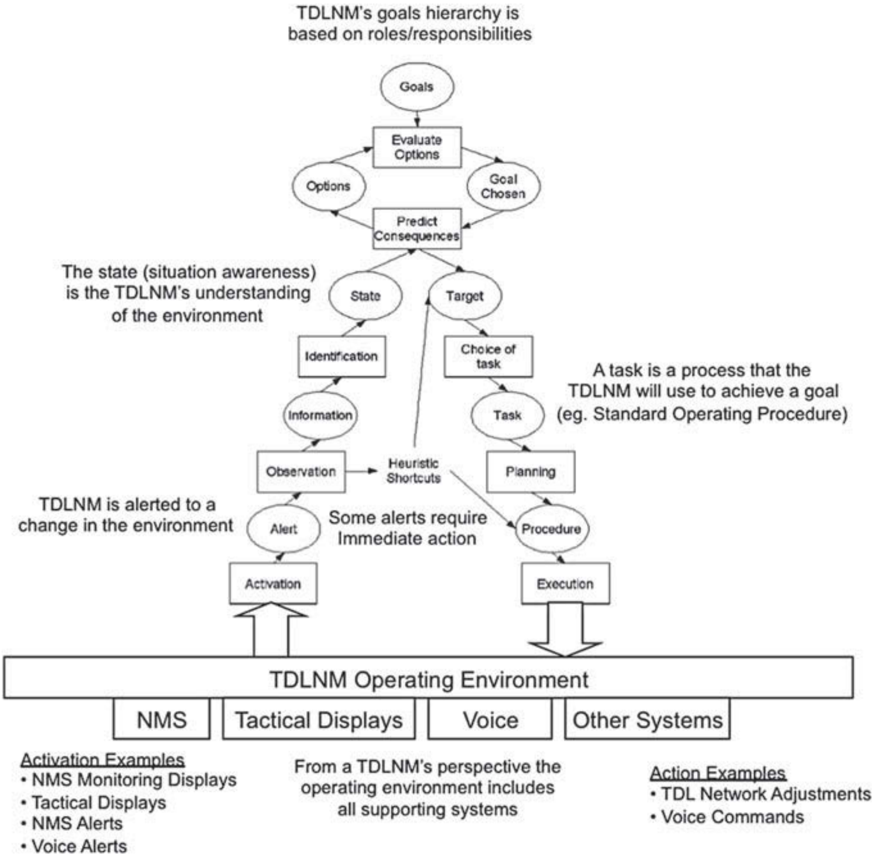


Fig. 8.4 Decision ladder from the TDLNMs perspective [9]

available. In such cases a more detailed analysis of the environment is required with respect to the goals required to be achieved.

In applying the decision ladder to L16 network management, it becomes apparent that the decision ladder does not appear to adequately capture the diverse set of activities that is required to be represented. Whether the operator is monitoring, troubleshooting or attempting to confirm that a problem is being resolved can greatly impact the underlying concepts being represented by a node in the model [13]. This has led to modifying the basic decision ladder in order to represent the L16 monitoring process, as shown in Fig. 8.5.

The new design captures a process that is executed in a circular fashion as new information arrives and is processed. This information is categorized as one of: general updated information; alerts pertaining to a known issue; or feedback. This feedback is typically focused on fixing previously detected issues. A feedback activity is also included that is required to monitor the TDL network and ascertain if the

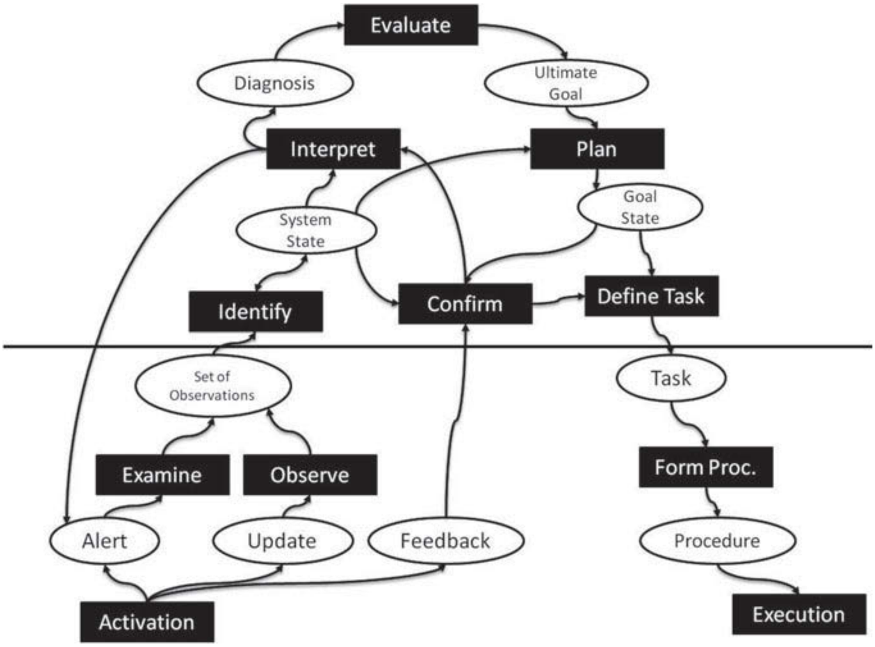


Fig. 8.5 Customized decision ladder link specific to TDLNM tasks

implemented actions have the desired effect. If this is not the case, then the TDLNM may need to decide whether to select an alternate action, or form a new diagnosis.

The system state/knowledge is persistent between iterations, which aims to constantly refine, rather than merely generate, the system state knowledge. Depending on the process to be supported, and the needs of the inference mechanisms, this knowledge may include the history of the system and previous suppositions made to resolve past problems in addition to the current state of the system. The ladder was also customized to separate the various forms of the observe activities. These categories have been segregated into: examine, observe and confirm. The feedback will depend on how it was derived and returned as an: alert; update; or feedback prompt. The decision ladder is therefore executed from one of the following categories:

- The update of information without problems in the system;
- The arrival of an alert to a problem (as per the template's original design);
- The update of information with a problem that is previously undetected; and
- The update of feedback pertaining to a known problem in the system.

Lastly, Table 8.1 provides a generic template against which scenarios for TDLNM activities can be defined. In the example provided, the TDLNM is alerted to the fact that a deployed unit is not able to synchronize their radio and thereby join the TDL network. It can be seen that to determine the probable cause of the problem, the TDLNM must have an indication of where the connected platforms are with respect

Table 8.1 Generic task template

Alert	Observe	Diagnose	Task	Procedure	Feedback
Fail to synchronize	Current network time setting	Inaccurate time setting on platform	Adjust time on platform of nearest network	Use voice communication to instruct the correct time	Confirmed when platform appears on the network
	Distance from platform to known platforms in the network	Platform out of network range	Arrange platforms to ensure ranges are appropriate	Use voice communications to command platforms	Monitor distances to ensure reduction
		Incorrect settings	Reload terminal with correct settings	Use voice communications to instruct platform	Confirm when platform appears on network

to a particular location and the current time on the network. The ability to determine the locations would be partially met with a tactical display that shows the locations of connected platforms and the ability to determine the distance to a given location. This would allow DSS to preclude certain diagnoses and offer suggested courses of action to the TDLNM (along with explanations of the decision made).

8.4 Software Agent Design Patterns

The general concept of an agent involves an entity that is situated within an environment. The environment generates sensations, triggering reasoning in the agent and causing it to take actions. These actions in turn generate new sensations, hence forming a sense-reason-act loop. This is reinforced by Wooldridge's widely referenced definition: 'An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design requirements' [14]. Wooldridge continued to define a number of properties that an agent-based system should exhibit:

Autonomy: operating without the direct intervention of humans;
Social Ability: interacting with other agents;
Reactivity: perceiving and responding to changes in the environment; and
Proactiveness: initiating actions to achieve long term goals.

Agent-oriented development extends upon traditional software development techniques with concepts that are specifically suited to expressing autonomous behaviour with complex interactions. A very good agent-oriented development methodology is described by Padgham [15].

A number of very capable agent frameworks have been developed to aid in implementing agent-based systems; for examples see [16–18]. These frameworks primarily provide new constructs and algorithms specifically suited to aid a developer in designing an agent-based system. Each framework however, introduces its own concept of what an agent is and how its behaviours are implemented. Each framework therefore brings its own advantages/disadvantages and is suited to different applications. This means that developers must choose the appropriate framework for their application.

After exploring a number of agent frameworks, one can deduce that they have conceptual similarities, albeit with different implementations. This hints at the existence of generic design patterns to describe the mechanisms of agent behaviour in an implementation independent way. An understanding of these patterns allows a developer to focus on the design of the required agent behaviours whilst being able to switch between frameworks as needed. Researchers have already discovered the possible advantages of using patterns for designing agents. Weiss [19] describes a hierarchical set of patterns for agents. For example, the Agent as a Delegate pattern begins to describe how a user delegates tasks to an agent. An attempt is also made to classify agent patterns based on their intended purpose by Oluyomi et al. [20] and a two-dimensional classification scheme is proposed with the intent that it is problem driven and logically categorizes agent-oriented problems. Although not specifically mentioning patterns, related work by Jayatilleke et al. [21] has attempted to describe agent behaviour as a model. Key behavioural elements utilized by the popular Beliefs, Desires and Intent (BDI) [14] reasoning model are defined as:

Goals: a desired state of the world, as understood by the agent.
Events: notifications of a certain state of the internal or external environment of the agent.
Triggers: an event or goal which invokes a plan.
Plans: respond to predefined triggers by sequentially executing a set of steps.
Steps: primitive actions or operations that the agent performs.
Beliefs: storing the agents view of its internal state and external environment. These can trigger goal completion/failure events as well as influence plan selection through context filtering.
Agent: a collection of the above elements, designed to exhibit a required behaviour.

8.5 Distributed Object Computing

Modern computer software is inherently complex, comprised of many interactions and combining to provide a structured cohesive whole. In support of managing (and even reducing) this ever increasing complexity, new approaches in software design have been developed. The reference architecture of the software plays a central role in specifying generic function layers enclosing patterned behavioural elements.

Distributed Object Computing (DOC) frameworks, on the other hand, provide partially-built component implementations focused on a particular problem domain [22]. The frameworks provide not only cohesive re-usable components but also re-usable behaviour [23]. Modular software components can be defined through pattern-orientated framework mechanisms. This allows for a higher order of re-use and enhanced application developer productivity and reliability. It can also provide guidance to developers adopting a solid and reliable approach to the software being produced.

DOC middleware connects application components as shown in Fig. 8.6a. Middleware transparently manages the connectivity environment between distributed application components. Components can concurrently interoperate across different operating systems, hardware platforms and even language environments. Middleware technologies have evolved significantly over the past decade in support of the move to standardization. One example is the CORBA [24] standard published by the OMG. Coupled with this standardization is the capability to simplify the integration of disparate distributed application components and systems within a heterogeneous system-of-systems context [22].

DOC middleware is typically comprised of different layers responsible for providing a given functionality. Lower layers provide generic services and capabilities which are utilized by higher layers that are in-turn increasingly more domain or application specific. Applications can be developed to integrate with a particular layer that they support and interoperate with other applications operating on other layers. As a result, most of the work in such systems happens in the underlying layers, whilst the business logic itself is exposed as a service. This is the essence of the SOA approach [22]. Figure 8.6b illustrates how SOA differs to DOC systems. The dotted horizontal lines in the middleware indicate logical connections that can be routed in different ways. The difference in sizes of the services in Fig. 8.6 alludes to the fact that services built on higher layers need less development because they leverage on the functionality of the layers underneath them.

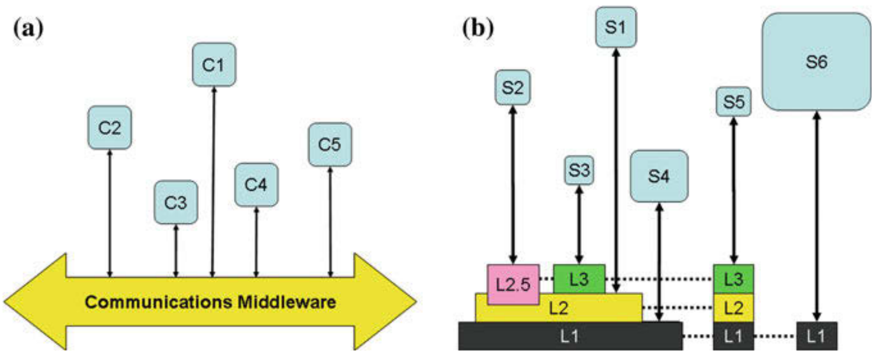


Fig. 8.6 Connecting software components with middleware [22]. **a** Distributed Computing Architecture, **b** Service Oriented Architecture

8.6 Agent Architecture Framework (AAF)

The DOC methodology provides standardized ways for different applications to communicate and also provides some services (Naming or Trading), which can be very useful when utilized in conjunction with agents. For example, in [25] middleware is utilized for inter-agent communication as well as access to additional databases. CORBA-based agents operate the same as traditional CORBA components. When plugged into a large DOC system they instantly have access to other agents, information and services. Legacy systems can also be wrapped with CORBA interfaces and become available to the agents [22].

Extending the above concept, the CORBA Component Model (CCM) is a component modelling architecture built on top of CORBA middleware. Researchers have already identified the CCM as a possible approach for merging agents with DOC concepts. Melo et al. [26] describes a CCM-based agent as a software component that: exposes a defined interface; has receptacles for external plan components; and utilizes sinks/sources for asynchronous messaging. Similarly, Otte et al. [27] describes an agent architecture called MACRO built on CCM that introduces additional algorithms for agent planning and tasking in the application of sensor networks. The main advantage of this application is noted as the abstraction of the details of the underlying communication and system configuration from the agents. Specific limitations identified in this application include overheads imposed by the middleware (due to the limited processing capacity in the embedded systems employed for the sensor network) and having limited control data routed around the network [22].

The AAF will implement generic agent behaviours and algorithms through well understood software design patterns. There are three aims of the AAF shown in Fig. 8.7a. First, the AAF will link to specific base support libraries used to implement the required algorithms. Second, it will merge the library application interfaces with the workflow and conventions of larger SOA. Third, it will be built using

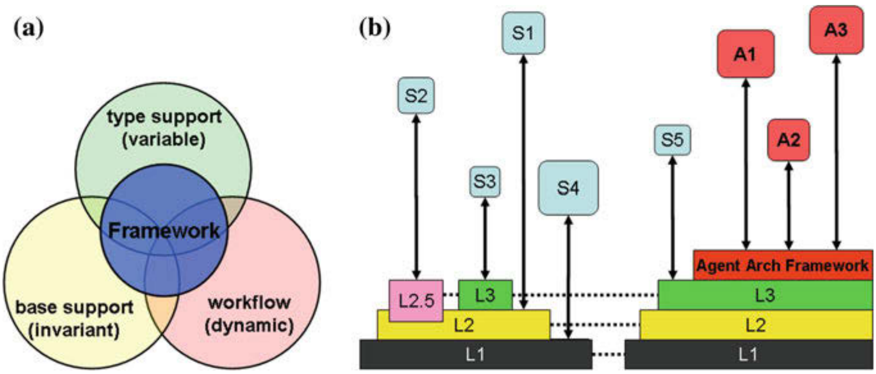


Fig. 8.7 Agent architecture framework (AAF). **a** AAF building blocks, **b** AAF layer in a SOA system

templates in order to capture the algorithmic logic while allowing it to work against any given type. The intent is for the AAF to be used to implement agents utilizing the same middleware architecture as larger SOA systems. This way agents and services will be able to exchange data and operate seamlessly. This concept is illustrated in Fig. 8.7b [22].

8.7 Design of the Agent-Based Services

Agent-based services will be developed to perform the various tasks defined using the generic task template described in the previous section. The design of the agents will be based on the methodology described in [15] and implemented using the AAF. The AAF will in-turn be integrated with a NMS through communications middleware, as shown in Fig. 8.8.

The initial concept design includes four service modules, each having multiple agents, with responsibilities that are mapped to the customized decision ladder described in the previous section. These service modules are:

Monitoring: service will be responsible for developing a Situation Awareness that is complementary to the TDLNM and encompases tasks identified for observing the network state.
Diagnosing: service will be responsible for determining the cause of NMS alerts and/or problems relayed back to the TDLNM via voice.
Alerting: service will define ways to alert the TDLNM based on a diagnosis made. These will be additional to any alerts generated by the NMS itself.
Actioning: service will be responsible for taking actions on behalf of the TDLNM and monitoring the network for the desired effect.

The astute reader may have noticed a horizontal line shown in Fig. 8.5. This line indicates a hint at the initial level of automation expected to be provided by the DSS. Specifically, the Monitoring, Alerting and Actioning services will be developed first in order to provide initial support for the decision ladder elements indicated below the line. The Diagnosing service will need to make use of the functions provided by the other three services. As development progresses, the horizontal line indicated in the figure will shift up to encompass additional decision ladder elements, like

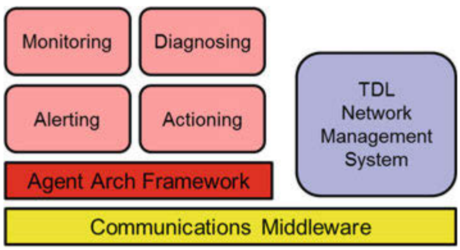


Fig. 8.8 Initial set of agent services

interpreting the network state and confirming the effects of actions taken upon the network. It is intended that the final evaluation and decision (the top of the decision ladder) will always be deferred to the TDLNM . The DSS will however offer support to the TDLNM in diagnosing problems, offering possible courses of action and monitoring the network to ensure that actions produce the desired effects.

8.8 Conclusion

This chapter describes the conceptual design of an agent-based DSS to assist in monitoring and management of Link 16 tactical data links. The DSS is intended to work alongside the TDL network manager and provide assistance in the identification and diagnosis of network issues, followed by the presentation of possible courses of action to resolve them. The agents will be developed using an architectural framework that is based on DOC middleware technologies and generic agent behaviours/algorithms based on well understood software design patterns.

A customized decision ladder is used to more closely map the tasks related to the specific application of TDL network management. Additionally, four modules have been identified for initial implementation (Monitoring, Diagnosing, Alerting and Actioning) each providing a set of agents to assist the TDLNM as required.

Acknowledgments The author would like to acknowledge the contribution of William Scott (performed through collaboration with the University of South Australia) in support of the research described in this chapter. In particular, the customized decision ladder and generic task template.

References

1. DoD: Tactical data link (TDL) 16 message standard. Military Standard - Revision E, Defense Information Systems Agency, United States Department of Defense, Arlington County, Virginia (2012)
2. Hura, M., McLeod, G., Schneider, J., Gonzales, D., Norton, D.M., Jacobs, J., O'Connell, K.M., Little, W., Mesic, R., Jamison, L: Interoperability: A Continuing Challenge in Coalition Air Operations. Number MR-1235-AF. Rand Corporation, p. 272. Santa MOnica, California (2000)
3. Tobin, B., Baillie, C., Sioutis, C.: Achieving high fidelity tactical data link simulations through modelling of radio transmissions. In: Simulation Technology, Adelaide Convention Centre, South Australia (2014)
4. Basagni, S., Conti, M., Giordano, S., Stojmenovic, I.: Mobile Ad hoc Networking (2004)
5. Freeman, R.L.: Fundamentals of Telecommunications, 2nd edn. Wiley, Institute of Electrical and Electronics Engineers, Hoboken, N.J. (2005)
6. HQ TRADOC: Introduction to tactical digital information link J and quick reference guide. Quick Reference Guide ADA404334, Army training and doctrine command, Fortmontroe, VA (2000)
7. Gore, A.: JTIDS/MIDS frequency clearance agreement FCA management. In: International Data Link Symposium, SyntheSys, UK (2010)
8. Johnson, M.: MIDS link 16 network design and management. In: JTIDS/MIDS Link 16 Network Design and Management, NICTA, Canberra, ACT, SyntheSys Australia (2010)

9. Ding, Y., Sioutis, C., Baillie, C., Tobin, B., Scott, W.: Human machine interface analysis through cognitive modelling. Australian International Aerospace Congress (2015)
10. Vincente, K.J.: CongnitiveWork Analysis: Towards Safe, Productive and Healthy Computer-Based Work. Lawrence Erlbaum Associatesn, Mahwah, NJ, USA (1999)
11. Rasmussen, J., Pejtersen, A.M., Goodstein, L.P.: Cognitive Systems Engineering. Wiley (1994)
12. Hutton, R.J.B., Klein, G.: Expert decision making. Syst. Eng. **2**(1), 32–45 (1999)
13. Bodker, S.: Through the Interface: A Human Activity Approach to User Interfaces. CRC Press, Boca Raton, FL (1990)
14. Wooldridge, M.: Reasoning about rational agents. In: Intelligent Robotics and Autonomous Agents. MIT Press, Cambridge, Massachusetts and London, England (June 2000)
15. Padgham, L., Winikoff, M.: Developing Intelligent Agent Systems: A Practical Guide. Wiley, West Sussex, England (2004)
16. AOS: JACK Intelligent Agents: Users Manual. Agent-Oriented Software, Carlton, Melbourne. Release 4.1 edn. (2004)
17. Bellifemine, F., Caire, G., Trucco, T., Rimassa, G.: Jade Programmer's Guide (June 2007)
18. Laird, J.E., Congdon, C.B.: The soar user's manual. Technical report, University of Michigan, MI, United States (2004)
19. Weiss, M.: Patterns for motivating an agent-based approach. In: Jeusfeld, M.A., Pastor, O. (eds.) Conceptual Modeling for Novel Application Domains. Lecture Notes in Computer Science, vol. 2814, pp. 229–240. Springer, Berlin Heidelberg (2003)
20. Oluyomi, A., Karunasekera, S., Sterling, L.: Description templates for agent-oriented patterns. J. Syst. Softw. **81**(1), 20–36 (2008)
21. Jayatilleke, G.B., Padgham, L., Winikoff, M.: A model driven component-based development framework for agents. Comput. Syst. Sci. Eng. **20**(4), 273–282 (2005)
22. Sioutis, C., Dominish, D.: Developing intelligent agents with distributed computing middleware. In: Watada, J., Phillips-Wren, G., Jain, L., Howlett, R. (eds.) Intelligent Decision Technologies. Smart Innovation. Volume 10, Systems and Technologies, pp. 863–872. Springer, Berlin (2011)
23. Schmidt, D.C., Stal M., Rohnert H., Buschmann F.: Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects. Wiley (2000)
24. OMG: Common object request broker architecture corba specification. Technical Report Version 3.1, Object Management Group, Needham, MA, USA (2008)
25. Cheng Tao, C.T., Guan Zailin, G.Z., Liu Liming, L.L., Wu Bo, W.B., Yang Shuzi, Y.S.: A corba-based multi-agent system integration framework. In: Ninth IEEE International Conference on Engineering of Complex Computer Systems, pp. 191–198. (2004)
26. Melo, F., Choren, R., Cerqueira, R., de Lucena, C.J.P., Blois, M.: Deploying agents with the corba component model. In Emmerich, W., Wolf, A.L., (eds.) Component Deployment. Volume 3083 of Lecture Notes in Computer Science, pp. 234–247. Springer (2004)
27. Otte, W., Kinnebrew, J., Schmidt, D., Biswas, G.: A flexible infrastructure for distributed deployment in adaptive sensor webs. In: Aerospace Conference, pp. 1–12. IEEE Computer Society Press, USA (2009)

Chapter 9
AC^{3}M: The Agent Coordination and Cooperation Cognitive Model

Angela Consoli

Abstract Coordination is the management of the interdependencies of activities, whereas cooperation is where two parties form a voluntary relationship to share resources for a goal. With the introduction of Intelligent Multi-Agent System (I-MAS), Coordination and Cooperation (CO$-$O^2) is required to ensure agents act coherently. This ensures that agent tasks and actions are completed in a timely manner. Current CO$-$O^2 models are ubiquitous and rigid. I-MAS have matured allowing agents to exhibit personification and cognitive processes. This now renders tradition agent coordination and cooperation models useless. The Agent Coordination and Cooperation Cognitive Model (AC^{3}M), is a first generation of hybrid CO$-$O^2 models which exploits an agent's cognitive and mental processes to establish a link between CO$-$O^2. The methodology and design of AC^{3}M focuses on how the organisational structure, environment and mental modelling of an intelligent agent can affect CO$-$O^2. The implementation of AC^{3}M features three frameworks, which enhances the link between CO$-$O^2. Each of AC^{3}M's framework also assist in establishing the importance of perception, situation awareness and decision-making in CO$-$O^2 methodologies. The application of AC^{3}M is particularly successful in team automation within dynamic environments. Teaming in an unknown or dynamic environment heavily relies on how effectively CO$-$O^2 is established.

9.1 Introduction

This chapter outlines an innovative agent Coordination and Cooperation (CO–O^2) model that enhances team automation within dynamic and unknown environments. Current CO–O^2 techniques have been designed and developed in isolation. Each methodology is heavily reliant on rules, coordination methodologies focus purely on roles, and cooperation focuses on sharing resources. This has resulted in CO–O^2 systems being unable to be integrated into new-age agent based systems that are dominated by Intelligent Multi-Agent Systems (I-MASs).

A. Consoli (✉)
Defence Science and Technology Organisation, Edinburgh 5111, Australia
e-mail: Angela.Consoli@dsto.defence.gov.au

J.W. Tweedale et al. (eds.), *Intelligent Decision Technology Support in Practice*, Smart Innovation, Systems and Technologies 42,
DOI 10.1007/978-3-319-21209-8_9

I-MASs contain multiple Intelligent Agents (IAs) and must be able to support four main characteristics of IAs: autonomy, reactivity, pro-activity and social abilities. I-MAS must also accommodate IAs being assigned task, goals, actions and capabilities. Designers of I-MASs need to exploit the characteristics of IAs to effectively model governance and promote cooperative relationships. Achieving this outcome depends on how an IA's mental modelling, environment and organisational structure are constructed. The primary role of AC^3M was to procure the link between Coordination and Cooperation $CO–O^2$. This was achieved by concentrating on developing $CO–O^2$ methodologies that focus on IAs mental model, I-MASs environment and organisational structure. An important aspect of AC^3M's design was to encompass traditional properties of $CO–O^2$. Coordination has been perceived as the managing of interdependencies between activities, whereas cooperation is the voluntary relationship between two parties.

One application of AC^3M is to enhance team automation. IAs have been used extensively to help simulate automation of human-machine systems, particularly in hostile, dynamic and unknown environments. Automation is seen as techniques and mechanisms employed to achieve automatic operations. There are three areas of importance in team automation:

1. Successfully coordinating activities, tasks and goals of each entity,
2. Cooperation of shared resources, and
3. Effective decision-making.

The aim of AC^3M is to emphasise $CO–O^2$ models to improve an agent's autonomy and decision-making capabilities. AC^3M's $CO–O^2$ models ensure that a decision to act can be made quickly and effectively.

This chapter is structured as follows. Section 9.2 investigates the importance of intelligence, organisational theory, agent systems, coordination and cooperation and how they are interlinked. Section 9.3 describes and defines the methodology used for AC^3M. Section 9.4 details the implementation of AC^3M and focuses on the multi-agent environment, organisational structure and agent mental model. Section 9.5 defines the frameworks of AC^3M and how they are related to the environment, organisational structure and the agents. Section 9.6 illustrates how AC^3M links agent coordination and cooperation. Section 9.7 shows how AC^3M can be applied to enhancing team automation and finally Sect. 9.8 discusses the future work of AC^3M.

9.2 Background and Motivation

The design, development and implementation of AC^3M required the use of five principle themes:

1. Intelligence, Distributed Systems and Organisational Theory;
2. Agent Theory and Multi-Agent Systems;
3. Cognitive Systems;

4. Agent Coordination; and
5. Agent Cooperation.

Theme one is quintessential and considers the motivation and founding theories of AC^{3}M. This theme is discussed in Sect. 9.2.1. Themes two and three are combined and discussed in Sect. 9.2.2. Themes four and five are discussed in Sects. 9.2.4 and 9.2.5 respectively.

9.2.1 Intelligence, Organisational Theory and Distributed Artificial Intelligence (DAI)

The term *intelligence* has been used across many disciplines. These include, but are not limited to: human [1], artificial [2], emotional [3, 4], cognitive [3], social [3] and animal [5]. Since the term intelligence is used and applied across a diverse range of domains, there is no one universally accepted definition.

One particular area of interest is in the study of human intelligence. Researchers have devoted considerable time investigating, simulating and modelling human intelligence by artificial means. The principle focus was to disseminate the process of individual human thought, cognition and decision-making capabilities [6, 7]. This research coined the term AI. As the research into AI matured, the focus shifted from problem-solving to analysing individuals and then organisation of humans.

Organisational theory is the study of groups of entities within an organisation. This theory has existed before the birth of AI, but has only theorized how to design and implement an organisation without focusing on individual behaviours. AI was able to bridge organisational theory with human intelligence by using three main areas of organisational theory:

1. Establishment, or the creation of an organisation,
2. Functions and dynamics of an organisation, and
3. How the first two themes impact on the culture of an organisation [8].

As the research into AI and organisational theory progressed, a methodology was developed to simulate and analyse how entities, both individually and collectively, can affect an organisation structure. This methodology is known as Computational Organizational Theory (COT).

COT was developed by Carley and Gasser [9] using mathematical and computational models to study and evaluate human organisations. The development of COT created a mature understanding into how to organise multiple information processing [9, 10]. The application of COT yields three results:

1. Social phenomena and individual behaviours;
2. Advances of cognitive behaviours and their effects on the deliberation and decision-making capabilities within organisation structures; and
3. Bounded rationality of organisational entities [11].

To ensure these results can be attained, COT required contributions from agency theory and cognitive science [11]. The outcome of applying these contributions to COT was the formalisation of Distributed Artificial Intelligence (DAI).

Distributed techniques concentrate on how intelligent entities act within their environment [12, 13]. DAI focusses on the distributed problem-solving techniques by assessing an entity's ability to gain knowledge, plan, perceive and learn [12, 14, 15]. Bond and Gasser [16] believe that DAI requires contributions from a number of disciplines to assist DAI in achieving its prescribes objectives. They include:

1. Cognitive science,
2. Distributed systems,
3. Human-computer interaction,
4. Basic AI research, and
5. Engineering of AI systems

It is important to note that these 'contributions' have topics of expertise, which assist in establishing and defining DAI. Table 9.1 catalogues these subjects against their contribution counterparts.

Bond and Gasser [16] explains the importance and relevance of applying DAI. DAI facilitates coordination of actions of entities within an organisation. Additionally, DAI can demonstrate tools to represent and reason their knowledge, actions and plans to other entities [16]. In other words, DAI encourages coordination, cooperation, decision-making and problem-solving capabilities in an organisational setting [17]. One area of particular importance and focus is how cognitive sciences and IA theory can affect CO–O^2 techniques and decision-making capabilities within an organisation.

Table 9.1 Contributing domains in distributed artificial intelligence

Contribution	Subjects
Cognitive science	Mental modeling social cognition
Distributed systems	Reasoning
	Rationality
	Architectural and language specifications
Human-computer interaction	Task allocation
	GUI design and development
	Speech recognition
Basic AI research	Knowledge representation
	Epistemology
	Reasoning and problem solving
Engineering of AI systems	Cooperative expert systems
	Distributed sensing systems
	Robotics

9.2.2 Cognitive Science and Agency Theory

The definition and discussion of agency theory, agents and multi-agent systems have been a topic of interest for many years and are summarised in [18–21]. An important aspect of *intelligent* agent theory is an agent's ability to react, perceive, decide and act within their environment [21]. This is the cognitive abilities of agents.

Situational awareness, more specifically the Observe-Orient-Decide-Act (OODA) model and the C^{4}ISR-inspired command and control structures were the inspirations of the cognitive modeling of AC^{3}M. In order to implement the command and control within AC^{3}M, an agent's mental model needed to be defined. AC^{3}M's agent model consisted of two main components:

1. Their internal state, and
2. Cognitive and reasoning state.

Consoli [22] discusses that the agents within AC^{3}M are designed from first principles using features discussed by Bergenti and Huhns [23]. Table 9.2 summaries these features.

When designing AC^{3}M's multi-agent environment, [22] uses the Gaia Methodology's 'checklist' on defining a multi-agent environment. At a basic level, the multi-agent environment of AC^{3}M possessed the following assumptions:

1. Not be implicitly assumed;
2. Must contain characteristics that are identifiable; and
3. Models should be configured and massaged to meet an applications specific purpose.

Bergenti and Huhns [23] illustrate the mental states of an agent as "its state …represented in terms of its …beliefs and what it is currently pursuing (intentions)" [23, p. 22]. Their methodology on designing an agent model was incorporated within AC^{3}M. Their design shows that an agent model needs to consider the following:

1. Allowing an agent to explicitly represent their goal;
2. Allowing an agent to perceive and represent their knowledge of the environment, including all other agents;

Table 9.2 Object and agent-oriented specific design features and considerations

Feature	Object-oriented	Agent-oriented
State	Attributes and relations	Mental attitude
Communication	Meta-object protocols	Agent communication language
Delegation of responsibility	Task delegation	Task and goal delegation
Interactions between entities	Interfaces	Capability descriptors
Interaction with environment	Events	Update of belief-set

3. Not allowing any manipulation of the state of another agent direct, only by means of communication; and
4. Not containing any public attributes.

The agent's internal, or mental state within AC^{3}M was based on the Beliefs, Desires and Intent (BDI) methodology defined by Bratman [24]. This paradigm enabled agents to possess *folk psychology*, or human-centric behaviours, allowing practical reasoning to be implemented [24, 25]. Internal states of an agent is defined in terms of beliefs, goals and plans models [24–26], where AC^{3}M labeled these as belief, desire and intentional models [22].

The design of AC^{3}M also championed concepts explained by Omicini [27]. He showed that multi-agent design methodology needed to follow "an understanding which societies are required, which social laws they need, how social rules should be designed and where they should be embedded" [27]. Omicini [27] also concluded that an agent's social behaviours also needed to be implemented at this stage since they are dependent on the design of organisational infrastructure [28–30]. CO–O^2 methodologies and techniques can enforce social laws and substantiate interactions between agents in an organisation. The effectiveness of CO–O^2 is how successfully each can be designed and how they are implemented in union.

9.2.3 Coordination and Cooperation Methodologies

Consoli [31] illustrated that AC^{3}M established techniques that allowed agents to be coordinated and form relationships with other agents. The primary roles of CO–O^2 in AC^{3}M was to assist in the completions of an agent's task and to ensure that coherency was maintained [22, 31]. AC^{3}M's CO–O^2 methodologies were not only focused on resource and task management of each agent, but also an agent's situation awareness [32].

AC^{3}M's CO–O^2 methodologies allowed enhanced decision-making processes and action. This means information needs to be evaluated and stored. Thus, AC^{3}M consisted of three intelligent components: coordination, cooperation methodology and a knowledge representation framework.

9.2.4 Coordination Methodology of AC^{3}M

The coordination within AC^{3}M existed as two interconnected systems: team coordination and entity-based coordination. The design and development of each followed the basic principles of coordination protocols [3]. AC^{3}M possessed coordinables, coordination media and coordination laws. In terms of AC^{3}M, coordination laws were uniform throughout the entire system and the coordinables were the actual agents [33–35]. The intention of AC^{3}M was to illustrate how coordination leads to

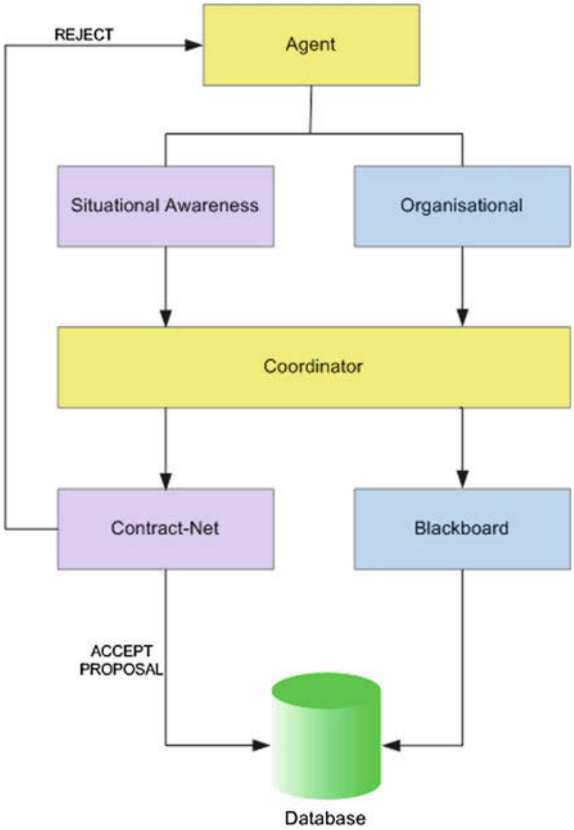


Fig. 9.1 Overall coordination of AC^{3}M

cooperative relationships by exploiting an agent's cognitive properties. AC^{3}M was not intended to illustrate how to coordinate resources of an agents based solely on their role or position within an organisation [31, 36, 37]. This design decision led to the coordination methodology of AC^{3}M consisting of two types of coordination:

1. Organisational, and
2. Entity-Based [22].

Figure 9.1 shows the relationship between an agent's internal state and the overall coordination system of AC^{3}M.

9.2.4.1 Organisational Coordination

The Organisational Coordination model illustrates ex-ante and explicit coordination characteristics [22]. This was to ensure that the collective behaviours of agent's were coordinated for the completion of system objectives. 'Behaviours', in this case, are not those behaviours defined in an agent's internal structure, instead they are an agent's expertise given in their initial coordination when structuring the multi-agent system.

The 'coordination media' in Organisational Coordination was designed upon one of the earliest models of coordination: the Blackboard System [38]. It was seen as a monitoring system which allows basic problem-solving capabilities. Nii [38] demonstrated the blackboard architecture as consisting of three main components: Knowledge Source, Blackboard Data Structure and Control. The Knowledge Source contain attributes required to solve the problem. Blackboard Data Structure is a data repository that stores the knowledge sources which are reproduced when necessary and Control is how the Knowledge Source responds to the change.

The blackboard system described by Nii [38] was modified when used in AC^{3}M. The main reason was the agents were intelligent and cognitive. The shared data space was not for all to view, which reduced any coherency issue regarding self-interested agents [22, 32]. The coordinator only coordinated at an organisational level, which provided AC^{3}M with the resource and task management for all agents. Additionally, agents periodically communicated with the system to provide updates to the data space. This allowed instant coordination of agents.

9.2.4.2 Entity-Based Coordination

Entity-based coordination followed an ex-post and implicit coordination characteristics [22]. Entity-based coordination allowed an agent to *self-coordinate* to ensure they were able to complete their task. As a result, Entity-Based Coordination was designed to encourage inter-agent relationships [22].

Entity-Based Coordination employed a Contract-Net-inspired technique. Smith [39] shows the Contract-Net protocol as a collection of knowledge-source nodes which are decentralised, where the agents exhibited a manager-contractor relationship. The knowledge-source nodes do not contain enough information to allow full coordination as they are mostly computational, not communicative. Coordination was achieved through resource allocation and focus. Resource allocation is the coordination which occurred at the Organisation-Based coordination level and were located at the knowledge-source computational nodes. Coordination via focus was the selection of local tasks by agents [39].

Agents needed to negotiate and bid to allocate local tasks [22]. Local task coordination required an agent to report on their situation awareness activities and this included:

1. Perception of environmental information, or beliefs;
2. Re-evaluating task status and completions, or desires; and
3. Self-delegation of internal tasks to complete their goal, or intention [22].

9.2.5 Cooperation Methodology of AC^{3}M

Cooperation within AC^{3}M was solely at an entity-to-entity level and was a result of two processes:

1. Instantiation of interactions within the Entity-Based Coordination technique, and
2. Ratification of a plan between two agents [22, 31].

The cooperation methodology was completely focused on an agent's internal state and mental model. AC^{3}M's cooperation was concerned with how the mental models, that were created from situation awareness and Entity-Based Coordination affected the relationship between agents [17, 22]. The quintessential component of this methodology was the relationship between coordination, desires and intentions [31]. Figure 9.2 shows the overview of cooperation within AC^{3}M.

This methodology closely followed the *Cooperative Problem Solving Methodology* and seven laws of cooperation [40, 41]. The methodology used was a modified version of Wooldridge and Jennings [41] four stages of cooperation. These design decisions included:

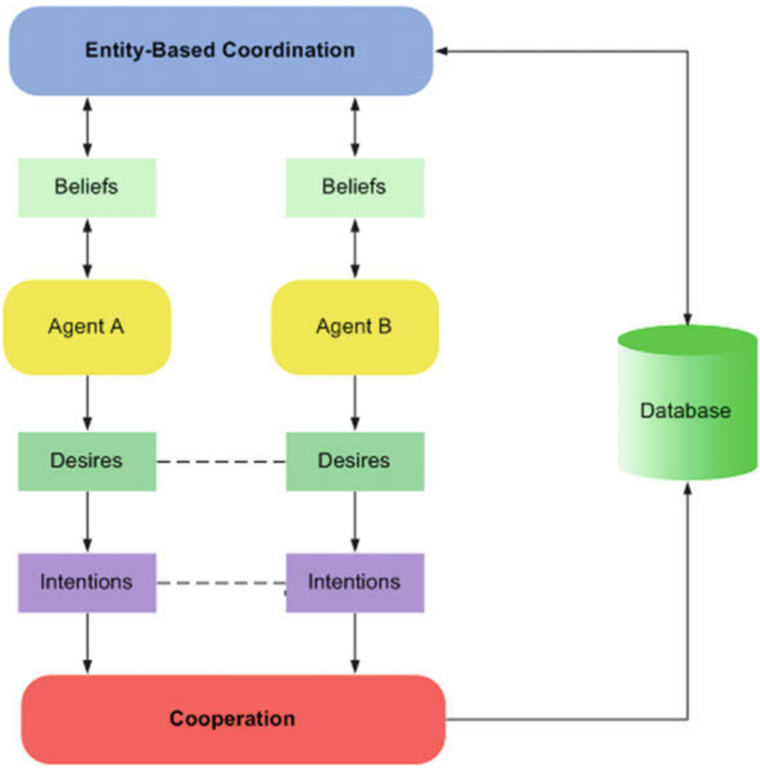


Fig. 9.2 Cooperation methodology within AC^{3}M

1. The cooperation methodology focused primarily on an agent's situation awareness rather than their goal formation.
2. Cooperation was viewed as a system-wide mechanism to assist in achieving an agent's goal. To accommodate this change, *Plan Formation* was promoted to the second, not third stage.
3. Planning is used to negotiate terms of cooperation.
4. Cooperation was deemed to have failed if planning failed. The result was any team formation failed, or did not proceed.

The highlight of AC^{3}M's cooperation methodology was incorporating re-planning if the agreement of a plan could not be reached [22]. The failure of plan resulted from organisational-based coordination. Re-planning would continue until the beliefs of an agent resulted in a set of desires and intentions that were equal for each agent. When re-planning was deemed successful, team formation could proceed [22].

9.3 The Methodology of AC^{3}M

The merge of social sciences with information technology is not novel. Ramesh et al. [42] illustrate that computing has been used in areas of social science that includes and not limited to informational, organisational and societal studies. To successfully utilise the computing and social science components of the study, techniques needed to be investigated that allowed cognitive and organisational sciences to be effective within artificial intelligence and software engineering [22]. AC^{3}M incorporates three areas of science:

1. Artificial Intelligence,
2. Social and cognitive science, and
3. Agency theory [22].

The use of these domains of science contributed to AC^{3}M design methodology that support the mental modelling and processes required for the agents to act. Additionally, these domains of science aided in the design and implementation of the multi-agent organisational structure, environment, and the CO–O^2 methodologies.

9.3.1 Design of AC^{3}M's Multi-Agent Organisation and Environment

The design of AC^{3}M's organisational structure was inspired by the Gaia Methodology developed by Zambonelli et al. [43]. To commence designing a multi-agent organisation, they addressed three principles. These were used to form the concept of *organisational design patterns* and include organisational:

1. Structure,
2. Environment, and
3. Governance.

Zambonelli et al. [43] elaborated on his organisational design patterns by describing what each principle needs to contribute when designing agent-oriented systems [22]. The Zambonelli et al. [43] organisational design patterns were used in AC^{3}M and are described in Table 9.3.

When designing AC^{3}M's multi-agent environment, [22] uses the Gaia Methodology's 'checklist' on defining a multi-agent environment. At a basic level, the multi-agent environment of AC^{3}M possessed the following assumptions:

1. Not be implicitly assumed,
2. Must contain characteristics that are identifiable, and
3. Models should be configured and massaged to meet an applications specific purpose.

Zambonelli et al. [44] I-MAS environmental design checklist matured and provides insight into the importance of evaluating resources, perception and other environmental components. Consoli [22] uses this checklist to consider how other subsystems and agents can affect the design of AC^{3}M's environment. A summary into the design decisions are as follows:

1. **Evaluating environmental resources**: Need to consider constraints that exist when accessing, using and manipulating resources. Such constraints may be new to the dynamics of the environment, hence these dynamics need to be identified and evaluated based on AC^{3}M.
2. **Evaluating perception and representation of the environment**: This is in relation to the agent and relates to the how the agent is modeled.

Table 9.3 Organisational design patterns for AC^{3}M

Organisational design component	Description and rationale
Organisational structure	Automate, monitor and support functions
	Design of a role-based, or hierarchical system
	Consideration of sub-structures
	Impact of structure on entities in a dynamic environment
Organisational environment	Efficiency, robustness and openness
	Optimisation of dynamic environments
	Facilitation of online reorganisation for dynamic environments
Organisational governance	Committing rules on a priori structure
	Facilitation of rules for dynamic reorganisation
	Affect of roles and hierarchy

3. **Evaluating components required in the environment**: This relates to the interaction protocols within the environment [43, 44].

In terms of the organisational structure, Zambonelli et al. [43] recommended three *do not's* when designing and developing an I-MAS structure. Of the three, AC^{3}M rejected only one of the recommendations which states that a multi-agent system should not automatically exhibit a human-centric approach. The recommendation continued by stating that a human-centric approach may be detrimental to a multi-agent system design because of:

1. Structural issues,
2. Human organisations that may not fit well within an agent-oriented system, and
3. Presence of a software system may inevitably change the actual human organisation, especially in the case of organisational rules [43, 44].

Consoli [22] discounts this recommendation because an organisational structure is not static. An organisational structure must be dynamic and must be able to change from external stimuli. If processes are refined to assist in *ironing out* design issues in a human-centric organisational structure, the interaction protocols and rules bound to the structure will mirror such changes [22]. The organisational structure of AC^{3}M follows the view that rules and interactions change with the maturity of the structure.

Another reason for the rejection of this '*do-not*' is the actual centrepiece of AC^{3}M's design. Zambonelli et al. [43] believes that the use of a human-centric approach to organisational structure will require further automation by means of software, especially in coordination and cooperation, which may be detrimental to a multi-agent organisation. The implementation of a software-inspired system for AC^{3}M can be viewed as an organisational re-structure of current human-centric organisation patterns. Automating CO–O^2 processes in dynamic environments can reduce CO–O^{2}costs, especially when organisations possess a large number of entities [22].

9.4 Implementation of AC^{3}M

The Agent Coordination and Cooperation Cognitive Model (AC^{3}M) [31], utilises the traditional concepts of coordination and cooperation to form a link between CO–O^2 [22, 32]. AC^{3}M consists of four main components:

1. Agents,
2. Environment,
3. Frameworks, and
4. Knowledge database known as the `ReAcT` Server.

The modeling of the agents is based on the BDI methodology. Using the BDI methodology allows the agents to be intelligent; thus possessing cognitive and reasoning characteristics, as well as decision-making capabilities. They are able to react within their environment and decide on outcomes to execute plans and goals. It is their intelligent characteristics that enables the interactions amongst agents, which is critical for the successful completion of actions, and for their survival within their environment.

9.4.1 Multi-Agent Environment and Organisational Structure

AC^{3}M is a Java real-time system which can be either a standalone application or a external component to an existing agent system. AC^{3}M does not inherit the organisational structure, environment or governance and work processes of an existing system, instead inherits the capabilities to form its own defined organisation [22]. Figure 9.3 shows the overall structure of AC^{3}M.

The organisational structure is responsible for the instantiation and management of the agents. This is where their role, beliefs, desires and intentions, as well as their mental model are formed and updated. The organisational structure is based on a hierarchical system, where there are a set of subordinate agents and a lead agent. Each subordinate agent has an expertise, which is defined within a scenario database.

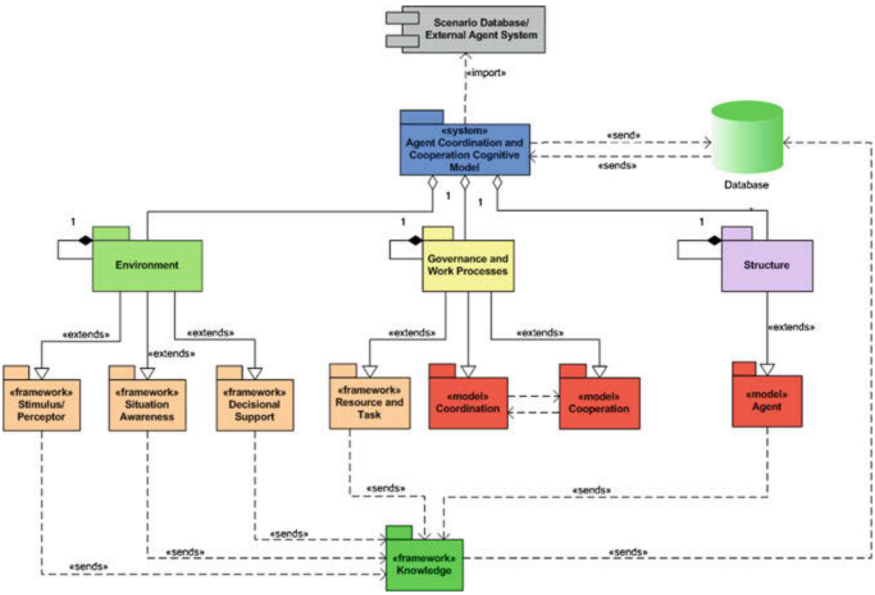


Fig. 9.3 Overall structure of AC^{3}M [22]

Figure 9.4 shows the relationship between the organisational structure and the set of agents in AC^{3}M.

AC^{3}M's environment has three main processes which enable goals, tasks and interactions between agents:

1. Handling of External Stimulus Events,
2. Handling of physical attributes of the environment, and
3. Perception of events and situation awareness of agents [22].

Consoli [22] illustrates the environment in AC^{3}M as a 3-tuple as shown in Eq. 9.1.

$$env \rightarrow \{ag^{+}, SE, PA\} \tag{9.1}$$

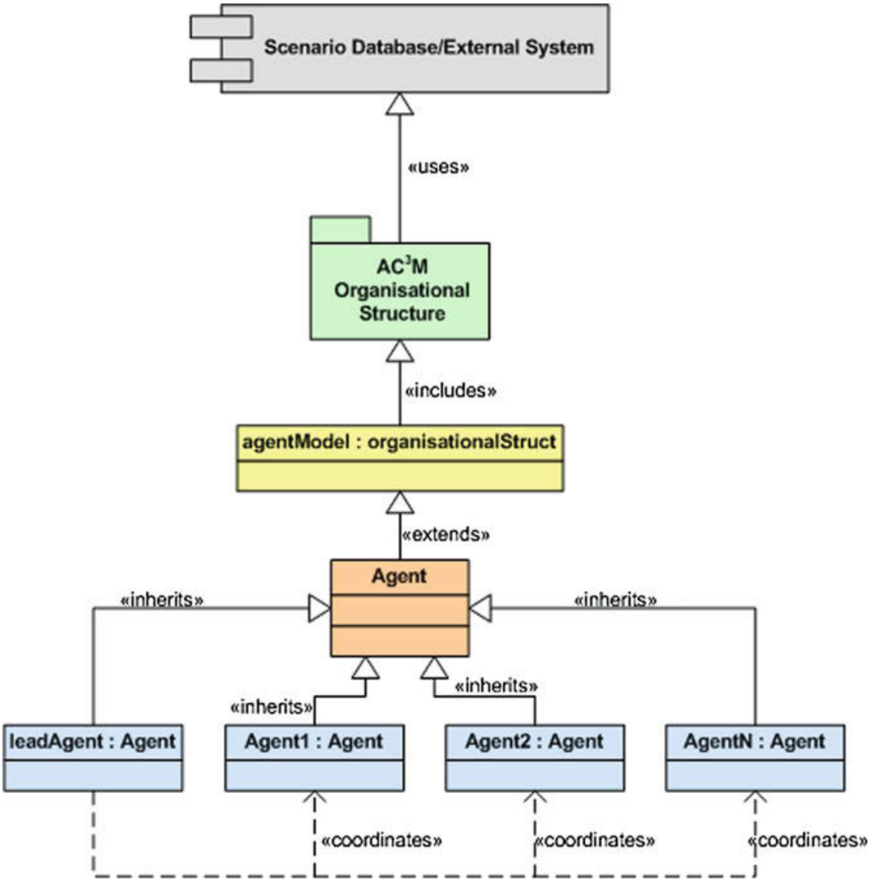


Fig. 9.4 AC^{3}M organisational structure [22]

where,

ag^+ is a set of agents within the environment,
SE is a set of Stimulus Events that are dynamic and enter the environment at any time, and
PA is a set of Physical Attributes of the environment.

The environment in AC^{3}M is dynamic and ensures an agent is able to detect External Stimulus Events. These events are entered by either a *scenarioTimedEvents* file or from the external system. AC^{3}M has two classes that ensure:

1. Retrieval of the External Stimulus Event and importation of the events into a container component. Once this event has been fired, it is sent to another interface for further analysis.
2. Monitoring of the current state of the Stimulus Event.

The environment also contains and maintains physical attributes. This includes the location of the environment, number of agents, obstacles that may affect the physical attributes of an agent as well as other user-defined environmental attributes. These attributes are contained either in an external agent system or the scenario database. As with the handling of external stimulus events, the physical attributes are also controlled by classes that ensure:

1. Retrieval and sending of initial physical attributes to the environment,
2. Periodical monitoring for any new attributes sent by an external system, and
3. Monitoring the current state of each physical attribute.

9.4.2 Agent Model

AC^{3}M models agents based on three groups of characteristics:

1. **Organisational**: roles and expertise;
2. **Mental and Cognitive**: beliefs, desires and intentions; and
3. **Physical**: an agent's physical attributes.

AC^{3}M contains an n-tuple of agents, Ag, where the first element of the set is a `leadAgent`, LAg and the remaining elements are k number of agents, ag [22]. The number of agents is defined within an external system or scenario file. Even though both types of agents share the same class definitions, their purpose differs within AC^{3}M. At the top-level of the system, an agent is seen as an object that contains cognitive and physical attributes. Consoli [22] shows that AC^{3}M defines each agent as an 8-tuple of attributes defined as follows.

1. Set of beliefs, $B = \{b_k, b_j \ldots b_n\}$; where b_k is the initial belief;
2. Set of desires, $D = \{d_j \ldots d_n\}$;
3. Set of intentions, $I = \{i_j \ldots i_n\}$;
4. Set of physical attributes, $PA\{j \ldots n\}$;
5. Expertise, e;
6. Role, r;
7. Resource allocation percentage, *RA*; and
8. Task allocation, *TA*.

These agents exist within the environment, thus they are a subset of the environment. Consoli [22] shows the agent definition within AC^{3}M as shown in Eq. 9.2.

$$Env \to \{LAg, ag_k \ldots ag_n\} = \begin{Bmatrix} B,D,I,PA,r, & \text{for } LAg \\ B,D,I,PA,e,r,RA,TA, & \text{for all other agents} \end{Bmatrix} \quad (9.2)$$

As seen in the definition of the agents, the model contains two different types of agents: `leadAgent` and `agents`. The `leadAgent` agent within AC^{3}M is solely responsible for the organisation coordination and governance of all other agents. The `leadAgent` must ensure the interactions between agents are not of a self-interest nature and are of benefit to the system. Through this governance, the `leadAgent` coordinates an agent's resource and task allocation based on the initial resource allocation, role and expertise. This agent can fail a cooperative relationship between agents if an agent's resource and task allocation exceeds their threshold. In addition, this agent can dynamically change an agent's resource and task allocation at any given time. The `leadAgent` will have the same physical characteristics as all other agents.

Unlike the `leadAgent`, all other agents within AC^{3}M are responsible for the completion of individual and system goals. Through the coordination from the `leadAgent`, `agents` will use their reasoning to perceive, decide and act on external stimuli from the environment [22]. To enhance their ability to perceive and act, agents each have an expertise and role that are assigned at their instantiation. Their expertise is static; this is defined in an external system or scenario file and cannot be amended. An agent's role is dynamic and is changed via organisational coordination. Like the `leadAgent`, each agent has a set of dynamic and physical attributes.

Each agent has dynamic beliefs, desires and intentions, which can change based on the following two events. The first is the perception of an external stimuli and subsequent situational assessment and the second is coordination. The information from these events are processed and analysed using three frameworks:

1. Stimulus/Perceptor,
2. Situational Assessment, and
3. Decisional Support.

All three frameworks promote the link between coordination and cooperation, the interaction between agents and the goals and tasks associated with the action required for the event. Each framework of AC^{3}M is discussed in Sect. 9.5.

9.5 The Frameworks of AC^{3}M

As shown in Fig. 9.3, the frameworks are an extension of AC^{3}M's environment and allows agents to interact, analyse and change their environment. More importantly, these frameworks allow agents to coordinate and cooperate in order to complete goals and tasks against events that may arise. AC^{3}M required an framework where agents can perceive an event, as well as initiate situation awareness and uses a Stimulus/Perceptor Framework.

9.5.1 Stimulus/Perceptor Framework

This framework consists of two main components:

1. Intelligence, surveillance and reconnaissance, and
2. Command and control.

The first occurs within the `ISR` interface, which allows an agent to retrieve and process information regarding the observations from the environment. Agents will form their beliefs based on the perceptions and stimuli that is processed from this component. The second component, `commandControl` interface initialises the coordination of an agent via a `Coordinative Event`. This is where the `leadAgent` will coordinate the agent perceiving the event to complete tasks, which will eventuate in the completion of a goal for that specific event.

The Stimulus/Perceptor framework is the first step in illustrating the importance of an agent's cognitive and mental modeling. This framework provides the foundations for the decisions and actions required for a Stimulus Event and how they can affect the entire system [22]. The environment is the key to an agent's dynamic behaviour since it provides the Stimulus Event and enables an agent to commence perception. By using their mental model, agents are themselves stimulated to perceive, hence requiring their ability to observe to update their beliefs. Updating of their beliefs requires further coordination, since their organisational requirements may have changed as a result of this event. This process is vital, as it is the foundations of their situation awareness and the command and control of AC^{3}M [22].

9.5.2 Situational Assessment Framework

The Situational Assessment Framework is used if the `Stimulus Event` requires further analysis, which occurs when the *expertise* characteristic of the event does not match the expertise of the agent perceiving the event. This framework is where the desires and intentions of the event are determined and the command and control components of the system exist.

There are three main interfaces in this framework. The first, `Coordination`, allows the `leadAgent` to fire the `Coordinative Event` constructed in the Stimulus/Perceptor framework. The `Coordination` interface is where organisational coordination occurs as an agent's coordination relies on their resource, task and constraint allocation. The coordination in this case is based on the Stimulus Event, thus changing an agent's role and beliefs.

When a `Coordinative Event` has been fired, the `dynamicPlanReplan` interface creates a plan which outlines the expertise required to complete actions against the event. This leads to the third and final interface, `orientation`. This interface determines the desires and intentions for the agent to complete actions against the event [22]. To assist in determining a set of actions, agents must be able to use their decision-making capabilities.

9.5.3 Decisional Support Framework

Even though some of the decision-making processes occurs within the Stimulus/Perceptor and Situational Assessment frameworks, the decision-making within the Decisional Support is entity-based. This framework gives rise to three main events:

1. Cooperation amongst agents,
2. *Coordinative Cooperation*, and
3. Actions required for the completion of tasks and plans associated with the `Stimulus Event`.

Coordinative Cooperation is a result from two situations: entity-based coordination between agents and the negotiation between the `leadAgent`. Cooperation and Coordinative Cooperation are not related or equal; they are two separate theories and have been discussed further in Consoli et al. [31, 36].

The Decisional Support Framework consists of three main interfaces. The first, `cooperation` establishes which agent is able to assist in completing the goal of the event, based on their expertise. The second interface, `Coordinative Cooperation` is where an agent will cooperate with others to form entity-based coordination. In addition, this is where the course of action will be instantiated. The third interface, `decisions` is where the decision on how to act will be constructed.

9.6 Linking Coordination and Cooperation in AC^{3}M

Consoli et al. [31] discusses how AC^{3}M establishes the link between CO–O^2 through Coordinative Cooperation. Consoli [22] provides a mature explanation on the link between CO–O^2 by highlighting how AC^{3}M boosts the governance of agent's behaviours and their interrelationship on a system and entity-level. AC^{3}M aligns an agent's belief, desire and intentions together with their ability to observe, orient, decide and act to link cooperation and coordination [17]. Consoli et al. [17] also highlights how this alignment was linked to coordination and cooperation, strengthening the concept of the link between CO–O^2. They discussed how the perception and observation of each agent leads to the update of their beliefs and instantiates the organisational coordination of an agent. Consoli et al. [17] continue by stating that orientation updates each agent's desires and intentions. At this stage, each agent engage in decision-making. The first step of this decision-making process is to determine whether the act of cooperation is required. If it is deemed to be necessary, the two agents cooperating will use their decisions to form intentions, which leads to a set of actions [17, 22].

To illustrate how this is implemented in AC^{3}M, Fig. 9.5 shows how each framework uses the BDI, OODA and CO–O^2 to establish and maintain the link between CO–O^2. Here Consoli [22] shows how this is actually implemented in AC^{3}M. Initially the `coordinativeCooperation` Interface formulates the entity-based coordination. By agreeing to cooperate, an agent must now coordinate their beliefs to determine tasks and actions that are required. Furthermore, this coordination, through cooperation, must be negotiated with the `leadAgent` to ensure organisational needs of the system can still be met. When negotiated, the goals and tasks associated with the plan are retrieved. The actions are decided upon the beliefs, desires and intentions of the agents. In the case of the agent perceiving the event, they use their beliefs from their organisational coordination and in terms of the expert agent, their beliefs are represented from their expertise and role. Once a course of tasks and actions have been completed, cooperation is deemed to have been successful, and the relationship ceases.

Prior to the completion of all actions against the event, the `decisions` interface determines which actions are required to satisfy the plan, goals and tasks associated with the plan and the beliefs of the agent. This is the final step of an agent's decision-making process. The agent will construct a final course of action, which will be executed within the Agent Model.

When this course of action is released to the Agent Model, `decisions` interface will also send an update to the environment. The result of updating the environment may result in a new Stimulus Event as the environment has changed due to the actions being performed. The `decisions` interface will also request to update an agent's desires and intentions, as they are no longer required for the Stimulus Event. In terms of the beliefs, an update of the beliefset of the agent occurs once all actions have been performed. This is to ensure that the coordinative cooperation is valid and will not result in failure of completing tasks.

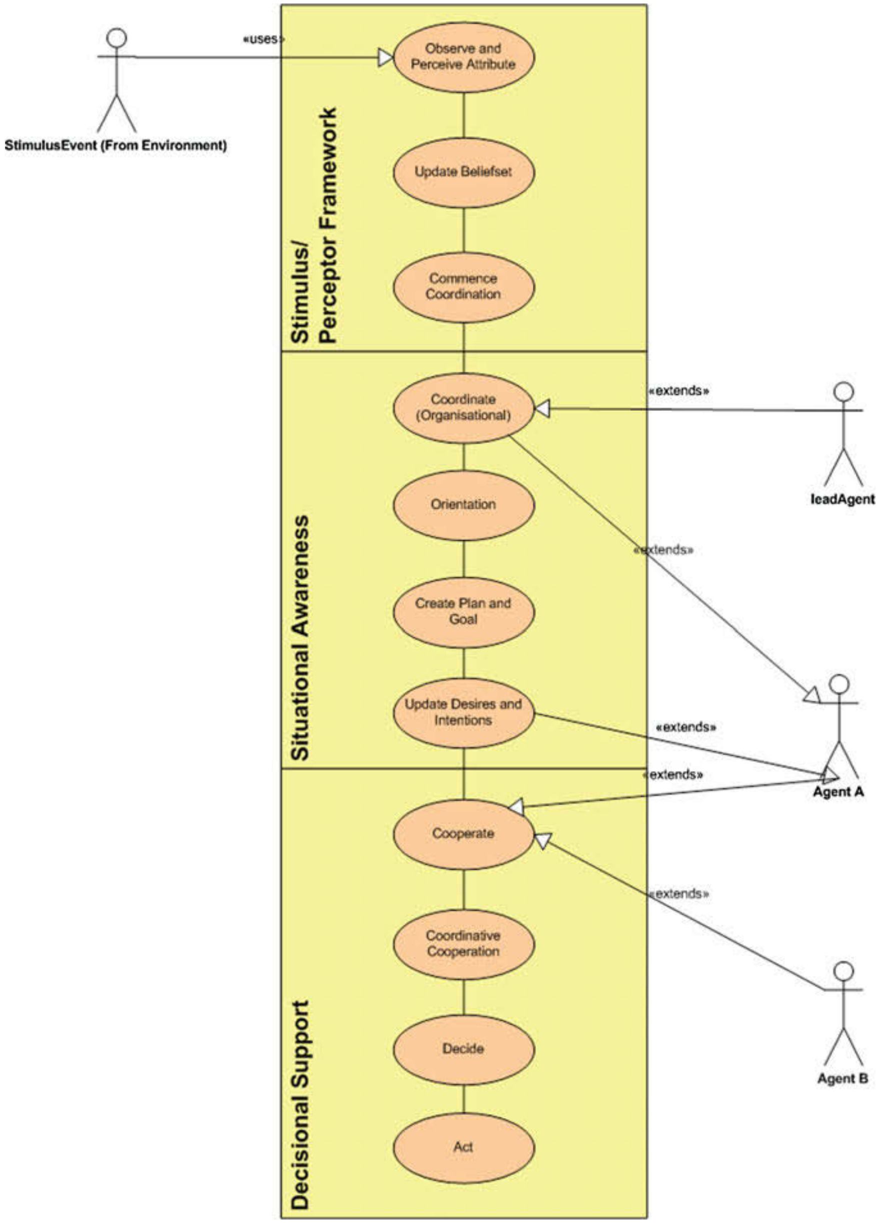


Fig. 9.5 Relationship between all frameworks within AC^{3}M

9.7 Application of AC^{3}M

The current state of automation still involves a large participation of multiple human operators. The role of the human in an automation system is to provide the coordination, control, psychological, cognitive, situation awareness, social and decision-making properties [45, 46]. The increase of complexity of automated systems has placed pressure on the human, resulting in several performance issues. Cuevas et al. [47] show these issues to include:

1. Automation surprises,
2. Poor situation awareness, and
3. Loss of manual control.

The need for enhanced automation becomes apparent when there is an excessive workload for a human operator [48]. If multiple human operators are under considerable amounts of stress, this can result in the degradation of the system and the creation of critical errors [48, 49]. Wickens [50] discusses three main functions where enhanced automation is needed:

1. Where human knowledge is limited;
2. Where a human performs a task that may be a detriment to the system. In addition, this may create a high workload to the human; and
3. Where humans have shown to possess physical limitations.

The issues listed are exasperated when a team of automated systems are positioned in dynamic environments. This is due to the sophistication of these systems, which leaves multiple operators responsible for the socio-cognitive processes [45, 47, 51]. The result is the need of enhancing automation of these teams by reducing the reliance on the human.

9.7.1 The Need for Enhanced Team Automation in Unmanned Vehicle Systems

An area of interest for the development of enhanced automation and autonomous technologies are within unmanned vehicles systems. This area establishes a desire in the reduction, or removal of the human from the immediate control of these vehicles [52]. There are three main reasons for this reduction or removal:

1. Reduction of cost in production and maintenance;
2. Operational use in hostile and dynamic environmental; and
3. Reducing human loss within these environments.

The implementation of current automated and autonomous technologies have seen the increase of unmanned vehicles in dynamic environments, which requires an increase of performance in the completion of complex tasks. Therefore, merely implementing automation and autonomy is not sufficient and many elements of the human factor needs to be considered and evaluated when designing a system for unmanned vehicles [52, 53]. Galster [52] believes unmanned combat vehicles need to be broadened beyond simple human command and control. Autonomy can be used to enhance CO–O^2 that compliments desired behaviours in dynamic environments and results in enhanced planning and interactions amongst unmanned vehicles. Additionally, autonomy is an important component in these type of systems as it supports command and control strategies [53, 54].

9.7.2 Concept of Control

An important feature with aircraft is their ability to control their flight as well as complete tasks and respond to commands. When the pilot is removed, operational control is removed, thus requiring the unmanned platform to perform telemetry, as well as cognitive commands [55, 56]. Tweedale [56] discusses how presently, the Human-in-The-Loop (HIL) commands and controls a platform remotely. By reducing the human from the environment can severely impact the platform's situation awareness, response time and cognitive interactions. The result is automation being implemented into control. Tweedale [55] expands this concept to show that control is the result of a hierarchical chain of commands that exist. Figure 9.6 depicts the types of commands required for human control.

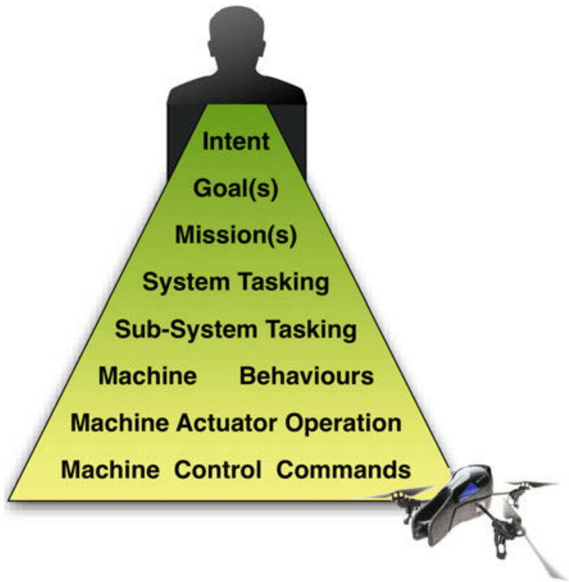


Fig. 9.6 Continuum of command and control (most primitive bottom) [55]

This concept describes control by highlighting the modes of operation required for an aircraft. He uses the example of an Unmanned Aerial Vehicle (UAV). Control is achieved by focusing on the Brooks sub-sumption methodology, which enables the incorporation of partitioned functionality [55, 57]. Such functions include, and are not limited to: seeking, landing, obstacle avoidance and route following. Using the navigation system example from Tweedale [56], AC^{3}M can be used to automate the control of multiple navigation systems from unmanned platforms. Since each agent is given a set of belief, desires, intentions from a stimulus event, once a new event enters the environment, the agent must adjust and accommodate. In this instance, a stimulus event would be a newly determined waypoint or geographical point. The `Stimulus/Perceptor` framework within AC^{3}M processes the event which will allow the lead agent to coordinate a number of agents to this waypoint.

The new waypoint event is analysed using the current instance of `Situational Awareness` to gather information. In this case, the mission only requires one agent to arrive at this waypoint, but two or more start to manoeuvre toward it. Entity-based coordination will occur allowing agents to cooperate. For example, if agent *A* arrives first to the new waypoint, it can coordinate the agent *B* to the next waypoint. Agent *A* will use the `Decisional Support` framework which allows this agent to issue a *Coordinative Event*. Once this has occurred, agent *B* will complete the event, which is to move onto the next waypoint, however requires navigational information from agent *A*. The information shared between the two agents is not just cooperation, but *Coordinative Cooperation*. A set of actions will then follow, which is agent *B* manoeuvring toward the next waypoint. By linking coordination and cooperation resulted in very little to no use of a supervisory role and enhanced automation.

An extension of Tweedale [55] and the introduction of synthesized expertise in Tweedale [56] are used to achieve further streamlined automation. Instead of an agent either failing to complete a goal due to their lack of expertise, an agent can call upon another which has the expertise to complete a goal. AC^{3}M heavily relies on this concept, since coordination is the managing of interdependencies and cooperation is the creation of a relationship, which are both based on the successful completion of a goal. More importantly, the expertise of an agent enables them to perceive, create mental models and use decision-making capabilities.

An example is as follows. Consider five entities, known as UAV-agents. Each UAV-agent is to detect a type of platform, which is labeled as their expertise and includes maritime, land, air and unknowns. One of UAV-agents is known as the `lead_UAV`. The UAV's must perform Intelligence, Surveillance and Reconnaissance (ISR) by reporting each event to the knowledge framework for further analysis. If an UAV-agent detects an event that is not associated with their expertise, it must use their situational awareness, mental modeling and belief, desires and intentions to ensure the information is reported.

The command with AC^{3}M is the coordination. AC^{3}M automates this process so the lead agent will coordinate the unmanned vehicle. Figure 9.7 shows an example when a UAV-agent within the simulation detects a water threat, and both the water UAV's are present.

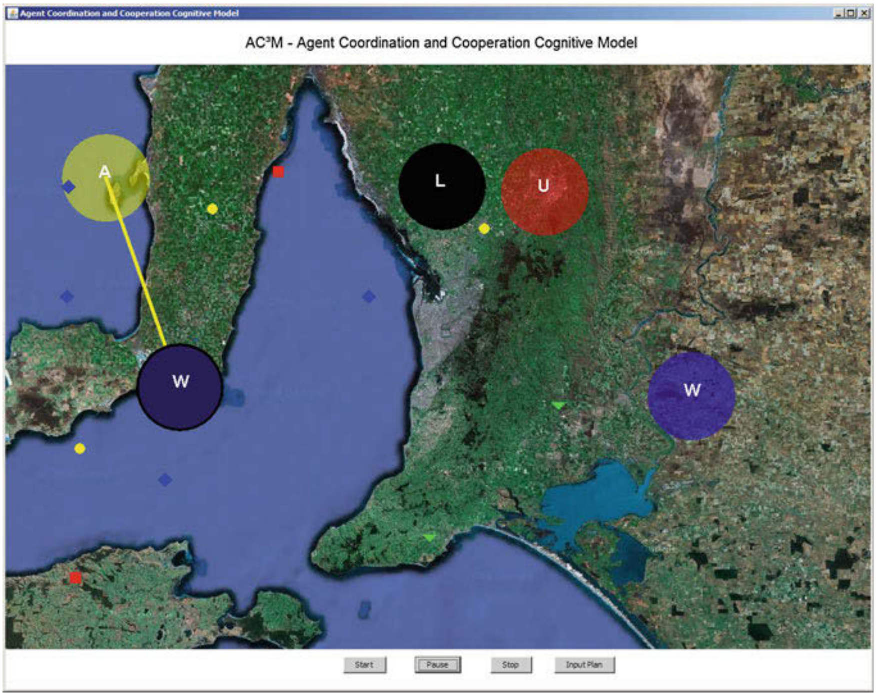


Fig. 9.7 Organisational coordination in AC^{3}M

When this situation arises within the system, the `lead_UAV` will coordinate based on a set of rules. The agent closest to the agent requiring cooperation will be coordinated. In the case where there are two expert agents present and one is constrained, the other will be coordinated to assist in the cooperation. Where two or more two expert agents exist and not constrained, the distance rule will be invoked. If all are constrained, the lead agent will wait until one is not constrained to coordinate. Additionally, the coordination of multiple expert agents will need to also consider resource and task allocations. Figure 9.7 also shows where there are two UAV-agents with a maritime expertise, and both are not constrained. The `lead_UAV` will command and coordinate the `water_UAV` that is closer to the `water_UAV`.

This example of command shows the relationship between an UAV-agent's beliefs, desires and intention and organisational coordination. When the `Air_UAV` has been organisationally coordinated into their new role, a `Coordinative Event` is fired and a plan is deduced. When the desires and intentions are updated, the `lead_UAV` notifies the existence more than one expert-UAV agent. At this stage, the `lead_UAV` had to organisationally coordinate one of the `Water_UAV_2` agents into this new *role*. In current systems, the human would coordinate the other UAV-agent to go and complete the ISR task. AC^{3}M automates this by *Coordinative Cooperation*. In this example, the `lead_UAV` will organisationally coordinate the `Water_UAV_2` to assist the `Air_UAV`. `Water_UAV_2` needs to agree to the cooperation, as the relationship must be voluntary. Cooperation can fail if the cooperating agent exceeds their resource or task allocation.

Once cooperation commences, the expert UAV-agent will cooperate by giving expert information on the target it has detected. Once this has occurred, the `Air_UAV` will be coordinated by the `Water_UAV_2` to complete their task. This process will continue until each of the goals have been completed successfully. In a team scenario, the automation of tasks, plans and actions are enhanced by linking coordination and cooperation. A supervisory role is still required, but CO–O^2 and their link is achieved autonomously and does not require the constant situation awareness from multiple sources. Instead, the UAV-agents can use their own mental modelling, cognitive processes and ISR to process information on a Stimulus Event.

9.8 Conclusion and Future Work

AC^{3}M capability in being the start of a new generation of hybrid coordination and cooperation methodology has signaled an exciting time in the area of team automation. Further, it has allowed researchers to view coordination and cooperation as a cognitive process, like it is viewed in traditional organisational theory. By encapsulating the theories of DAI with organisational theory in I-MASs, researchers are now in a position to investigate how mental models, personification and the environment need to be designed so they can fit with the ever changing needs of team automation. Furthermore, as more agents are added to these systems, the design must compensate for further processing, and this is where coordination and cooperation becomes vital.

AC^{3}M also investigated into how the mental model of an agent can affect coordination and cooperation. Through an agent's belief, desires and intentions, the coordination, both at an entity and system level, can change dramatically. The same can also be seen with cooperation. The importance of the link between an agent's belief, desires and intentions with coordination and cooperation has proven that AC^{3}M can enhance not only the automation of a team, but also their situational awareness. One application is in self-organising micro-mission systems. AC^{3}M is currently being used to coordinate and allow for cooperation between components of a mission flight system for each UAV.

By fusing coordination, cooperation and situational awareness together in one model, AC^{3}M establishes a new era of command and control systems. These systems can then facilitate, are more suitable, adaptable and more flexible for autonomous missions.

References

1. Guilford, J.: The Nature of Human Intelligence. McGraw-Hill (1967)
2. Simon, H.: Studying human intelligence by creating artificial. Am. Sci. **69**(3), 300–309 (1981)
3. Bass, B.: Cognitive, social and emotional intelligence of transformational leaders. In: Riggio, R., Murphy, S., Priozzolo, F. (eds.) Multiple Intelligences and Leadership, pp. 105–119. Lawrence Erlbaum Associates (2002)
4. Salovey, P., Deteiler-Bedell, B., Deteiler-Bedell, J., Mayer, J.: Emotional intelligence. In: Lewis, M., Haviland-Jones, J.M., Feldman Barrett, L. (eds.) Handbook of Emotions, pp. 533–548. Guilford Press (2008)
5. Thorndike, E.: Animal intelligence: an experimental study of the associate processes in animals. Am. Psychol. **53**(10), 1125–1127 (1998)
6. Kirn, S.: Organisational intelligence and distributed ai. Technical report, Institute of Business Informatics, University of Munster (1995)
7. van Aart, C., Wielinga, B., Schreiber, G.: Organizational building blocks for design of distributed intelligent system. Int. J. Hum. Comput. Stud. **61**(5), 567–599 (2004)
8. Perrow, C.: An organizational analysis of organizational theory. Contemp. Sociol. **29**(3), 469–476 (2000)
9. Carley, K., Gasser, L.: Computational organization theory. In: Wiess, G. (eds.) Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, pp. 299–330. The MIT Press (2000)
10. Carley, K.: Computational and mathematical organization theory: perspective and directions. Comput. Math. Organ. Theor. **1**(1), 39–56 (1995)
11. Prietula, M., Carley, K.: Computational organization theory: autonomous agents and emergent behavior. J. Organ. Comput. **4**(1), 41–83 (1994)
12. Eduardo, L., Herna, C.: On distributed artificial intelligence. Knowl. Eng. Rev. **3**, 21–57 (1988)
13. Moulin, B., Chaibdraa, B.: An overview of distributed artificial intelligence. In O'Hare, G., Jennings, N. (eds.) Foundations of Distributed Artificial Intelligence, pp. 3–55. John Wiley (1996)
14. Doran, J.: Distributed Al and its Applications. In: Advanced Topics in Artificial Intelligence, pp. 368–372. Springer (1992)
15. Gasser, L.: Distributed artificial intelligence. AI Expert **4**(7), 26–33 (1989)
16. Bond, A., Gasser, L.: Readings in Distributed Artificial Intelligence. Morgan Kaufmann (1988)
17. Consoli, A., Tweedale, J., Jain, L.: Aligning cognitive models using ac^{3}m. In: ICETET '08: First International Conference on Emerging Trends in Engineering and Technology, pp. 882 – 886. IEEE Computer Society (2008)
18. Nwana, H.: Software agents: an overview. Knowl. Eng. Rev. **11**(3) (1996)
19. Wooldridge, M.: Intelligent agents: the key concepts. In: Maik, V., Tpankova, O., Krautwurmova, H., Luck, M. (eds.) Multi-Agent Systems and Applications II, vol. 2322 of Lecture Notes in Computer Science, pp. 3–43. Springer (2002)
20. Wooldridge, M.: An Introduction to Multi-Agent Systems. Wiley, West Sussex, England (2002)
21. Wooldridge, M., Jennings, N.: Intelligent agents: theory and practice. Knowl. Eng. Rev. **10**(2), 115–152 (1995)
22. Consoli, A.: Enhancing team automation in dynamic environments, Ph.D. thesis. School of Electrical and Information Engineering, University of South Australia (2009)
23. Bergenti, F., Huhns, M.: On the use of agents as components of software systems. In: Bergenti, F., Gleizes, M.P., Zambonelli, F. (eds.) Methodologies and Software Engineering for Agent Systems, pp. 19–31. Kluwer (2004)
24. Bratman, M.: Intention, Plans and Practical Reasoning. Harvard University (1987)
25. Bratman, M., Israel, D., Pollack, M.: Plans and resource-bounded practical reasoning. Comput. Intell. **4**(4), 349–355 (1988)
26. Bratman, M.E.: Intention and personal policies. Philos. Perspect. **3**, 443–469 (1989)

27. Omicini, A.: From objects to agent societies: abstractions and methodologies for the engineering of open distributed systems. In: Corradi, A., Omicini, A., Poggi, A. (eds.) WOA 2000: 1st Joint Workshop "From Objects to Agents": Evolutive Trends of Software Systems, pp. 29–34. Pitagora Editrice Bologna, Parma, Italy, 29–30 (May 2000)
28. Henderson-Sellers, B., Giorgini, P.: Agent-oriented methodlogies: an introduction. In: Henderson-Sellers, B., Giorgini, P. (eds.) Agent-Oriented Technologies, pp. 1–19. Idea Group (2005)
29. Iglesias, C.A., Garijo, M., Centeno-González, J.: A survey of agent-oriented methodologies. In: ATAL'98: 5th International Workshop on Intelligent Agents V, Agent Theories, Architectures, and Languages, pp. 317–330. Springer, London, UK (1999)
30. Singh, M.: Agent-based abstractions for software development. In: Bergenti, F., Gleizes, M.P., Zambonelli, F. (eds.) Methdologies and Software Engineering for Agent Systems, pp. 5–18. Kluwer (2004)
31. Consoli, A., Tweedale, J., Jain, L.: An architecture for agent coordination and cooperation. In: Apolloni, B., Howlett, R.J., Jain, L.C. (eds.) KES'2007: 11th International Conference on Knowledge-Based Intelligent Information and Engineering Systems, pp. 934–940. Springer, Vietre sul Mare, Italy, 12–14 Sept 2007
32. Consoli, A., Tweedale, J., Jain, L.: Defining a decision-support framework in ac^{3}m. In: Nakamatsu, K., Phillips-Wren, G., Jain, L., Howlett, R. (eds.) IDT'09: First KES International Symposium in Intelligent Decision Technologies, pp. 203–212 (2009)
33. Bocchi, L., Ciancarini, P.: A perspective on multiagent coordination models. In: Huget M.-P. (ed.) Communications in Multiagent Systems. Volume 2650 of Lecture Notes in Computer Science, pp. 146–163. Springer (2003)
34. Busi, N., Ciancarini, P., Gorrieri, R., Zavattaro, G.: Coordination model: a guided tour. In: Omicini, A., Zambonelli, F., Klusch, M., Tolksdorf, R. (eds.) Coordination of Internet Agents: Models, Technologies and Applications. Lecture Notes in Computer Science, pp. 6–23. Springer (2000)
35. Denti, E., Omicini, A., Ricci, A.: Coordination tools for the development of agent-based systems. In: Trappl, R. (ed.) EMCSR 2002: 16th European Meeting on Cybernetics and Systems Research, pp. 671–676. Austrian Society of Cybernetic Studies, Vienna, Austria (2002)
36. Consoli, A., Tweedale, J., Jain, L.: The link between agent coordination and cooperation. In: Shi, Z., Shimohara, K., Feng, D. (eds.) ICIIP2006—3rd International Conference on Intelligent Information Processing, vol. 228, pp. 11–19. Adelaide, South Australia, Australia, Springer (2006)
37. Consoli, A., Tweedale, J., Jain, L.: An overview of agent coordination and cooperation. In: Gabrys, B., Hewlett, R., Jain, L. (eds.) KES2006: 10th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (2006)
38. Nii, H.: The blackboard model of problem solving. AI Mag. **7**(2), 38–53 (1986)
39. Smith, R.: The contract net protocol: high-level communication and control in a distributed problem solver. IEEE Trans. Comput. **29**(12), 1104–1113 (1980)
40. Durfee, E., Lesser, V., Corkill, D.: Trends in cooperative distributed problem solving. IEEE Trans. Knowl. Data Eng. **1**(1), 63–83 (1989)
41. Wooldridge, M., Jennings, N.: The cooperative problem-solving process. J. Logic Comput. **9**(4), 563–592 (1999)
42. Ramesh, V., Glass, R., Vessey, I.: Research in computer science: an empirical study. J. Syst. Softw. **70**, 165–176 (2004)
43. Zambonelli, F., Jennings, N., Wooldridge, M.: Developing multiagent systems: the gaia methodology. ACM Trans. Softw. Eng. Methodol. **12**(3), 317–370 (2003)
44. Zambonelli, F., Jennings, N., Wooldridge, M.: Multi-agent systems as computational organizations: the gaia methodology. In: Henderson-Sellers, B., Giorgini, P. (eds.) Agent-Oriented Methodologies, pp. 136–171. Idea Group (2005)
45. Parasuraman, R., Riley, V.: Humans and automation: use, misuse, disuse, abuse. Source: Human Factors: J. Hum. Factors Ergon. Soc. **39**(2), 230–253 (1997)

46. Sheridan, T., Parasuraman, R.: Human-automation interaction. Rev. Hum. Factors Ergon. **1**, 89–129 (2005)
47. Cuevas, H., Fiore, S., Caldwell, B., Strater, L.: Augmenting team cognition in human-automation teams performing in complex operational environments. Aviat. Space Environ. Med. **78**, 63–70 (2007)
48. Nisser, T.: Human factors challenges in unmanned aerial vehicles (UAVs): A literature review. Technical Report TFHS:05:1, Lund University School of Aviation (2006)
49. Parasuraman, R., Sheridan, T., Wickens, C.: A model for types and levels of human interaction with automation. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **30**(3), 286–297 (2000)
50. Wickens, C.: Engineering Psychology and Human Performance. Harper Collins (1992)
51. Endsley, M., Bolstad, C., Jones, D., Riley, J.: Situation awareness oriented design: from user's cognitive requirements to creating effective supporting technologies. In: Human Factors and Ergonomics Society Annual Meeting: Cognitive Engineering and Decision Making, pp. 268–272 (2003)
52. Galster, S.: Human automation integration. In: Uninhabited Military Vehicles: Human Factors Issues in Augmenting the Force, pp. 1–108. NATO Research and Technology Organisation (2006)
53. Schulte, A.: Artificial cognition and cooperative automation. In: Uninhabited Military Vehicles: Human Factors Issues in Augmenting the Force. NATO Research and Technology Organisation (2006)
54. McCarley, J., Wickens, C.: Human factors implications of UAVs in the national airspace. Technical Report AHFD-05-05/FAA-05-01, Institute of Aviation, University of Illinois (2005)
55. Tweedale, J.: Fuzzy control loop in an autonomous landing system for unmanned air vehicles. In: WCCI 2012: 2012 IEEE World Congress on Computational Intelligence (2012)
56. Tweedale, J.: Using mutli-agent systems to improve the level of autonomy for operators controlling unmanned vehicles. In: KES'12: Advances in Knowledge-Based and Intelligent Information and Engineering Systems, pp. 1666–1675. IOS Press, (Sept 2012)
57. Brooks, R.A.: A robust layered control system for a mobile robot. IEEE J Robot. Autom. **2**(1), 14–23 (1986)

Chapter 10
Wind Rendering in 3D Modeling Landscape Scenes

Margarita Favorskaya and Anastasia Tkacheva

Abstract The modeling of 3D landscape scenes includes two main issues: the creation of landscape scene and the natural effects rendering. A hybrid approach based on laser data scanning and templates of L-systems was developed to design trees and brush of various types and sizes. The space colonization algorithm was applied to make the tree models realistic and compact described and stored. Wind rendering is a necessary procedure, without which any modeling scene looks non-realistic. Three algorithms for wind rendering under changeable parameters were proposed. They have a minimal computational cost and simulate weak wind, mid-force wind, and storm wind. The approach based on mega-texture visualization was used to make a 3D landscape scene with wind effects a real-time application. The user can tune the various trees and wind parameters and manipulate a modeling scene by using the software tool "REWELS" designed in the development environment RAD Studio 2010.

Keywords Wind rendering · Laser scanning · Space colonization algorithm · L-system · 3D landscape scene · Mega-texture

10.1 Introduction

A landscape scene modeling depends from the solved task, and is based on two different approaches, i.e. a full generation of scene by software models and a realistic rendering of natural scene using laser data and aerial/Earth images. In any case, the rendering of natural scenes with vegetations is based on two types of models. As shown in survey of Zhang and Pang [1], the models for separate types of multi-level vegetation and co-influence models, connecting with growing and stagnation of some

M. Favorskaya (✉) · A. Tkacheva
Siberian State Aerospace University, 31 Krasnoyarsky Rabochy, 660014 Krasnoyarsk, Russia
e-mail: favorskaya@sibsau.ru

A. Tkacheva
e-mail: tkacheva@sibsau.ru

J.W. Tweedale et al. (eds.), *Intelligent Decision Technology Support in Practice*, Smart Innovation, Systems and Technologies 42,
DOI 10.1007/978-3-319-21209-8_10

plants under their life-cycle, received the most propagation. Some researches deal with very complex biological modeling, which require many initial parameters and their values. Thus, in original research of Qu et al. [2], an agent-based functional–structural model for orange tree growth simulation was developed, which is based on 42 various input parameters considering a photon flux density, an air temperature, a soil humidity, the state transitions. This is a prediction model for orange tree growth in several years duration. Such modeling is based on very complex geometrical and probability dependences.

The recent researches are focused on a geometry reduction technique in 2D or 3D spaces. Neubert et al. [3] represented an improved model of pruning in large botanical scenes. A 3D representation has the unchallengeable visualization advantages for the user. In literature, three main approaches of trees modeling exist: a rule-based generation, an interactive modeling, and an image-based modeling. The first approach was introduced by Lindenmayer [4] in 1968 (L-system) as the foundation of an axiomatic theory of biological development. It is often used in a computer graphic. The high level controlling the grammar-based procedural models was proposed by Talton et al. [5]. A procedural method for irregular tree models based on a weighted irregular graph (the Yao graph) was developed by Xu and Mould [6]. As a result, very realistic trees models can be built. The second approach is based on 2D sketch models of trees with following 3D models creation. A technique of plant structures modeling using sketches was developed by Anastacio et al. [7]. Chen et al. [8] designed an original approach for a sketch-based tree modeling using markov random field. The third approach connects with a plant or a tree model creation from a set of images. If such shooting will be around the plant, then a 3D plant model can be built. Reche et al. [9] used a set of photos for a volumetric reconstruction and an interactive rendering of trees. A hybrid approach was proposed by Anastacio et al. [10], combining the advantages of L-systems and sketch models. The use of botanical illustration- inspired construction lines were proposed to parameterize global features of L-system models. Illustrators apply such construction lines to determine the global features of the plant being depicted. The application of these lines makes faster the illustration process indicating a posture, contours, proportions, topology, and constraints. Using the pressure data obtained from the pen of a tablet permitted the authors to built 3D view from 2D sketch.

A separate approach connects with realistic modeling of leaves during large-scale shooting. The synthesis algorithm proposed by Assa and Cohen-Or [11] can generate a large set of differently looking leaves based on a small number of structural layers, each of which includes a texture with certain characteristics of a leaf, shape features, and colour spots.

Some methods deal with a tree model reconstruction from a cloud of laser points. A heuristic-based modeling was proposed by Xu et al. [12]. Livny et al. [13] designed an automatic reconstruction of tree structures from the point clouds. The development of hybrid methods based on the existing approaches of trees modeling is an actual research task. Jaakkola et al. [14] proposed a mobile mapping system operating onboard a car and a mini unmanned aerial vehicle for individual tree measurements.

It includes a Global Positioning System/Inertial Navigation System (GPS/INS), the laser scanner, charge-coupled device-cameras, and other sensors.

This chapter provides the hybrid method based on laser data of tree representation, selection of tree template by using a visual aerial image, and modification of geometrical parameters for the required tree template. Tree templates are built as L-systems. They are stored in a special database. When a tree template is "fitted" under laser data, a space colonization algorithm for more realistic visualization of branches and leaves are used. In this research, it is suggested that a landscape scene has a uniform vegetation distribution with a middle density of trees in small part of forest. It is accepted that a modeling scene owns a complex structure with mountains and multi-level surfaces. A wind behavior near trees is a weak predictable process. Therefore, the main task was to achieve the effect of maximum similarity and visibility for the user. The authors contribution connects with more accurate trees modeling based on laser data and L-system and development of real-time algorithms of wind rendering according to desirable wind parameters.

The remainder of the chapter is organized as follows. In Sect. 10.2, a brief description of the existing approaches for a wind rendering is presented. Section 10.3 describes the proposed basic tree modeling. Section 10.4 includes the algorithms for a wind rendering (weak wind, mid-force wind, and storm). A software tool description and examples of rendering are provided in Sect. 10.5. Conclusions are situated in Sect. 10.6.

10.2 Related Work

The rendering of dynamic effects of trees under natural wind is a common task in the outdoor modeling scenes but this task has not a unique satisfactory decision. The model of wind distribution strongly depends from the scientific or application scope. Usually the wind models are applied to trunk and branches. Leaves remain non-movement and non-realistic. However, a condition classification can be introduced for conifers with non-movement acerose leaves and broad-leaved trees in the pre-determined scales. In this research, the algorithms for realistic and real-time realization in 3D landscape scenes are developed.

In their researches, Habel et al. [15] and Akagi and Kitajima [16] considered wind effects near trees as a physical phenomenon. Chuang et al. [17] investigated tree motion textures under a wind field by harmonic oscillations. Another possibility of trees motion modeling connects with a filtering of white noise in the Fourier domain. One of serious attempts of tree animation based on physical fidelity was made by Zhang et al. [18]. The authors suggested the motion graph algorithm that includes a set of representative motions for each tree model. A motion is generated by inputting a sampling configuration based on three components:

1. A set of physical parameters for each tree branch.
2. An initial state of tree model.
3. A time-varying local wind field around a tree model.

Also they proposed a dynamic vibration of wind over time to make the process more realistic. Such approach requires a great variety of templates for each type of trees and wind conditions.

The technique of response function of branches to turbulent wind in a frequency domain and a branch motion by sampling 2D motion texture was presented by Habel et al. [15]. The key idea for a real-time performance was to calculate vertex displacements in a tree model. The hierarchical deformations of all parent branches can be explicitly performed inside the vertex shader. Each vertex is associated with an index into a texture, and four hierarchical levels of vertexes are simulated.

Fugmann et al. [19] proposed the anisotropic mathematical model to simulate the sway of a Scots pine tree in response to turbulent wind loading. The authors solved the differential equation of displacement of each conditional endpoint of segments for 0.2 m. A tree tilt was approximated by a cubic spline interpolation of segment displacement provided by Eq. 10.1, where $\mathbf{M}$ is a mass matrix with sizes $n \times n$ whose elements are m_{ij} $(i, j = 1, \ldots, 15)$, $\mathbf{D}$ is a damping matrix with sizes $n \times n$ and elements d_{ij}, $\mathbf{K}$ is a $n \times n$ stiffness matrix with elements k_{ij}, $\mathbf{F}$ is a $n \times 1$ load vector with elements f_j, and $\mathbf{z}$ is a $n \times 1$ trunk displacement vector with elements z_j.

$$\mathbf{Mz}'' + \mathbf{Dz}' + \mathbf{Kz} = \mathbf{F} \tag{10.1}$$

According to biomechanics foundations, Newson et al. [20] considered the soil-root-tree system and studied the influence of wind forces on such system. The authors assumed that a tree model is a combination of vertical (V), horizontal (H) and moment (M) loadings due to a self weight and a wind loading. The component of wind velocity field $u(y)$ applying to a spatially variable horizontal load on a tree $H(y)$ is estimated by Eq. 10.2 suggested by Peltola and Kellomaki [21], where ρ_{air} is a fluid density, C_d is a drag coefficient, $A_f(y)$ is a frontal area, $u(y)$ is a wind horizontal velocity, y is a coordinate in XOY space.

$$H(y) = \frac{\rho_{air}}{2} C_d A_f(y)\; u^2(y) \tag{10.2}$$

The moment applied to a root of tree, may be separated into two components presenting by Eq. 10.3.

$$M = H \cdot y_w + V \cdot x_w \tag{10.3}$$

This moment is tuned clockwise. Let us notice that Eq. 10.2 needs in large empirical data for estimation of spatially variable horizontal load on a tree by a known wind horizontal velocity $u(y)$.

Seidl et al. [22] and Hale et al. [23] investigated wind damage in the simulated forest. They calculated a vertical wind profile, individual tree turning coefficient stand and soil conditions. Then neighborhood effects on tree stability are estimated, and, at least, critical wind speeds were calculated separately for uprooting and stem breakage. The Individual-based LANdscape and Disturbance model (iLand) was used as the simulation platform for the development of this wind disturbance model (Seidl et al. [24, 25]).

There are four main rendering primitives, which are used in the computer graphics:

1. Polygons. For forest rendering, it is the enough costly representation.
2. Image-based primitives. An image-based rendering became the major technique for rendering realistic scenes. In spite of low cost of a single image, various images are required (with lighting, scaling, and animation conditions) only for one tree modeling. A high memory volume is needed for such approach.
3. Volumetric and shader-based approach. Sometimes fuzzy primitives with a detailed merge in low scale can be interpreted as a natural tree. This approach is used in computer virtual reality.
4. Point-based primitives. Such approach is based on reliable data of laser scanning. It is well known two types of forest laser scanning: from the Earth' surface and from helicopter or airplane equipped by special devices.

In this research, a hybrid method based on a point-based approach for tree modeling and a shader-based algorithm for trees animation under wind conditions was developed.

10.3 Tree Modeling Based on a Cloud of Laser Points and L-System

The restoration of trees view from a cloud of laser data by using a visual aerial image is a very complex task, which sometimes has not a unique performance. The action of any laser scanners is based on a distance measure from the source of laser pulses to an opaque object (Bunting et al. [26]). A beam creating by a laser transmitter is reflected from an object surface. When a reflected signal comes back into a scanner receiver, one can calculate a distance value by a time delay (pulse method) or a phase displacement (phase method) between the emitted pulse and the reflected pulse. If scanner coordinates and a pulse direction are known then 3D points coordinates of object, from which a laser pulse was reflected, may be determined. Laser pulses are generated with a frequency near 200 pulses per s by a complex 3D trajectory, and as a result a cloud of points, which describes an object with a high accuracy, is received. If a reflected pulse is a single, then the Earth' surface or any opaque object lays on this direction. If two (maximum three) reflected pulses along any beam are appeared, then it will mean that any transparent object is situated before the Earth' surface or an opaque object. Laser data are the most reliable source of 3D coordinates into a scene. Trees are semitransparent objects. Therefore, it is needed to separate primary laser pulses from secondary laser pulses. A methodology for detection secondary laser pulses (Earth' surface) and primary laser pulses (trees into a landscape scene) was proposed by Favorskaya et al. [27]. Examples with various reflected laser impulses are situated in Fig. 10.1.

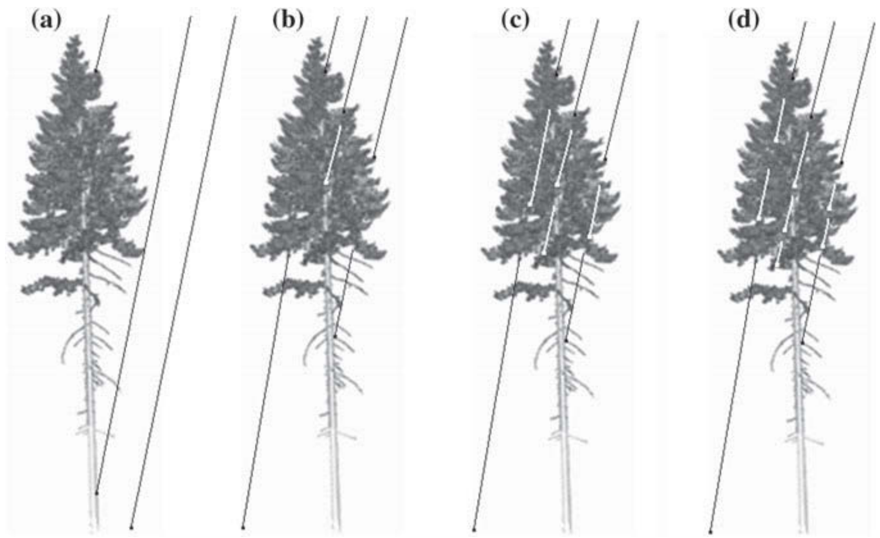


Fig. 10.1 Examples with reflected laser impulses: **a** with one reflected impulse, **b** with two reflected impulses, **c** with three reflected impulses, **d** with four reflected impulses

When a cloud of points concerning to tree $\mathbf{SS}_T$ is received, it is needed to separate this set into two subsets: a subset of crown points $\mathbf{SS}_C$ and a subset of branches points $\mathbf{SS}_B$ by using Eq. 10.4.

$$\mathbf{SS}_T \supseteq \mathbf{SS}_C \bigcup \mathbf{SS}_B \tag{10.4}$$

According to heuristic rules (different intensities of crown, trunk, and branches), empirical threshold values to separate subsets $\mathbf{SS}_C$ and $\mathbf{SS}_B$ from Eq. 10.4 can be determined. For restoration of tree geometry based on a cloud of points, additionally a stochastic grammar of Lindenmayer [4] and Space Colonization Algorithm (SCA) developed by Runions et al. [28] are applied. The SCA was originally introduced to model leaf venation patterns and the branching architecture of trees. It simulates the competition for space between growing veins or branches. However, other applications of SCA are known; the most famous among others is a crowd behavior simulation (de Lima Bicho et al. [29]). SCA interactively checks elements belonging to a geometric structure of tree and marks the real points of laser scanning.

L-system is constructed according to a stochastic grammar representing by Eq. 10.5, where $\mathbf{V}$ is an alphabet, W is an axiom, $\mathbf{PR}$ is a set of productions, a function $d : \mathbf{PR} \rightarrow (0,1)$ is a probability distribution. Such distribution converts a set of productions $\mathbf{PR}$ into a set of production probabilities. The sum of probabilities of all productions is equaled 1.

$$\mathbf{SG}_d = \langle \mathbf{V}, W, \mathbf{PR}, d \rangle \tag{10.5}$$

Favorskaya et al. [30] extended the stochastic grammar Eq. 10.5 applying the procedures of growing and aging of trees. An axiom W was specified by a set of initial parameters for a trunk generation. A growing procedure PR_G was introduced for periodically restarting and recalculating parameters of all branches and trunk. Also a degenerating procedure PR_D was represented for a periodically recall for deletion of separate tree branches. By using such procedures, a tree skeleton is built. There are provides some examples of monopodial structures, sympodial structures, and ternary branching of trees in Fig. 10.2.

The task is to select such structure of tree, which will be maximally matched with laser and visual data. Visual data of trees crowns help to determine a type of tree in automated mode by using the expert recommendations. Also during branches modeling, Hermit curves were used to make a branching more realistic (Fig. 10.3). This procedure is applied to the main branches of tree.

The main idea of SCA is in an interactive adding of new elements (branches) to a geometrical structure of object (tree) created on previous steps. The SCA considers the environment man-caused objects and neighbor trees. Initially, a cloud of points (leaves) and trunk points are determined. During each iteration, a branch is built to the nearest point of a tree crown. In such manner, a skeleton of a tree is built. SCA includes several steps:

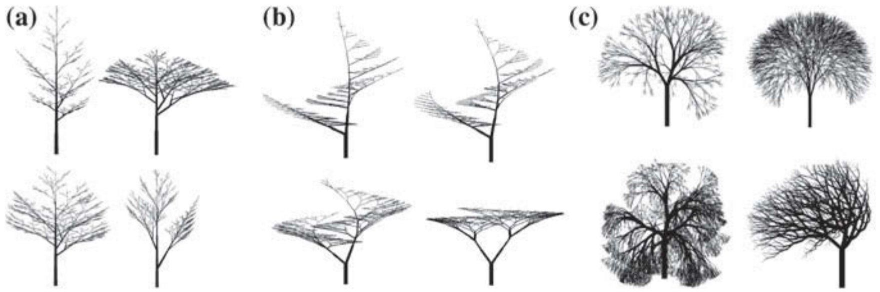


Fig. 10.2 Examples of trees structures: **a** monopodial structures, **b** sympodial structures, **c** ternary branching

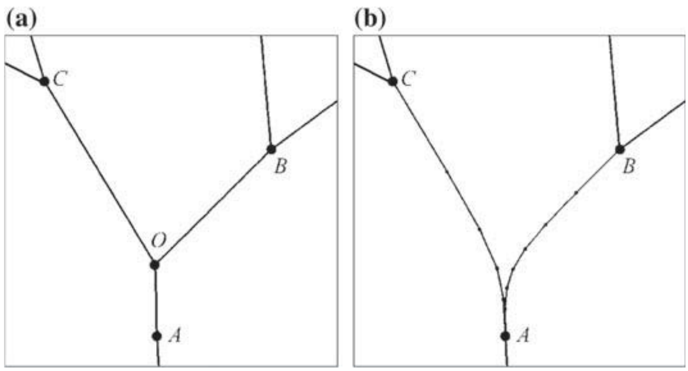


Fig. 10.3 Branches modeling: **a** by line segments, **b** by Hermit curves

1. A detection for an initial point of trunk. From this point, the building of tree skeleton is begun.
2. A crown points matching. During iterations for each current node, a branch to the nearest point of crown is generated and matched in a tree model. A tree model may be rotated for better matching. Such iterations are finished, when all crown points will be associated with branches or number of iterations determined by the user will be accomplished.
3. A resampling of branches points.
4. A smoothing sharp transition of branches.
5. A leaves mapping. A skeleton of tree is covered by a leaves texture according to a tree template in a desirable scale. Such leaves textures may be previously prepared and chosen by the user from a texture library.

The results of initial steps of SCA work are situated in Fig. 10.4.

The example of final tree modeling by using data of laser scanning (received from the Earth' surface) and L-system paradigms is shown in Fig. 10.5.

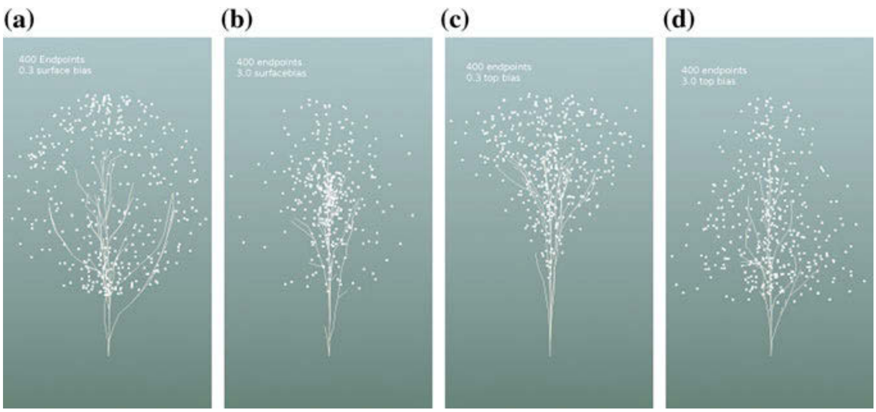


Fig. 10.4 The results of initial steps of SCA work with various distribution of cloud points: **a** boundaries distribution, **b** central distribution, **c** top distribution, **d** bottom distribution

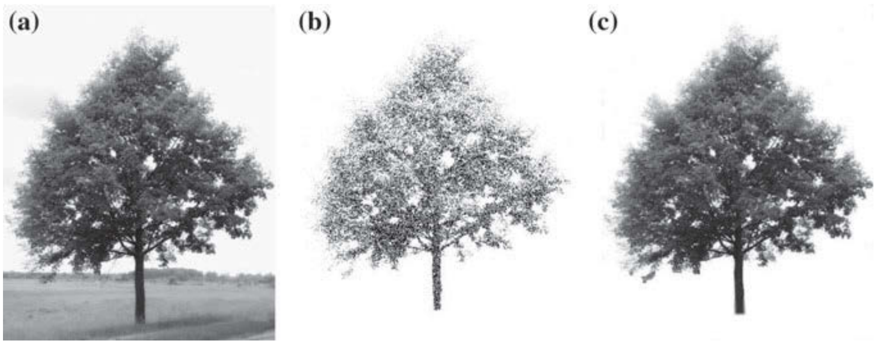


Fig. 10.5 Hybrid trees modeling: **a** a visual image of tree, **b** 3D cloud of laser data, **c** a modeling result

10.4 Algorithms of Wind Rendering in Landscape Scenes

Three basic wind rendering algorithms concerning to weak wind (Sect. 10.4.1), mid-force wind (Sect. 10.4.2), and storm rendering (Sect. 10.4.3) were developed. Storm hazard is enough rarely natural effect that may occur once during several years. Two other algorithms find more often applications into a landscape scenes modeling.

10.4.1 Weak Wind Rendering

A weak wind rendering is the enough simple procedure with random wind direction and minimum differences of leaves flicker. This effect is needed in a scene foreground and may be determined as a redrawing procedure of leaves texture on a tree skeleton. The flickering effect into a scene background is realized as a replacement of intensities of some random pixels concerning to leaves texture with a frequency 2–3 Hz. Such temporal periodicity can be simulated by using a generator of random values relatively pixels positions and pixels intensities.

In all three algorithms of wind rendering including a weak wind rendering, a mega-texture mapping is applied. It means that a landscape scene is covered by a single base texture with large sizes instead of a set of textures with small sizes. More detailed information about mega-texture mapping one can find in site.[1]

Such technology is a very perspective approach for rendering in outdoor spaces and has the following statements:

1. Mega-texture is stored in Hard Disk Drive (HDD) and is loaded into Random Access Memory (RAM), when the detailed repainting of landscape scene is necessary. Such approach saves RAM resources essentially.
2. The nearest foreground pixels are replaced by the highest quality texture with restricted sizes.
3. The following pixels are replaced by the high quality texture with a usual resolution.
4. The background pixels are the base texture with a decreased resolution.

Such technology was designed for a landscape texturing to improve a view range and higher realistic properties of modeling scene. A scene surface is divided on different Levels Of Details (LODs), and for each level the texture with required quality properties (clipmap) is designed. Usually it is enough 5 detailed levels. If clipmap sizes are 512×512 pixels, then data of $512 \times 512 \times 5 \times 4 \approx 5$ Mb for 5 levels and 32 bit for each pixels ($8 \times 4 = 32$ bit) are needed. Such calculation shows that the required volume of video RAM has very small values for repainting a landscape scene with the high quality and resolution.

---

[1]OpenGL Performer Programmer's Guide: ClipTextures. Available online, at http://techpubs.sgi.com/library/tpl/cgi-bin/getdoc.cgi?coll=0650\&db=bks\&srch=\&fname=/GI_Developer/Perf_PG/sgi_html/ch15.html.

The key property of such texturing approach deals with a texture loading into a video RAM only during virtual camera displacements or with changes of a view point. When a virtual camera is moved then transitions of texture, the LODs are realized. If a view point displacement is higher a pre-determined threshold value, then texture coordinates are recalculated. The main advantage of this technology is a possibility to work with small texture volumes that supports a high computer speed.

10.4.2 Rendering of Middle Speed Wind

For real-time application, a simulation model of wind influence on a tree is designed. In this case, a fractal model of tree is required, which is not based on data of laser scanning. Three main steps are involved into such fast algorithm: building of tree skeleton, wind rendering, and adding of branches randomness and inertia.

The design of tree model based on fractal schema has the following simplifications. Let two vectors with known origins and directions exist. Triply new directions disposing them on some angle are calculated. The length of a new vector must be shorter of previous vectors (it is achieved by using a correction coefficient). The new position becomes the end of previous vector. Thereby, the first vector is a trunk of a tree model, from which three branches are continued. The experiments show that the depth of recursion equaled to 7 is enough to simplify a tree model.

Wind rendering is a simulation of branches flexure caused by wind. It means that coordinates of "branch-vectors" ought to be recalculated. It is possible to calculate a rotation matrix around an axis passing through the origin of "branch-vector" and the normal to a direction of "wind-vector". But on seventh recursion level, such calculations will be too complex without a real-time execution.

The simple and less exact method was used under the assumption that a wind direction can not be vertical, and branch displacements are situated in one plane with "branch-vector" and "wind-vector". The task is to find displacements of "branch-vectors" ends along axes OX and OZ and to normalize the result so that the lengths of "branch-vectors" do not change. In such manner, the own oscillations of branches are simulated. Additionally the oscillations of supporting branches are considered. Finally coordinates of "branch-vectors" are recalculated. The trunk is considered as a stationary object that has a small tilt angle under a wind influence. Also the dependence between a thickness of branch and a branch flexure was used. Such correction coefficients are determined as tabulated values.

The adding of branches randomness and inertia is a necessary step for an acquisition of realistic effect. The wind randomness is well described by Perlin noise [31]. A Perlin noise generation is one of popular procedures for design of random procedural textures. The method consists in a composition of usual noises with difference frequencies and amplitudes inversely proportional to frequency. The branch inertia is the gradual position changes by an immediately changeable wind direction. For simulation of such effect, the branches positions do not changed but the inertial wind directions from frame to frame are corrected by a linear interpolation.

10.4.3 Storm Rendering

The storm damage is a hard prevented phenomenon. It may be estimated as a natural risk in a long time domain. A tree animation in environment of hurricane impacts was proposed by Singh et al. [32]. Usually the consequences of storm rendering are determined by the types of trees, their heights and state of roots, topography, climate, and soil properties. It is important to consider a tree model not only as a mechanical system, which is exposed to horizontal component of strong wind force, but also as ecological system having its own shapes of trunk, branches, and crown. The effect of storm wind may be not only a branches break, but also an uprooting or a trunk break on a determined height. Seidl et al. [22] developed a wind modeling approach that simulates wind impact at the level of individual trees and wind disturbance events iteratively. The algorithm includes the following steps:

1. The edges detection in the landscape.
2. The calculation of vertical wind profile.
3. The calculation of individual-tree turning coefficient.
4. Critical wind speeds calculation.
5. The simulation of wind impact on vegetation.

Seidl et al. used a research software tool 'iLand' in their experiments.

Two scenarios of behavior may be suggested dependently from "vertical" or "horizontal" roots of a tree. The first model (with deep root system) considers a tree as a rigid column on the elastic base. The deviation angle θ of such "column" is determined from Eq. 10.6, where P is a horizontal moment of wind, h is a point ordinate of application of force P, Q is a force moment of column weigh, l is a distance from root to a point ordinate of application of force P, J is an inertia moment of root system, c is a proportional coefficient.

$$Ph - Ql\theta + cJ\theta = 0 \tag{10.6}$$

In Eq. 10.6, a simplifying condition $\theta \approx \sin\theta$ was made. A critical load value will be when $P = 0$ in Eq. 10.6. Therefore Eq. 10.7 provides a critical load Q_c.

$$Q_c = cJ/l \tag{10.7}$$

Equation 10.7 is right until a root system would not break from the Earth surface. If this process begins, then Eq. 10.6 will become a nonlinear dependence.

The second model (with a non-deep root system) has another interpretation. Under a wind influence, a tree also tilts but in this case the gravity and the reaction force of root system are situated not in one plane. This model generates forces, which aim is to return back a tree into a normal position. A tree tilt is less for the second model and achieves a maximum durability by $\theta = 10$–$12°$. In the first model, this parameter achieves approximately 30–35°. The approximate values of θ (according to wind force, type of tree, and tree height) are stored in database. For simulation effects, the procedures of uprooting or trunk break are randomly executed.

10.5 Software Tool and Experimental Results

Software tool REndering of Wind Effects into Landscape Scenes (“REWELS”) contains three main modules: a module of tree generation based on laser data and L-system, a module of 3D scene generation, and a rendering module, which includes three developed algorithms of wind simulation in dependence of wind speed. The designed software tool has a possibility not only to tune wind parameters but also move, rotate, and scale the modeling scene. In Fig. 10.6, maximum tilts of tree models (on assumption that a wind direction is along axis OX) are shown by rendering of middle speed wind. Simplified fractal models of trees and real trees images with different wind speed from 0.3 up to 21.0 m/s are represented in Fig. 10.6a, b, respectively. The accuracy of rendering may be estimated only on a qualitative level based on the expert estimations. Such expert estimations are better for low-scaled landscape scenes.

The modeling results of storm wind rendering are shown in Fig. 10.7.

For program realization, the graphical jet GLScene applying the library OpenGL as application programming interface in development environment RAD Studio 2010 was used. Some standard objects from the jet GLScene such as TGLCamera (a camera object), TGLSceneViewer (3D viewer object), TGLMaterialLibrary (a library of materials), TGLFreeForm (a static 3D model), and the others were applied.

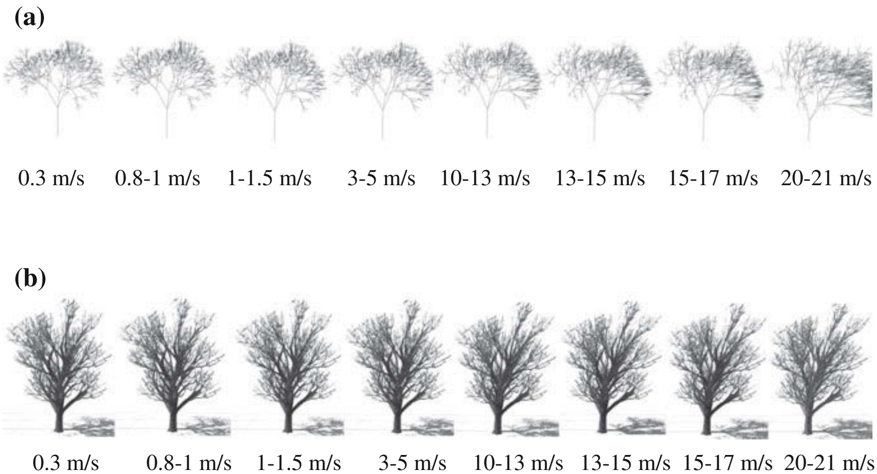


Fig. 10.6 Tilts of threes models. **a** Fractal models of trees, **b** real images of trees

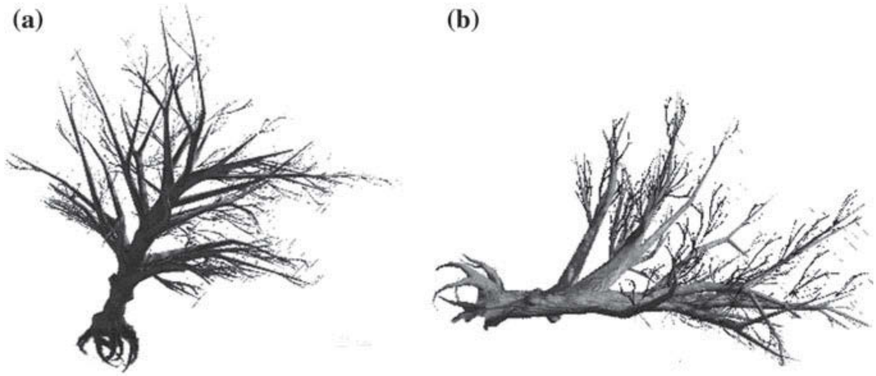


Fig. 10.7 Examples of storm wind rendering. **a** Maximum tilt of tree, **b** uprooting modeling

10.6 Conclusion

In this chapter, the process of tree restoration by using laser data with a tree modeling based on L-system templates for more realistic view of modeling forest scene was investigated. Laser data were used to restore a view of Earth surface and natural objects into landscape scenes. The hybrid method based on laser data of tree, selection of tree template using a visual aerial image, and modification of geometrical parameters for required tree template was designed. The mega-texture mapping was applied for a landscape texturing with increased view range and higher realistic properties of modeling scene. For wind effects rendering, three basic algorithms of trees animation under weak wind, middle speed wind, and storm wind were developed. Owing to the stochastic structures of trees, all algorithms are the approximate procedures with maximum visual effect for the user.

Three main modules include into software tool "REWELS". There are a module of tree generation based on laser data and L-system, a module of 3D scene generation, and a wind rendering module. The software tool is based on the graphical jet GLScene (with applying the open library OpenGL) and own designed components.

References

1. Zhang, Q.L., Pang, M.-Y.: A survey of modeling and rendering trees. In: Pan, Z., Zhang, X., Rhalibi, A.E., Woo, W., Li, Y. (eds) Technologies for E-Learning and Digital Entertainment, LNCS Edutainment, pp. 757–764. Springer Link, Heidelberg (2008)
2. Qu, H., Wang, Y., Cai, L., Wang, T., Lu, Z.: Orange tree simulation under heterogeneous environment using agent-based model ORASIM. Simul. Model. Pract. Theory **23**, 1935 (2012)
3. Neubert, B., Pirk, S., Deussenl, O., Dachsbacher, C.: Improved model- and view-dependent pruning of large botanical scenes. Comput. Graph. Forum **30**(6), 1708–1718 (2011)
4. Lindenmayer, A.: Mathematical models for cellular interaction in development. J. Theor. Biol. **18**(3), 300–315 (1968)

5. Talton, J.O., Lou, Y., Lesser, S., Duke, J., Mech, R., Koltun, V.: Metropolis procedural modeling. ACM Trans. Graph. **30**(11), 114 (2011)
6. Xu, L., Mould, D.: A procedural method for irregular tree models. Comp. Graph. **36**(8), 1036–1047 (2012)
7. Anastacio, F., Sousa, M.C., Samavati, F., Jorge, J.A.: Modeling plant structures using concept sketches. In: 4th International Symposium on Non-Photorealistic Animation and Rendering (NPAR ÃÅ06), ACM, 105–113 (2006)
8. Chen, X., Neubert, B., Xu, Y.-Q., Deussen, O., Kang, S.B.: Sketch-based tree modeling using markov random field. ACM Trans. Graph. **27**(5), 19 (2008)
9. Reche, A., Martin, I., Drettakis, G.: Volumetric reconstruction and interactive rendering of trees from photographs. ACM (SIGGRAPH-04) 720–727 (2004)
10. Anastacio, F., Prusinkiewicz, P., Sousa, M.C.: Sketch-based parameterization of L-systems using illustration-inspired construction lines and depth modulation. Comput. Graph. **33**(4), 440–451 (2009)
11. Assa, J., Cohen-Or, D.: More of the same: synthesizing a variety by structural layering. Comput. Graph. **36**(4), 250–256 (2012)
12. Xu, H., Gossett, N., Chen, B.: Knowledge and heuristic-based modeling of laser-scanned trees. ACM Trans. Graph. **26**(4), 1931 (2007)
13. Livny, Y., Yan, F., Olson, M., Chen, B., Zhang, H., El-Sana, J.: Automatic reconstruction of tree skeletal structures from point clouds. ACM Trans. Graph. **29**(151), 18 (2010)
14. Jaakkola, A., Hyyppa, J., Kukko, A., Yu, X., Kaartinen, H., Lehtomaki, M., Lin, Y.: A low-cost multi-sensorial mobile mapping system and its feasibility for tree measurements. J. Photogrammetry Remote Sens. **65**, 514–522 (2010)
15. Habel, R., Kusternig, A., Wimmer, M.: Physically guided animation of trees. Eurographics **28**(2), 523–532 (2009)
16. Akagi, Y., Kitajima, K.: Computer animations of swaying trees based on physical animation. Comp. Graph. **30**(4), 529–539 (2006)
17. Chuang, Y.Y., Goldman, D.B., Zheng, K.C., Curless, B., Salesin, D.H., Szeliski, R.: Animating pictures with stochastic motion textures. ACM Trans. Graph. **24**(3), 853–860 (2005)
18. Zhang, L., Zhang, Y., Jiang, Z., Li, L., Chen, W., Peng, Q.: Precomputing data-driven tree animation. Comp. Anim. Virtual Worlds **18**(4–5), 371–382 (2007)
19. Fugmann, H., Schindler, D., Mayer, H.: A dynamic tree sway model. In: 2nd International Conference Wind Effects on Trees, pp. 269–275 (2009)
20. Newson, T., Sagi, P., Miller, C., Mitchell, S.: Application of yield surfaces in three-dimensional VHM load space to the stability of trees under wind loading. In: 2nd International Conference Wind Effects on Trees, pp. 141–147 (2009)
21. Peltola, H., Kellomaki, S.: A mechanistic model for calculating windthrow and stem breakage at stand edge. Silva Fennica **27**, 99111 (1993)
22. Seidl, R., Rammera, W., Blennow, K.: Simulating wind disturbance impacts on forest landscapes: tree-level heterogeneity matters. Environ. Model. Softw. **51**, 111 (2014)
23. Hale, S.E., Gardiner, B.A., Wellpott, A., Nicoll, B.C., Achim, A.: Wind loading of trees: influence of tree size and competition. Eur. J. For. Res. **131**, 203217 (2012)
24. Seidl, R., Rammer, W., Scheller, R.M.: Spies TA an individual-based process model to simulate landscape-scale forest ecosystem dynamics. Ecol. Model. **231**, 87100 (2012)
25. Seidl, R., Spies, T.A., Rammer, W., Steel, E.A., Pabst, R.J., Olsen, K.: Multi-scale drivers of spatial variation in old-growth forest carbon density disentangled with Lidar and an individual-based landscape model. Ecosystems **15**, 13211335 (2012)
26. Bunting, P., Armston, J., Lucas, R.M., Clewley, D.: Sorted pulse data (SPD) library. Part I: a generic file format for LiDAR data from pulsed laser systems in terrestrial environments. Comput. Geosci. **56**, 197206 (2013)
27. Favorskaya, M., Zotin, A., Danilin, I., Smolentcheva, S.: Realistic 3D-modeling of forest growth with natural effect. In: Phillips-Wren, G., Jain, L.C., Nakamatsu, K., Howlett, R.J. (eds.) Advances in Intelligent Decision Technologies in Smart Innovation, Systems and Technologies, p. 191199. Springer, Heidelberg (2010)

28. Runions, A., Lane, B., Prusinkiewicz, P.: Modeling trees with a space colonization algorithm. In: Eurographics Workshop on Natural Phenomena, p. 6370 (2007)
29. de Lima Bicho, A., Rodrigues, R.A., Musse, S.R., Jung, C.R., Paravisi, M., Magalhaes, L.P.: Simulating crowds based on a space colonization algorithm. Comp. Graph. **36**(2), 7079 (2012)
30. Favorskaya, M., Zotin, A., Chunina, A.: Procedural modeling of broad-leaved trees under weather conditions in 3D virtual reality. In: Tsihrintzis, G.A., Virvou, M., Jain, L.C., Howlett, R.J. (eds.) Intelligent Interactive Multimedia Systems and Services in Smart Innovation, Systems and Technologies, p. 5159. Springer, Berlin (2011)
31. Perlin, K.: An image synthesizer. ACM SIGGRAPH **19**(3), 287296 (1985)
32. Singh, P.A., Zhao, N., Chen, S.C., Zhang, K.: Tree animation for a 3D interactive visualization system for hurricane impacts. In: IEEE International Conference on Multimeia and Expo, ICME, p. 598601 (2005)

Chapter 11
Extending the Service Oriented Architecture to Include a Decisional Components

Fatima Boumahdi and Rachid Chalal

Abstract The emergence of Service-Oriented Architecture (SOA) as an approach for integrating applications that expose services, presents many new challenges to organizations that ensure the decisional aspect of company by adopting the SOA like a support architecture. The proposed architecture will facilitate research and development of Decision-Making and decision support systems in SOA. This chapter describes Service Oriented Architecture with decisional aspect (SOAda). To define a new set of concepts necessary for modeling the three views: business, information and decision, a new meta-model called Decisional Model of Service (DMoS) will be presented. Some of these concepts are already known, whereas others are new and are proposed as an element of this chapter. The architecture leading elements are: the DMoS meta-model and the new architecture SOAda. Finally, the proposed architecture will be illustrated with a case study in inventory management.

Keywords Inventory management · Decision-making · Decisional model of service · Service-oriented architecture · SoaML

11.1 Introduction

While organizations are trying to become more agile to better respond markets changes, and in the midst of rapidly globalizing competition, they are facing new challenges. It is primarily a question of ensuring the decisional aspect of the information system by adopting Service-Oriented Architecture (SOA) like a support architecture.

F. Boumahdi (✉)
LRDSI Laboratory, Sciences Faculty, Saad Dahlab University,
BP 270 Soumaa Road, Blida, Algeria
e-mail: f_boumahdi@esi.dz

R. Chalal (✉)
Laboratory LMCS, ESI Wadi-Smar (Algiers), Algiers, Algeria
e-mail: r_chalal@esi.dz

J.W. Tweedale et al. (eds.), *Intelligent Decision Technology Support in Practice*, Smart Innovation, Systems and Technologies 42,
DOI 10.1007/978-3-319-21209-8_11

For example, inventory management comprises business activities which are connected with decision activities like:

- Decide whether to have inventory or ordered bound items: A company can either have articles in inventory or place their own order when they receive an order from the customer, order bound articles.
- Decide order quantity: Order quantity is the amount that an article should be purchased in.

This engineering must ensure on the one hand the flexibility of the information system and on the other hand, avoids the redevelopment of system at the request of the decisions. Several research tasks were interested in the problem of services identification [1–11]. A key limitation of those methodologies is that they are limited to the business and information level for SOA design, and they don't determine the way to integrate a decisional aspect in this type of architecture. The decisional vision of the enterprise is considered like a sub process, and it is left as a not developed black box. Hence, it is necessary to define guidelines that are applicable to find and realize decision services.

The main goal of this research is to design a new architecture to support the Decision-Making in a SOA. To reach this goal, several research questions are formulated. This chapter begins by the literature study, in which the topics of SOA and SOA design are discussed and the concept of Intelligent Design Choice (IDC) is explained. This leads to a new architecture and the first research question is:

1. What service types can be identified based on state of the art literature sources? The new architecture put the Metamodal in the second part of this research. New concepts were introduced into Service Oriented Architecture with decisional aspect (SOAda). It is the role of the meta-models which will be presented after. The second research question is:
2. What are the various service types (business, information, software and decisional) which exist in enterprise?

    - What is the relation between the services and the business processes?
    - Which level of granularity is necessary to define for the services in order to ensure an easy management and a maximum reuse?

    In the third part of this research, a new architecture is described. An extended SOA for supporting a decisional aspect. The third research question is:
3. What are the different layers constituting SOAda? and How to assign each services defined in to SOA layers?

The proposed SOAda integrates elements into two dimensions: The conceptual dimension aims to define concepts and relations between them to be used all over the SOAda. New concepts will be proposed about the SOAda in particular the concept of decision, intelligence, design and the choice service. In this chapter, a whole of concepts will be defined, that constitute SOAda as well as the relationship between them. It is the role of the meta-models. The main objectives for the inclusion of a meta-model in the SOAda proposal are: to define, organize and reuse knowledge about concepts involved in the business processes and their decisional aspect, as well as their design and implementation based on services and the relationship between them.

The architectural dimension will be proposed into SOAda. In search of agility, the SOA must exceed the technical framework related on business and information sight to touch the decisional level of the enterprise. In this direction, the true challenge consists in extending the SOA to the decisional aspect of the company. For the SOAda, an architecture that contained a decisional aspect is proposed.

The rest of the chapter is structured as follows. Section 11.2 describes and discusses the existing approaches of SOA. Section 11.4 presents the concepts used in the IDC approach. Section 11.5 discusses the different levels of the proposed architecture. Section 11.6 describes a case of study and the last section describes the future direction for this line of research.

11.2 State of the Art

The classification of services is not standardized [12]. Hence, different authors propose different classification schemas based on the scope of their proposed approaches. These works will be briefly presented according to a chronological order. The Service-Oriented Analysis and Design (SOAD) is an approach improved and interdisciplinary of service modeling, suggested by Zimmermann [11], on the basis of existing development and notations processes. Service-Oriented Modeling and Architecture (SOMA) illustrates the activities of a modeling method based on services, proposed by Arsanjani et al. [2]. For the identification and the specification of service, it combines the three approaches of ascending, downward analysis and middle-out.

A conceptual model, presented by Kim and Yun [4] and called M4SOD (Method for Service-Oriented Development), the goal is to formalize the development process SOA. This method explain the phases of identification and realization of the services. An approach of modeling and designing systems based on SOA is proposed by Rahmani et al. [5], which uses the architecture directed by Model Driven Architecture (MDA).

Service-Oriented Unified Process (SOUP), a development process intended for the system based on SOA and suggested by Mittal [10], reuse the best elements of RUP and XP (Extreme Programming). Chaari et al. [9] proposes the approach of the Services Oriented Company which treats the problem of collaboration between firms. Lemrabet et al. [13] treats the same problem.

The recent approaches are developed at the base of the Service oriented architecture Modeling Language (SoaML) [14]. Currently three equipped methods use this Service-OrientedModeling and Architecture (SOMA) language for services modeling. The first method was proposed by Amsden [1], it was integrated in Sect. 2.9

version of the SOMA method proposed by Arsanjani et al. [2] and supported by Rational Software Architect (RSA). The second method is proposed by Casanav [15] and is supported by the tool ModeDriven. Lastly, the third method is the Model-Based Development with SoaML (MBDS) which is proposed by Elvester et al. [16] and is supported by the modeling tool “Modelio” of SOFTEAM. The authors of these three methods take an active part in the SoaML specification, this leads to two alternatives of services modeling in the specification. But there is no work that treats the decisional problem.

Furthermore, some decision support systems based on SOA were also studied and they are published in the literature. The first work is that proposed by Xu et al. [17] whose title is “SOA-based precision irrigation decision support system”, the Decision-Making aid system select the adequate service according to the users needs of the different Business Process Management (BPM), then it set up the model. At the end, it gives the exact instruction to the irrigation. The second architecture is that of Vescoukis et al. [18] who proposes a flexible SOA for decision support systems in environmental crisis management.

SoaDssPm: Service-Oriented Architecture of the decision support system for the Project Management was presented by Boumahdi and Chalal [19]. The proposed architecture is made up of a user interface, Action layers, Technique and Data layer, each layer is made up of two under layers, to represent the business process, other intended for the Decision-Making process. There are other kind of works in the literature, which treat the decisional aspect for the composite service problem, like the work of Fan [20], who proposes a new method of Decision-Making for personalized composite service.

11.3 Discussion

All the approaches of services identification presented in the state of the art are limited to the business and information level for SOA design, there’s no approach or architecture in the literature that proposes a solution for the decisional aspect in this type of architecture.

The most important difference between the work introduced above and the work presented here is that none of the related works describes how the decisions were identified and designed in SOA. This chapter identify an initial set of new service classification and also describe the structured architecture used to solicit, classify and consolidate the service found in the SOA with a decisional aspect.

Another difference between the works above and this work is the way that the service have been classified: for instance, in the method developed by Papazoglouand and Van Den Heuvel [8], services are placed in a hierarchy. Business processes are translated to business services, which are converted to infrastructure services [8]. In the hierarchy proposed by Papazoglouand and Van Den Heuvel [8], only two types of services are distinguished, business and infrastructure services. In De Castro et al.

[21] services are classified into two views: Business, and Information system view; the SOAda contributes a new perspective by classifying service according to three views: business, information and Decision.

To reach the goals of a successful SOA implementation, enterprises need to reconsider how they make provision for decision services. If they continue with the unmanageable black-box approach of wrapping decision, they will not attain the rewards of SOA innovation. Instead of flexibility and reuse, they will get only additional layers of complexity. Enterprises need to move the decisional aspect to a new SOA architecture which leverages with decision services.

The goal of this work is to extend the SOA architectures by a decisional dimension, and always keep the business and informational level of the enterprise. For this purpose, a new SOA model based on decision model is proposed in this chapter: IDC of Simon [22], named Decisional Model of Service (DMoS).

The originality of this proposal is due to the simultaneous use of the SOA and the decision model IDC. The literature does not offer examples of coupling these two types of reality representations. The proposed architecture in this chapter is based on IDC model of Simon [22] to propose an innovating architecture which deals with the decisional sight of the business process.

11.4 The Decision-Making Processes in the Organizations

Business decisions are made in the process of conducting business to achieve its objective in a given environment. According to the work of Al-Zhrani [23], many models were developed, in order to represent the various phases of the Decision-Making process. The most known model is that proposed by Simon [22], called the IDC model, which is composed of the three following phases: Intelligence (comprehension), Design and Choice. A fourth phase called "Review or implementation" was added with the model (Fig. 11.1). These phases do not proceed in a sequential way; they are the subject of flashbacks, adjustments, corrections, resumptions of former results, etc.

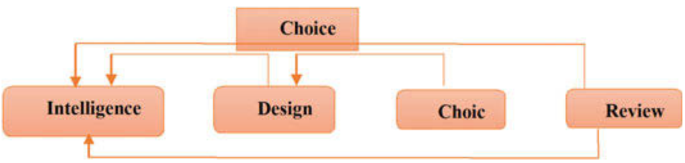


Fig. 11.1 Model IDC of the decision-making process [22]

11.5 Dimensions of SOAda

The SOAda integrates elements into two dimensions: conceptual and architectural. The conceptual dimension provides the basis for understanding all proposed concepts through the meta-model; the architectural dimension contains the different proposed layers of the new SOAda architecture.

11.5.1 Conceptual Dimension

SOAda defines a new whole of concepts necessary for modeling the three points of view: business, information and Decision. Some of them are already known, while others are new and are proposed within the framework of this work. By means of a class diagram, Fig. 11.2 shows the various concepts on which SOAda and their relations are based.

Figure 11.2 puts forward the principal concepts of SOAda architecture. The central element of this meta-model is the service. This generic concept, on which all the principle of SOAda is based, is refined in three under concepts which are the business, information and decisional services. The Business and Information sights are inspired from the works of [2, 6, 8, 9, 11, 15, 21]. The contribution consists on the proposal of the third sight which is: Decision.

The key concept of the business aspect of SOAda is the *Business Service*. A Business service is a reusable brick on a business level. It corresponds on trade functionalities. It calls upon functional services, these last are services of average granularity such as for example the services offering the functionalities related on the management of a customer order, or the calculation of an order amount. In fact, these services expose the concept of the information system function that allow the implementation of the trade activity concept. Example: The list of the orders of a customer on standby; the state of its accounts.

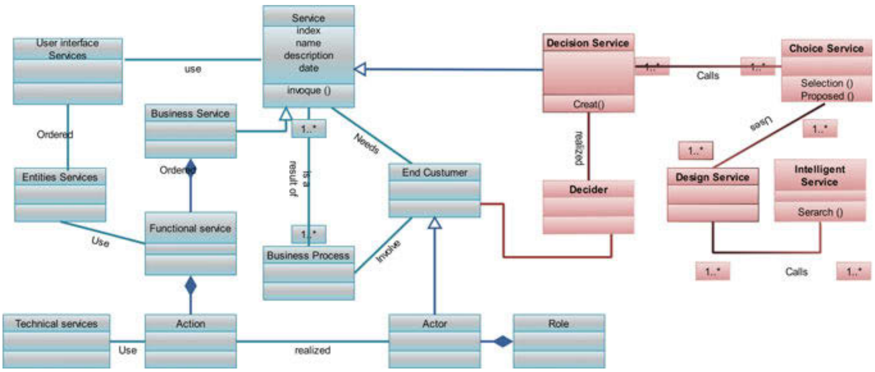


Fig. 11.2 Model decisional of the service (MDS)

The second part of the meta-model of the SOAda relates the *information vision* of SOAda. The information services gather the functionalities offered by the computerized part of the information system and are used by the functional services described previously. Three categories of information system departments are distinguish, which are: the entities services, user interface services of and technical services.

User interface Services: The user interface services allow to manage the communications and the dialog with the actors.

Entities Creation, Reading, Updating and Deleting (CRUD) Services: The entities services are the services which are focused on key trade objects of the information system of the company. They are primarily the services which give access to the information relative to the trade objects. Typically, they are the services which carry out operations of CRUD.

Technical services: The technical services gather the services from the low level and allow to manage the infrastructure of the information system of the company (example: services related on the network management, emailing and desktop publishing). The role of a technical service is to give access to a given technical resource: for example the access to the relational databases, printer and emailing.

As for the third part of the meta-model that relates the suggested *decisional dimension*. The decision services gather functionalities offered by the decisional part of the business process and they are used by the services of company described previously.

Three service types are proposed in this sight according to model which are: intelligent, design and choice [22]. These include:

1. *Intelligent service*: intended for the definition of the problem to be solved to reach the objectives or the goals of the decision maker. An intelligent service is a service that seeks the relevant information in relation with the problem to be solved, by using research algorithms (k-Nearest Neighbors algorithm (k-NN), decision trees, …).
2. *Design service*: consists in building and studying solutions, using the information that results from the intelligent service. According to the technique of Decision-Making, this service type is conceived.
3. *Choice service*: they have as a function to select one of the solutions elaborated and suggested by the design services, this one becoming the result of the decision. The Multi-Criteria Decision methods can be used [24] (such as ELECTRE family [24], or PROMETHEE [24]) to conceive the services of choice.

11.5.2 Architectural Dimension

According to SOAda, the company becomes a conglomeration of services with various types of abstraction. Indeed, the proposal is based on the definition of services on three different levels (Fig. 11.3):

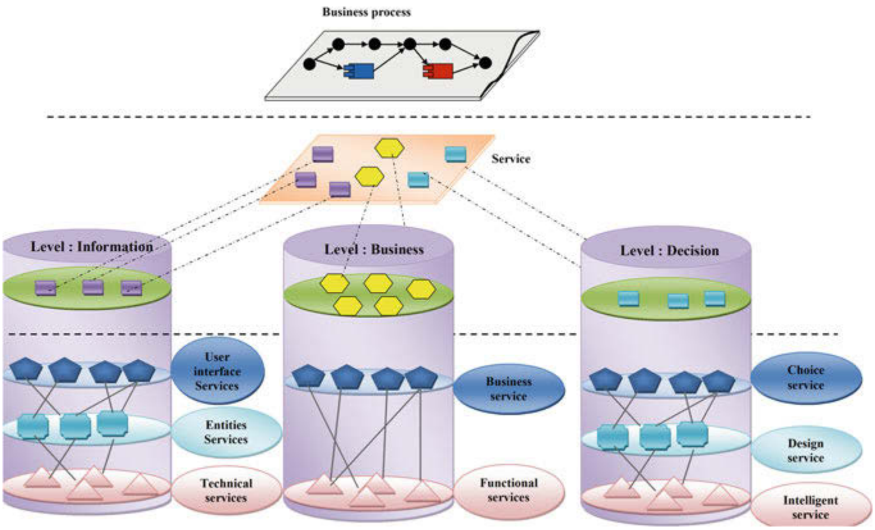


Fig. 11.3 SOAda architecture

Business Level: it concentrates on the characteristics and the requirements of the company in which the Information System (IS) will be built. The processes business of the company will be built following an orchestration of the various services of company. This layer is responsible for managing the overall execution of the business process.

Information Level: it concentrates on the functionalities and the processes which must be implemented on IS in order to satisfy the needs of the company. It results from the construction of a technical SOA within the company. It is about the whole of the data processing services of the company which support the business level.

Decision Level: present the Decision-Making services of sub process. This sight is based on IDC model of Simon [22].

11.6 Case study

The purposes of the inventory management choice are:

- The management of inventory is crucial to the success of most companies,
- The potential major role of decision in the inventory management,
- Inventory management comprises business activities which are connected with the decision activities.

The Decision-Making in inventory management must resolve mainly three basic issues: How often the inventory status should be determined? When a replenishment order should be placed? and How large the replenishment order should be [25]?

Inventory management has three major parts [25]:

Forecasting Inventory Demand: Forecasting calculates how much inventory is needed and when it will be needed. Its primary goal, is to have the right products in the right positions at the right time [25].

Inventory Control: Inventory control does the basic record keeping that is the basis of which more complex decisions are made in the forecasting and replenishment modules. It involves the creation of inventory records, the practices dealing with the maintenance of these records, and the counting or auditing of inventory. It also deals with inventory administration, methods of valuating inventory, and evaluating inventory performance.

Replenishment of Inventory: The Economic Order Quantity (EOQ) model is one technique used to determine the optimal quantity to order. The optimal ordering quantity, EOQ, can be found analytically. It occurs where the total annual holding cost equals the total annual ordering cost [25].

The next section explain how the SOAda works and what kinds of problems it is useful to be solved. The Fig. 11.4 shows the business process of inventory management with the language Business Process Management Notation (BPMN) [26].

Service granularity (Gra) defines how much functionality is exposed by one single service. Services can be fine grained or coarse grained (Papazoglouand and Van Den Heuvel [8]). The Table 11.1 represents a summary of the different services that are proposed by the SOAda architecture and their specificities. The granularity of service is expressed on a scale of 4. On this scale the business service is a high granularity (4/4). It also shows the inventory management services according to SOAda.

The architecture shown in the Fig. 11.5, supports Decision-Making inventory manager by SOAda.

The most promising and complete SOA modeling language is SoaML [14]. In the recent future, it is expected that SoaML becomes more and more adopted as a standard modeling language for SOA [27, 28]. This language is used for services modeling in inventory management.

The mapping from BPMN diagram to SoaML model requires first modeling language (SoaML) and foremost a correspondence between the elements of BPMN and SoaML elements. For this reason, the mapping defined in the research work already done, will be used [1, 15, 27, 29]. The Table 11.2 summarizes the implementation and modeling of inventory management.

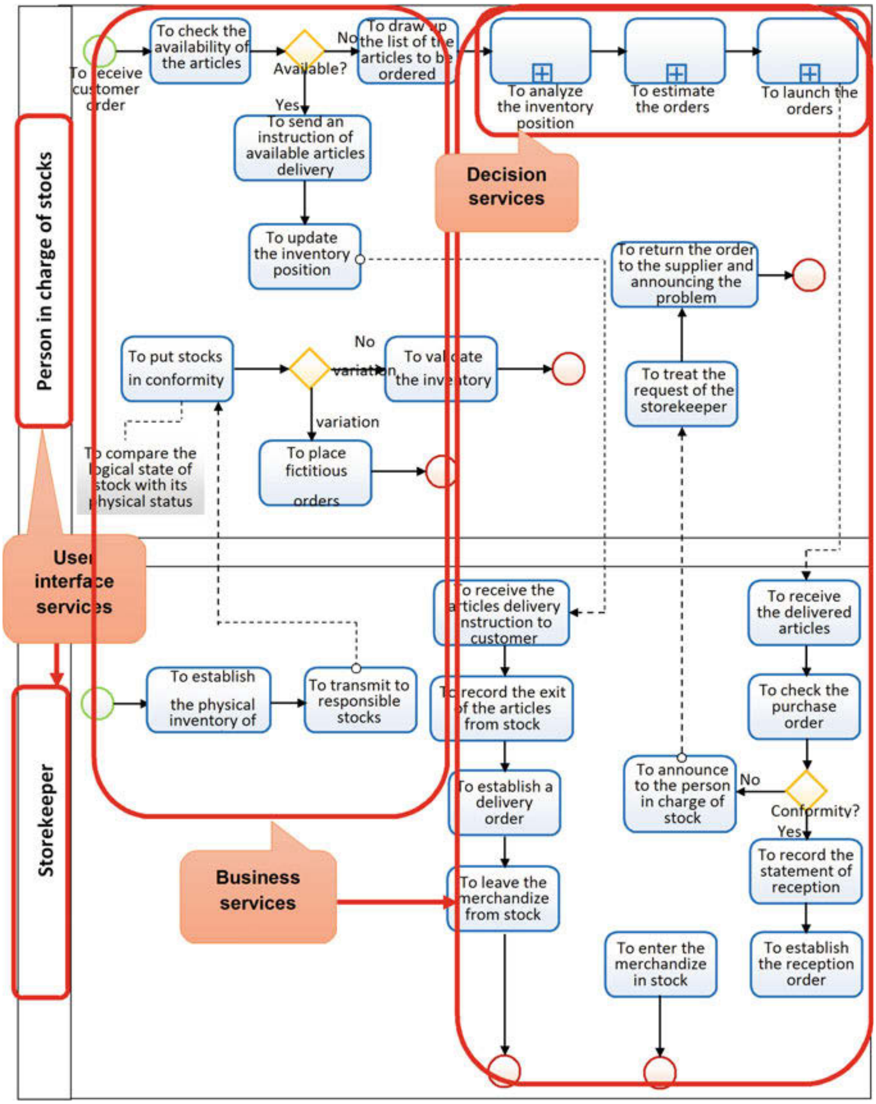


Fig. 11.4 The inventory management process

Table 11.1 Summary of the various SOAda services and their characteristics

Service	Description	Interaction	Gra.	Composition	Inventory management
Business level					
Business	– Abstract service –Participate in business processes	Synchronous	4/4	Composite	– Run commands –Checks the items ordered –Manager service
Functional	– Abstract –Orchestra several information and decisions services	Synchronous	3/4	Composite	–Report a problem –Receipt of the order –Delivery order –Department service
Information level					
User interface	– Concrete –Encapsulates the business applications	Synchronous	3/4	Composite	–Storekeeper Interface –Responsible Interface –Inventory –Costumer
Entities	– Concrete –Exposed by a business object	Asynchronous	2/4	Atomic	–Order –Provider –Product –Delivery
Technical	– Concrete –Service infrastructure of IS	Asynchronous	1/4	Atomic	–Resource access –Messaging Service

(continued)

Table 11.1 (continued)

Service	Description	Interaction	Gra.	Composition	Inventory management
Decision level					
Decision	– Concrete –Participate in business processes	Synchronous	4/4	Composite	–Forecasting service –Inventory Control service –Replenishment service –Safety stock
Choice	– Concrete –Expose the several choices	Synchronous	3/4	Composite	–Optimal cycle length –Optimal total cost –Optimal order quantity
Design	– Concrete –Expose the solution method	Asynchronous	2/4	Atomic	–Pareto method –ABC method –Wilson model
Intelligent	– Concrete –Includes the requirements of decision	Asynchronous	1/4	Atomic	–RC = reorder cost –HC = holding cost –T = cycle length – UC = unit cost –Q = order quantity

11.7 Contribution

The SOAda architecture assists the decision maker, without replacing him, during the Decision-Making process. It allows to the decision maker to have access to many knowledge, to synthesize them and test various possible choices.

The various contributions carried out in this work are summarized as followed:

Introduction of the decision aspect into SOA: where SOA intervens mainly on the information level of the company. The majority of work which treats the service orientation looked at the problem from a technical point of view and

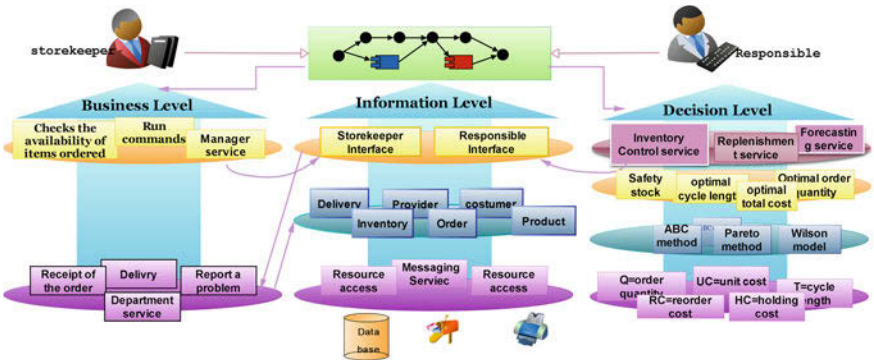
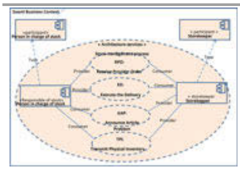
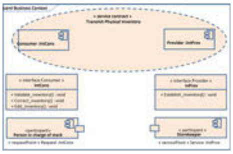
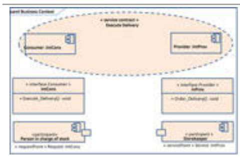
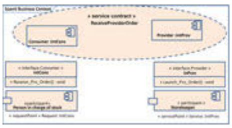
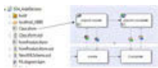


Fig. 11.5 SOAda applied to inventory management

Table 11.2 The modeling and realization of inventory manager

Modeling (SoaML)	Realization	Modeling (SoaML)	Realization
			
			

their starting point was always the information and business level. The developed SOAda extend the principles of SOA on the totality of the company system. It brought new concepts and it restructure the company architecture in a manner that it is nimbler and able to take part in decisions from a request. The various concepts introduced by SOAda were detailed by presenting the meta-model.

SOAda: extend SOA with decisional aspect: There is no SOA that integrates the decisional dimension. The majority of existing works deal on an industrial scale which does not present the decisional aspect. This contribution consists in presenting a new construction architecture of SOA taking account the needs of decision of its various elements. The developed architecture is based on IDC model of Simon who differentiates the proposal from the existing architectures.

11.8 Conclusion and Future Work

For the last few years, a rise has been observed in research activity in SOA, with applications in different sectors. Several new technologies have been introduced and even more are being currently researched and aimed to the future. To reach the goals of a successful SOA implementation, enterprises need to reconsider how they provide decisional aspect. This chapter puts forward one novel idea and architecture about how enterprises move to a new SOA which leverages with Decision aspect. Throughout this chapter the concept of the SOAda was developed. The basic principle of the enterprise appears by triple vision Business/Information/Decision. In this direction, the SOAda is the juxtaposition of three models: business SOA for the business view, information SOA for the information view and decision SOA for the decisional vision of the company. This duality guaranteed the alignment of the Information System on the business and the decisional needs and thus allow to the enterprise to gain in agility. Thus, an application of the proposal is shown to illustrate how to use the architecture. In the future works, the integration of decisional aspect in Cloud computing is envisaged.

References

1. Amsden, J.: Modeling with SoaML, the service-oriented architecture modeling language. Technical report, IBM (2010)
2. Arsanjani, A., Ghosh, S., Allam, A., Abdollah, T., Ganapathy, S., Holley, K.: SOMA: a method for developing service-oriented solutions. IBM Syst. J. **47**(3), 377–396 (2008)
3. Börner, R., Goeken, M.: Identification of business services. In: 15th Americas Conference on Information Systems (AMCIS), San Francisco, California (2010)
4. Kim, Y., Yun, X.: An approach to modeling service-oriented development process. In: IEEE International Conference on Services Computing, pp. 273–276. IEEE Press, New York (2006)
5. Rahmani, A.T., Rafe, V., Sedighian, S., Abbaspour, A.: An mda-based modeling and design of service oriented architecture. In: Computational Science–ICCS, pp. 578–585. Springer, Berlin (2006)
6. Erl, T.: Service-oriented architecture: concepts, technology, and design. Prentice Hall, Upper Saddle River (2005)
7. Berkem, B.: From the business motivation model (BMM) to service oriented architecture (SOA). J. Object Technol. **7**(8), 57–70 (2008)
8. Papazoglou, M.P., Van Den Heuvel, W.J.: Service-oriented design and development methodology. Int. J. Web Eng. Technol. **2**(4), 412–442 (2006)
9. Chaari, S., Biennier, F., Favrel, J., Benamar, C.: Towards a service-oriented enterprise based on business components identification. In: Enterprise Interoperability II, pp. 495–506. Springer, Berlin (2007)
10. Mittal, K.: Build your SOA, part 3: the service-oriented unified process. Technical report, IBM developer Works (2006)
11. Zimmermann, O.: An architectural decision modeling framework for service-oriented architecture design. Ph.D. thesis, Stuttgart, University, Dissertation (2009)
12. Klose, K., Knackstedt, R., Beverungen, D.: Identification of services-a stakeholder-based approach to SOA development and its application in the area of production planning, pp. 1802–1814 (2007)

13. Lemrabet, Y., Clin, D., Bigand, M., Bourey, J.: From bpmn 2.0 to the setting-up on an esb–application to an interoperability problem. In: Collaborative networks for a sustainable world, pp. 722–729. Springer, Berlin (2010)
14. OMG: Service oriented architecture modeling version 1.0.1. Technical report, OMG. http://www.omg.org/spec/SoaML/1.0.1/PDF (2012)
15. Casanave, C.: Enterprise service oriented architecture using the omg soaml standard, a model driven solutions. http://ModelDriven.org. Technical report (2012)
16. Elvesæter, B., Berre, A.J., Sadovykh, A.: Specifying services using the service oriented architecture modeling language (SoaML)-a baseline for specification of cloud-based services. In: CLOSER, pp. 276–285 (2011)
17. Xu, L., Chen, L., Chen, T., Gao, Y.: Soa-based precision irrigation decision support system. Math. Comput. Model. **54**(3), 944–949 (2011)
18. Vescoukis, V., Doulamis, N., Karagiorgou, S.: A service oriented architecture for decision support systems in environmental crisis management. Future Gener. Comput. Syst. **28**(3), 593–604 (2012)
19. Boumahdi, F., Chalal, R.: Soadsspm: a new service-oriented architecture of the decision support system for the project management. In: International Conference on Control, Engineering and Information Technology (CEIT13), Sousse, Tunisie, pp. 6–10 (2013)
20. Fan, X.Q.: A decision-making method for personalized composite service. Expert Syst. Appl. **40**(15), 5804–5810 (2013)
21. De Castro, V., Marcos, E., Vara, J.M.: Applying cim-to-pim model transformations for the service-oriented development of information systems. Inf. Softw. Technol. **53**(1), 87–105 (2011)
22. Simon, H.A.: Administrative Behavior, vol. 3. Cambridge University Press, Cambridge (1976)
23. Al-Zhrani, S.: Management information systems role in decision-making during crises: case study. J. Comput. Sci. **6**(11), 12–47 (2010)
24. Vincke, P.: Multi-criteria decision-aid. Wiley, Chichester (1992)
25. Silver, E.A., Pyke, D.F., Peterson, R., et al.: Inventory management and production planning and scheduling, vol. 3. Wiley, New York (1998)
26. OMG: Business process modeling notation (BPMN). Technical report, OMG. http://www.omg.org/spec/BPMN/2.0/PDF (2011)
27. Todoran, I., Hussain, Z., Gromov, N.: Soa integration modeling: an evaluation of how soaml completes uml modeling. In: 15th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW), pp. 57–66. IEEE Press, New York (2011)
28. Berre, A., Roman, D., Elvesaeter, B., Carrez, C.: Service oriented architecture modeling language (SoaML). Technical report (2010)
29. Elvesaeter, B., Panfilenko, D., Jacobi, S., Hahn, C.: Aligning business and it models in service-oriented architectures using bpmn and soaml. In: Proceedings of the First International Workshop on Model-Driven Interoperability, pp. 61–68. ACM Press, New York (2010)

Chapter 12
An Extended Dependability Case to Share Responsibility Knowledge

T. Saruwatari, T. Hoshino and S. Yamamoto

Abstract Recently, critical incidents have occurred in complex Information Technology (IT) systems. Thus, how to confirm the dependability of a system using dependability cases is becoming necessary. Information related to dependability is important knowledge that must be shared among stakeholders. However, in the previous methods used to describe dependability cases, the relationship between a dependability claim and responsibility cannot be clearly specified. Thus, the cause investigation cannot be completed at the occurrence of the incident, since system knowledge could not fully be utilized. In this chapter, the d* framework is proposed to define the responsibility attributes for sharing knowledge and achieving agreements among stakeholders. The d* framework extends the dependability case to add an agent and an actor to the dependability case representing the responsibility attribute. A Meta model for the extended dependability case is also shown. Moreover, to show the effectiveness of the d* framework, three example applications are described.

Keywords Reliability · Dependability case · d* framework

T. Saruwatari (✉)
Graduate School of Information Science, Nagoya University,
Furo-cho, Chikusa-ku, Nagoya, Japan
e-mail: saruwatarit@nttdata.co.jp

T. Saruwatari · T. Hoshino
Software Innovation Center NTT,
NSS-II Bldg. 6F, 2-13-34, Konan, Minato-ku, Tokyo, Japan
e-mail: hoshino.takashi@lab.ntt.co.jp

S. Yamamoto
Strategy Office, Information and Communications Headquarters, Nagoya University,
Furo-cho, Chikusa-ku, Nagoya, Japan
e-mail: yamamotosui@icts.nagoya-u.ac.jp

J.W. Tweedale et al. (eds.), *Intelligent Decision Technology Support in Practice*, Smart Innovation, Systems and Technologies 42,
DOI 10.1007/978-3-319-21209-8_12

12.1 Introduction

Recently, critical incidents have occurred in complex Information Technology (IT) systems. The assurance of dependability in IT systems is becoming increasingly important and should be considered during system implementation. The information related to dependability is important knowledge that must be shared among stakeholders. Therefore, the use of dependability cases is attracting attention as a method of sharing between stakeholders and a confirmation of system dependability. However, in previous dependability cases that describe methods represented by Goal Structuring Notation (GSN), the relationship between the dependability claim and the responsibility cannot be clearly specified. Thus, since the cause investigation cannot be completed at the time of the incident, the system knowledge could not fully be utilized. Hence, a method to express a responsibility attribute for sharing information to attain an agreement between stakeholders is proposed. Here the responsibility attribute specifies who takes responsibility of the dependability case. In this study, a concept for incorporating a responsibility attribute into dependability cases is proposed. By incorporating responsibility attribute, accountability of the dependability case by humans can be achieved. For introducing responsibility attributes, the authors considered d* framework as the target method. The d* framework is one of numerous methods for creating dependability cases. After considering the concept, the authors indicate a simple notation for it.

In this chapter, the authors propose an extension of dependability case. Section 12.2 describes the related work. Section 12.3 describes dependability case and d* framework. Section 12.4 introduces responsibility attributes for dependability case. Section 12.5 discusses considerations of the proposed method. Section 12.6 summarizes the chapter.

12.2 Related Work

A dependability cases are used to assure the dependability of the highly reliable critical systems. Dependability cases are described by GSN [1]. Modular GSN [2, 3], D-Case [4], and d* framework [5] are also proposed to describe dependability cases. Modular GSN can be used to systematically develop large dependability cases using modules. Each GSN module includes a GSN. D-Case is an extension of GSN with monitor nodes. A monitor node is a solution node used for monitoring some aspect of the target system. The d* framework is used to describe the inter-dependency of open systems based on D-Case. The meaning of d* framework is the network of dependability.

Deviation analysis [6] and scenario [2] can be used to develop GSN. Keep All Objectives Satisfied (KAOS) provides a system model for requirements engineering based on a goal model [7, 8]. The goal model of KAOS describes satisfaction argument trees. For the bottom goals, agents are assigned to represent that the assigned agent has the responsibility to achieve the specific bottom goal. Although KAOS can show the responsibility of agents, it is difficult to directly describe the inter-dependency relationship among agents. Sommerville et al. [9] defined the responsibility as follows: "A duty, held by some agent, to achieve, maintain or avoid some given state, subject to conformance with organizational, social and culture norms". They also proposed a method to elicit requirements based on the responsibility model. In this chapter, the notion of responsibility attribute for d* framework that describes inter-related dependability cases is clarified.

12.3 Dependability Case

Recently, safety cases, assurance cases, and dependability cases have been attracting significant attention [10–13]. They are prepared to ensure the safety of the given system. In particular, "dependability cases" are documents that are prepared to describe the dependability of a system. Dependability is defined as an integrated concept that encompasses availability, reliability, safety, integrity, and maintainability [14]. By creating a dependability case, the following can be realized:

- Arrangement of dependability information for ease of understanding.
- Confirmation of system dependability.
- Building consensus among stakeholders regarding system dependability.
- Accountability for system behavior.

In many cases, GSN is used to describe the dependability case. GSN was proposed as a graphical notation for safety cases [1]. After that, it was extended to be used as a notation for dependability cases [13]. Currently, there are other notations for dependability cases in addition to GSN. Modular GSN is a notation for modularized dependability cases [2]. Using modular GSN, you can create and arrange dependability cases by using modulus. The d* framework incorporates the concept of actors into the dependability case [5].

12.3.1 The 'd Framework'*

The d* framework is a graphical notation for describing a dependability case; it was proposed to incorporate the actor concept into dependability cases. Here actors are defined as having dependability attributes, such as goals, strategies, solutions, and contexts. This idea is considered with reference to i* [15], a framework for requirement engineering, and the d* framework was proposed by [5].

There are two type dependability cases in d* framework. These include:

1. The whole dependability case that describes responsibility relations that actor depends on other actors (Inter actor dependability case); and

2. The individual dependability case that describes that each actor is dependable (Intra actor dependability case).

12.3.1.1 Notation

The d* framework has a graphical notation that consists of five element types that are described using defined characteristic shapes in dependability cases. These shapes are shown in Fig. 12.1. Each element type is described below:

Actor: is the element type that constitutes a system. It has dependability attributes, such as goals, strategies, solutions, and context.

Goal: is the element type that an actor should satisfy. It may be decomposed into subgoals or substrategies.

Strategy: explains why elements are decomposed into subelements. A goal or strategy can be the object of decomposition and may be decomposed into subgoals or substrategies.

Solution: shows evidence that the supporting goal is satisfied. It is shown in various ways (For example: specifications, test reports, procedure manuals).

Context: is the external information that is required for goals and strategies. For example, you can consider a procedure list that is the goals target as context.

These elements are related to each other in a dependability case such that d* framework has four types of relationships. These are described in Fig. 12.2. The four types of relationships are explained below:

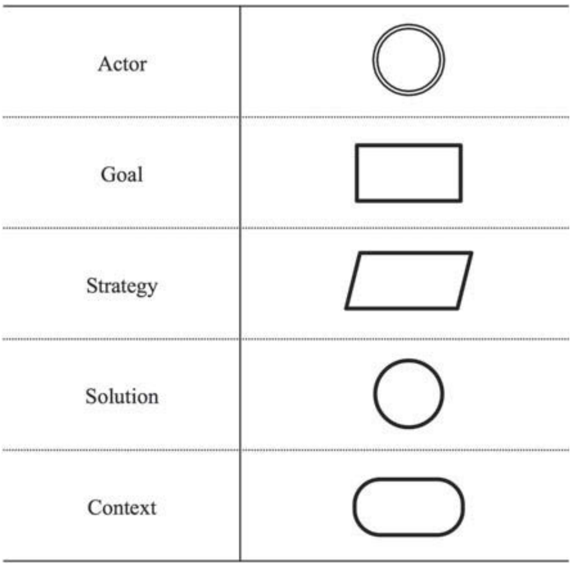


Fig. 12.1 Elements of d* framework

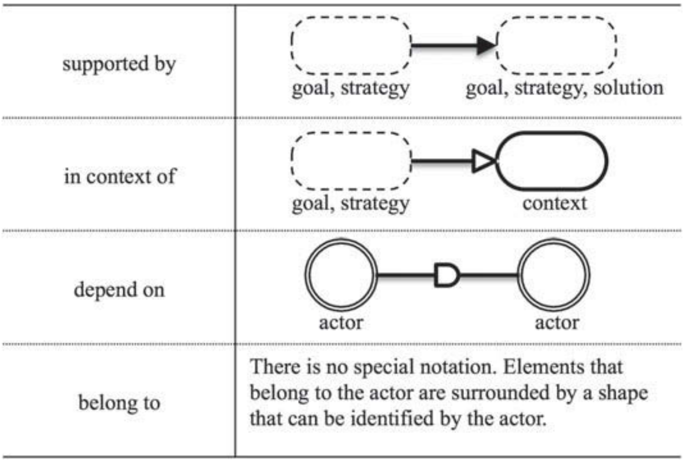


Fig. 12.2 Relationships of d* framework

“supported by”: relationship shows that an upper element is supported by one or more lower elements. Goals and strategies can be upper elements, whereas goals, strategies, and solutions can be lower elements.

“in context of”: relationship shows the addition of a context to goals and strategies.

“depend on”: relationship shows that one actor depends on another actor. When there is a “depend on” relationship between actors A and B, there is generally a “supported by” relationship between the elements of A and the elements of B.

“belong to”: relationship shows that goals, strategies, contexts, and solutions belong to a given actor.

12.4 Introduction of a Responsibility Attribute in Dependability Cases

This section introduces the responsibility attributes used within a dependability case. These include: concepts of responsibility, metal model of responsibility,notation and examples cases. The examples used discuss an elevator system, AP download system and LAN management system.

12.4.1 Concept of Responsibility

In order for a dependability case to have meaning, it is necessary that each element of the dependability case actually be satisfied. That is, someone must take responsibility for each element of the dependability case. More specifically, humans or organizations must take responsibility for each element of the dependability case. The introduction of a responsibility attribute has been carried out by various methods. Lamsweerde proposed an agent model in a requirement engineering method [7, 8], which includes the concept of responsibility. Sommerville et al. proposed a responsibility model for eliciting information requirements and socio-technical risk [9, 16, 17]. Feltus et al. proposed a responsibility model for verifying organizational structure and detecting policy problems [18]. Boness and Harrison proposed a responsibility model for appraising the intention of a development [19]. Strens et al. proposed a responsibility model for identifying and specifying requirements [20]. Responsibility attribute concepts are listed in Table 12.1.

Although there is a marginal difference between these concepts, five proposals resemble each other. It is discovered that these concepts have a basic pattern of responsibility relationships. That is, there are relationships in which an agent takes responsibility for an object. There are two main concepts that are included in a responsibility relationship: (1) an agent and (2) a responsibility object. A responsibility object was expressed as a responsibility or a goal element in the models. Therefore, when the authors consider the incorporation of a responsibility concept into d* framework, two key elements (responsibility and agent) had to be considered. Some of the proposals could use agent as a non-human system. However, the authors define an agent as a human or an organization of humans, because non-human systems cannot take responsibility in a human society.

12.4.2 Meta Model of Responsibility Attribute

In this study, the introduction of responsibility attributes into d* framework is considered. The d* framework is one of numerous methods for describing dependability cases. In d* framework, actor elements are explicitly considered. Here actors are elements of a system and become a dependability subject. A human, an organiza-

Table 12.1 Comparison of proposed responsibility attributes

Proposer	Purpose	Main elements
Lamsweerde	Show the distribution of responsibilities within the system and define system scope and configuration	Goal, agent, operation, object
Sommerville	Explore the structure and dependability of socio-technical systems	Responsibility, resource, agent
Feltus	Verify the organizational structure and detect policy problems	Organization, responsibility, user (agent), task
Boness	Appraise the intention of a development	Goal, actor (agent)
Strens	Identify and specify requirements	Responsibility, resource, agent

tion, a mechanical system, an IT system, etc. can become an actor in a dependability case of d* framework. When considering responsibility attributes in d* framework, it is necessary to consider that non-human objects cannot take responsibilities. More specifically, mechanical and IT systems cannot take responsibility themselves. Therefore, the introduction of the agent concept is necessary to consider responsibility in d* framework. Agent elements and "responsibility for" relationships are added into d* framework. They are defined as follows:

- An agent takes responsibility for elements of a dependability case in d* framework. That is, an agent must have accountability for the contents of elements. A human or an organization consisting of humans can become an agent for actor responsibilities. Further, it is acceptable for the same object to become an actor and an agent simultaneously in d* framework.
- The "responsibility for" relationship shows that elements other than an agent take responsibility held by an agent.

Meta model of d* framework is shown in Fig. 12.3. The model describes elements of d* framework and their relationships. In the model, agent was defined as subclass of actor. ArgumentElement is the super class of goal, strategy, solution, and context. ArgumentElement is used to represent arguments of d* framework. ArgumentElement belongs to actor. ArgumentElement is supported by other argumentElement. ArgumentElement is in context of other argumentElement. Agent has responsibility for argumentElement and Actor. Actor can be a depender and dependee of depend on relationship. ArgumentElement belongs to depend on relationship.

In the elements of Meta model, agent is a new elements to represent the responsibility concept for these model of dependability case. Therefore, responsibility for relationships is defined between agent and other elements.

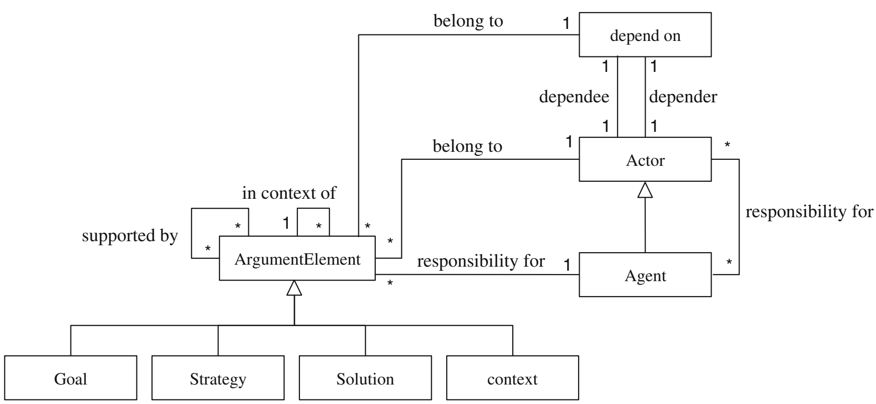


Fig. 12.3 Meta model of d* framework

12.4.3 Notation

In d* framework, a new simple graphical notation for the responsibility attribute is adopted. As an alternative, it was possible to extend the format of statements that explain the elements in a dependability case. The d* framework originally had graphical notation for dependability cases. Therefore, it is natural that the same graphical notation be adopted. In this chapter, the shape enclosed by a dashed line is proposed to describe "responsible for" relationship. The notation for "Agent" and "responsible for" relationship are shown in Fig. 12.4.

12.4.4 Examples

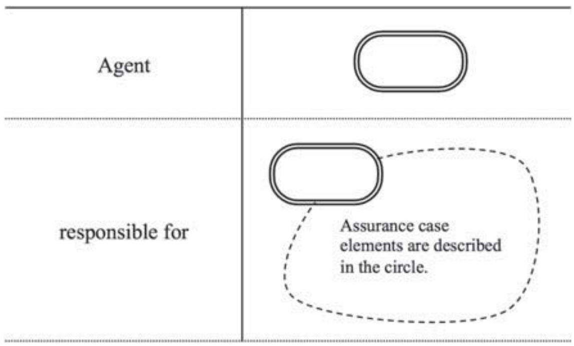
Examples of assurance case for several systems are created. These assurance cases are described by d* framework and include responsibility attribute. These include: Elevator system, AP download system,and LAN device management system.

In the case one (Elevator system), whole dependability case and individual dependability case are shown. In the case of other two systems, whole dependability case is shown.

12.4.4.1 Elevator System

The elevator system is a system for managing of elevators. "Rope", "Door", "User", "Case", "Control panel", "Balance weight", "Rupture detection equipment", "Shock mitigation equipment", "Speed regulator" and, "Lifting equipment" were defined as actors. The "development section", "IT system development section", "maintenance section", "wire company", and "user" were also defined as agents. The dependability case using new notation is shown in Fig. 12.5. In the figure, some actor's responsibilities are supported by a single agent. For example, "Cage" is such an actor. More specifically, the dependability elements of "Cage" are supported by a "development

Fig. 12.4 New element and relationship relationships of d* framework



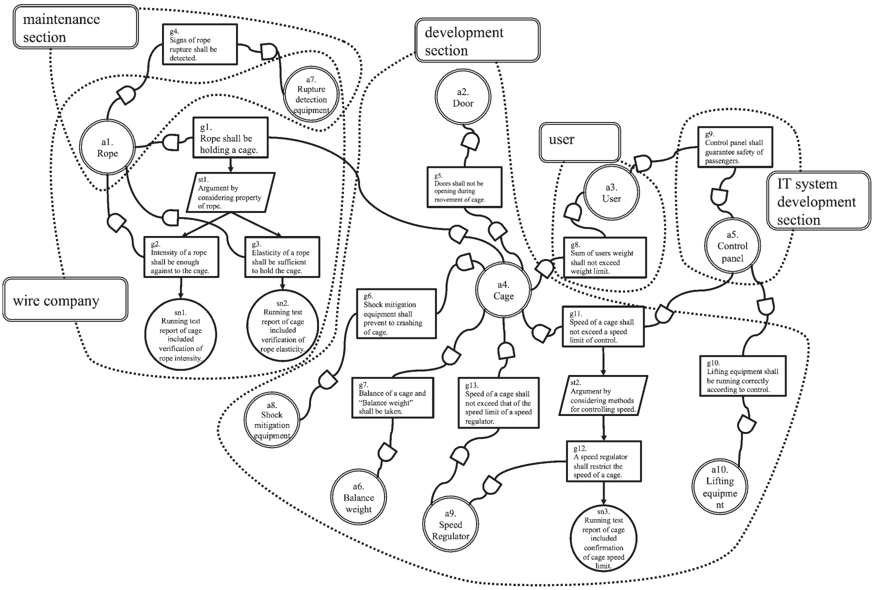


Fig. 12.5 Dependability case example that shows inter dependability between actors (Elevator system)

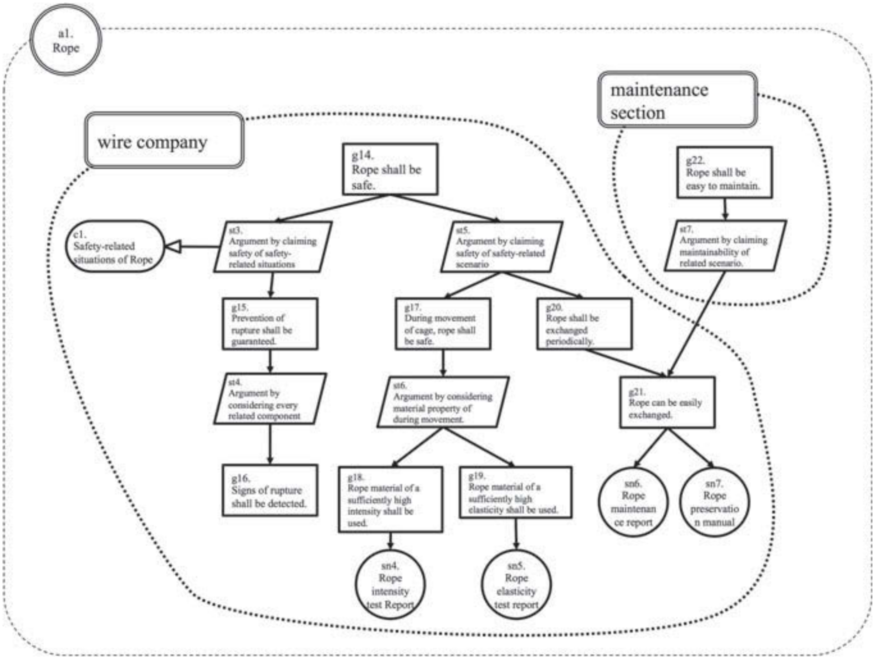


Fig. 12.6 Dependability case example that shows inner dependability of actors (Elevator system: Rope)

section". On the other hand, some actor's responsibilities are supported by multiple agents. For example, "Rope" is such an actor. More specifically, the dependability elements of "Rope" are supported by "maintenance section" and "wire company". In this case, two shapes are overlapped at the actor in the graph.

An example of individual dependability case (that is for inner actor) is shown in Fig. 12.6. Elements of actor are divided and associated with an appropriate agent that has a responsibility for it. All elements of actor must be supported by the appropriate agent. If there is an element that is not supported by any agent, its dependability case is not complete, and the model would be considered as being at the middle phase of development.

12.4.4.2 AP Download System

AP download system is a system for downloading AP from system. Users can download AP from the system to their IC card using the system. "Issuer of cards", "Service provider", "AP download system", "Memory card", and "User" were defined as actors. And, "Issuer of cards", "Service provider", "System deveoper", and "User" were defined as agents. A dependability case for this system is shown in Fig. 12.7.

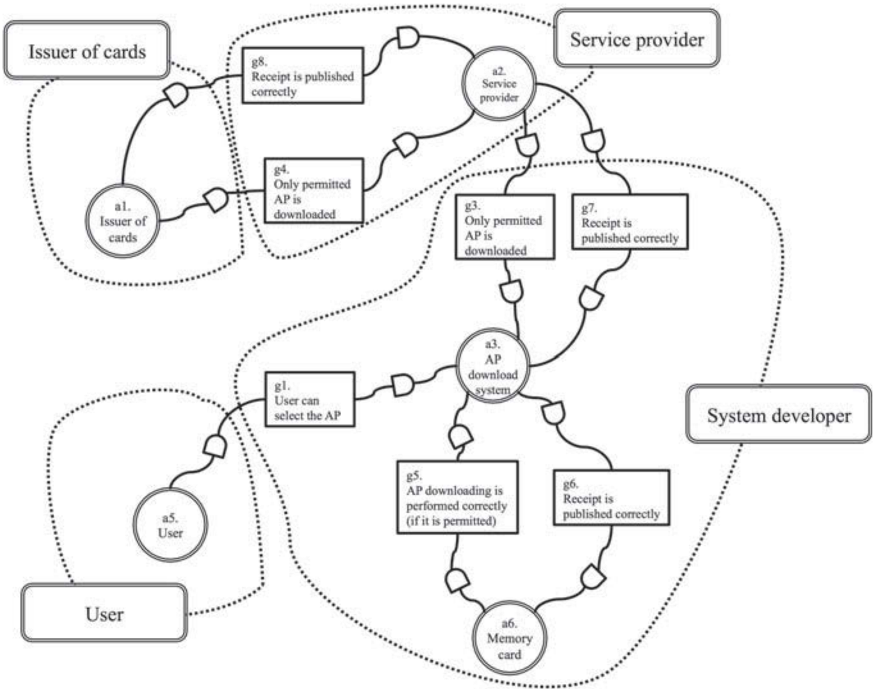


Fig. 12.7 Dependability case example that shows inter dependability between actors (AP download system)

The authors described this dependability case from collaboration diagram. In the collaboration diagram, "system developer" was not defined. Therefore, the authors defined it in the dependability case as an agent. Since "AP download system" is not persons, it cannot take a responsibility. It is still incomplete. Evidences that should support goals are not shown. Further argumentation will be needed. Its dependability case can be used in the argumentation.

12.4.4.3 LAN Device Management System

LAN device management system is system to detect invalid devices. This system intercepts the invalid device. "Surveillance server", "Manager", "Sensor", and "LAN device" were defined as actor. "Service provider", "Sensor vendor", and "User" were defined as agent. A dependability case for this system is shown in Fig. 12.8. This dependability case was also created from collaboration diagram. In this case, all actors of the dependability case were not human. Therefore, human agents that can take responsibility had to be considered. Here, "service provider", "Sensor vendor", and "User" was defined as agents. This dependability case is also still incomplete. More argument of system dependability is required. The dependability case that was shown here will be used for the argument.

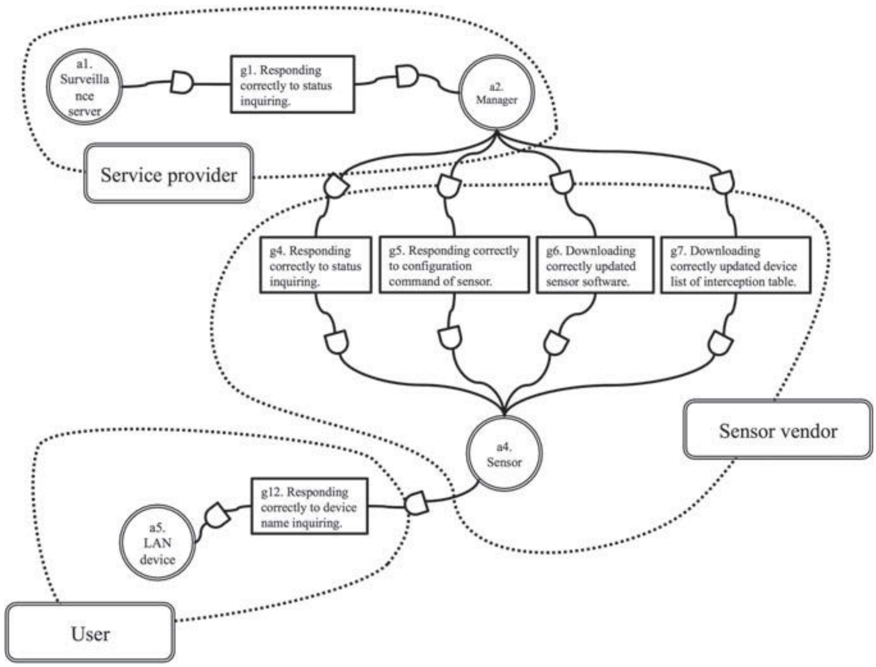


Fig. 12.8 Dependability case example that shows inter dependability between actors (LAN device management system)

12.5 Discussions

This section discusses the organizational structure and dependability cases, the actors and agents, notational extensions for d* framework, the meta model, applicability of the framework and finally any limitations.

12.5.1 Organizational Structure and Dependability Cases

Generally, agents belong to an organization that has a layered structure. In such an organizational structure, responsibility and authority are transferred from upper to lower layers. A dependability case also has a layered structure. In dependability cases, the layered structure is modeled via “supported by” relationships between upper and lower goals. That is, an upper goal is supported by one or more lower goals. Relationships of two elements in the layered structure are divided into two types. Hierarchical order or no hierarchical order. When two agents of an organization have a hierarchical order, the corresponding elements of the dependability case also have a hierarchical order.

In general, it is better if two hierarchical orders are combined. For example, consider a case in which agent A takes responsibility for goal A, agent B takes responsibility for goal B, and goal B is supported by goal A. It is better if agents A and B are the same or if agent A is at a lower layer than agent B. If the hierarchical order is different between the organizational structure and dependability case, a responsibility may be present in the structure of the dependability case, but not in the organizational structure. In such a case, additional action may be required. Checking dependability cases against an organizational structure can identify unsuitable transfers of authority.

12.5.2 Actor and Agent

The concept of an agent resembles the concept of an actor. More specifically, these may indicate the same object, but the authors stress that their concepts must be clearly distinguished. After considering a responsibility attribute, the concepts of agent and actor are different. Therefore, the authors defined an agent as subclass of an actor. An actor can become an agent when an actor is a human or an organization of humans, since a human can take responsibility. Conversely, when an actor is not a human or an organization of humans, the actor cannot become an agent, because the actor cannot take responsibility. For example, IT or physical systems can become actors in d* framework, but they cannot take responsibility. Therefore, they cannot become agents in d* framework. Instead, a human that can take responsibility must be an agent. In this case, the human also is an actor. Also, while an actor is a human, the responsibility of an actor might be supported by other humans. In such a case, actors and agents are different humans. In general, humans must take responsibility in human society. Researchers must consider supporting responsibility by humans in dependability cases. Only humans or organizations of humans can satisfy accountability in dependability cases.

12.5.3 Notational Extensions of d framework*

In this study, the authors adopted a graphical notation for describing a responsibility attribute in dependability cases by using d* framework. Other notations can also be considered for describing responsibility attributes in dependability cases. For example, the methods below can also be considered:

- Extend a statement format of an element to show responsibility.
- Use an external table to indicate the relationship between element and responsibility.

In the future, it is necessary to evaluate a suitable method while considering the purpose of the dependability case.

12.5.4 Meta Model

Relations between elements of d* framework had become complicated by introduction of Agent. Therefore, the authors created a Meta model that shows relation between elements of d* framework. As a result, relation between elements of d* framework can be shown correctly. Cardinality of relations also can be shown in Meta model. This model will be used when the new concepts will be introduced into d* framework. For example, when attacker is considered as special actor, the Meta model can be used. That is, attacker can be defined using the Meta model.

12.5.5 Applicability

Three examples of d* framework showed the wide applicability of the method. Elevator system is an example of embedded systems. AP download system is an application of business model domain. Because, it describes inter relationship between business roles that are represented by agents. LAN device management system is an application for security domain management.

12.5.6 *Limitation*

In this study, dependability case examples that introduced a responsibility attribute by using an existing dependability case was created. The dependability case example that has responsibility attributes were not created from the beginning. The responsibility attributes was appended to dependability case that did not have responsibility attributes. Therefore, the authors did not consider the procedure for introducing responsibility attributes into a dependability case. Nonetheless, the following can be concluded: (1) it is necessary to prepare the procedure of eliciting an agent; and (2) it is better to add an agent to a dependability case after dependability analysis (inter analysis and inner analysis). After creating a dependability case, the procedure that checks by comparing the organizational and dependability case-layered structures may be also needed.

12.6 Conclusion

In this chapter, a concept for incorporating responsibility attributes into dependability cases described by d* framework is proposed. A simple responsibility notation is also proposed. Traditional dependability cases only describe goals, strategies, contexts, and solutions. In this chapter, the responsibility attribute is introduced to the dependability case. The whereabouts of accountability in a dependability case can be clearly shown in the proposed d* framework. The responsibility for the system is needed to be clearly shown to achieve the accountability. To ensure achievement of the accountability in dependability cases, the responsibility of elements of the dependability case should be agreed on among stakeholders. As shown in this chapter, the responsibility attribute can be defined as the relationship among stakeholders in dependability cases. The proposed dependability case concept to incorporate responsibility attributes can be used for sharing responsibility knowledge among stakeholders. The agent concept is introduced to define human object who has the responsibility attributes in d* framework. Meta model of d* framework is also proposed. To show the effectiveness of the d* framework, three example applications are described. These examples are selected from different domains. This shows the wide applicability of d* framework.

In the future, It is planned to evaluate the effectiveness of the proposed concept and corresponding notation through experiments in describing practical dependability cases. Procedures for creating dependability cases that have responsibility attributes will also be future studies.

References

1. Kelly, T.: Arguing safety—a systematic approach to managing safety cases. Ph.D. thesis, University of York (1998)
2. Kelly, T.: Using software architecture techniques to support the modular certification of safety-critical systems. In: 11th Australian Workshop on Safety Critical Systems and Software, pp. 53–65. Australia (2005)
3. Despotou, G.T.K.: Design and development of dependability case architecture during system development. In: 25th International System Safety Conference, Baltimore, USA (2007)
4. Tokoro, M.: Open Systems Dependability: Dependability Engineering for Ever-Changing Systems. CRC Press, Florida (2012)
5. Yamamoto, S., Matsuno, Y.: d* framework: Inter-dependency model for dependability. In: DSN 2012 (2012)
6. Despotou, G., Kelly, T.: Extending safety deviation analysis techniques to elicit flexible dependability requirements. In: System Safety, 2006. 1st Institution of Engineering and Technology International Conference, pp. 29–38 (2006)
7. van Lamsweerde, A.: Requirements Engineering: From System Goals to UML Models to Software Specifications. Wiley, N.Y (2009)
8. van Lamsweerde, A., Letier, E.: Integrating obstacles in goal-driven requirements engineering. In: 20th International Conference on Software Engineering—Forging New Links (ICSE 98), pp. 53–62. Kyoto, Japan (1998)
9. Sommerville, I., Lock, R., Storer, T., Dobson, J.: Deriving information requirements from responsibility models. In: 21st International Conference on Advanced Information Systems Engineering, pp. 515–529. Amsterdam, The Netherlands (2009)
10. McDermid, J.: Software safety: Where's the evidence? In: 6th Australian Workshop on Safety Critical Systems and Software, pp. 1–6. Brisbane, Australia (2001)
11. Iain, B., Kelly, T.: Architectural considerations in the certification of modular systems. Reliab. Eng. Syst. Saf. **81**(3), 303–324 (2003)
12. Kelly, T., Weaver, R.: The goal structuring notation—a safety argument notation. In: Dependable Systems and Networks 2004 Workshop on Assurance Cases (2004)
13. Despotou, G., Kelly, T.: Extending the safety case concept to address dependability. In: 22nd International System Safety Conference (2004)
14. Avizienis, A., Laprie, J., Randall, B., Landwehr, C.: Basic concepts and taxonomy of dependable and secure computing. IEEE Trans. Dependable Secure Comput. **1**(1), 11–33 (2004)
15. Yu, E.S.K.: Towards modeling and reasoning support for early-phase requirements engineering. In: 3rd IEEE International Symposium on Requirements Engineering, pp. 226–235 (1997)
16. Greenwood, D., Sommerville, I.: Responsibility modelling for the sociotechnical risk analysis of coalitions of systems. In: IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 1256–1261 (2011)
17. Baxter, G., Sommerville, I.: Socio-technical systems engineering handbook. http://archive.cs.st-andrews.ac.uk/STSE-Handbook/FullHandbook.pdf (2012)
18. Feltus, C., Petit, M.: Building a responsibility model including accountability, capability and commitment. In: Availability, Reliability and Security, ARES '09. International Conference, pp. 412–419. Fukuoka, Japan (2009)
19. Boness, K.D., Harrison, R.: Goal sketching with activity diagrams. In: Software Engineering Advances, ICSEA '08. 3rd International Conference, pp. 277–283 (2008)
20. Strens, R., Dobson, J.: Responsibility modelling as a technique for organisational requirements definition. Intell. Syst. Eng. **3**(1), 20–26 (1994)

Chapter 13
Designing a Hybrid Recommendation System for TV Content

Na Chang, Mhd Irvan and Takao Terano

Abstract In the area of knowledge-based information systems, TV recommendation systems have attracted researchers' attention with the development of Smart TV and expansion of TV content. Traditional TV recommendation systems, which are based on individual's viewing activity and only recommending TV programs, have been unable to meet the requirements of Smart TV. Hence, this chapter proposes a hybrid recommendation system, which not only uses the information of single user's activity, but also takes into account other users' viewing habits and related information from the Internet, for different content such as TV programs, movies, and music. The proposed recommendation system integrates Content Analysis Component, User Analysis Component and Preference Learning Component. Moreover, this chapter also discusses several important design issues, such as diversity, novelty, explanation and group recommendations, which should be considered in designing/building a TV recommendation system. The proposed framework could be used to help designers and developers to design a TV recommendation system engine for Smart TV.

Keywords TV content · Recommendation systems · Smart TV

13.1 Introduction

With the development of Smart TV and expansion of TV content, hundreds of channels from cable or satellite provider, as well as Internet-based content providers like Netflix, Hulu, YouTube, are available through TV devices to users. The tremendous

N. Chang (✉) · M. Irvan · T. Terano
Tokyo Institute of Technology, 4259 Nagatsuta-cho, Midori-ku,
Yokohama, Kanagawa 226-8503, Japan
e-mail: changna@trn.dis.titech.ac.jp

M. Irvan
e-mail: irvan@trn.dis.titech.ac.jp

T. Terano
e-mail: terano@dis.titech.ac.jp

J.W. Tweedale et al. (eds.), *Intelligent Decision Technology Support in Practice*, Smart Innovation, Systems and Technologies 42,
DOI 10.1007/978-3-319-21209-8_13

TV content, on one hand, may bring users many choices; on the other hand, users sometimes may feel confused and find it hard to identify interesting TV content because of the massive volume of content available at the same time. Hence, personalized TV recommendation systems/engines have become increasingly important and have been adopted by many famous television device makers and content providers, such as Google, Apple, Sony, YouTube.

To address these situations, there are many related research papers published in recent years. For instance, Das and ter Horst [1] addressed several main research questions of TV program recommendations, such as profiling methods, recommendation algorithms, and group recommendation issue. Asabere [2] gave a comprehensive survey of TV recommendation systems. Unfortunately, most of existing research focuses on TV recommendation systems that are based on the concept associated with traditional television devices. For instance, television devices are not connected to the Internet or recommendation algorithms are only based on individual user or TV programs. Obviously, this situation cannot meet the requirements of Smart TV through which users can access sorts of content like TV programs, movies, and music. In addition, researchers mainly focus on the improvement of accuracy of TV recommendations, ignoring the importance of other aspects such as diversity, novelty, explanation of recommendations.

Based on a comprehensive literature review of TV recommendation systems, this chapter proposes a hybrid recommendation system for TV content. The proposed recommendation system consists of TV Content Analysis Component, User Analysis Component and Preference Learning Component. These components not only treat TV content and users direct feedback, but also suggest extracting related information such as TV watching statistics information, users preference/interest for the other content from social media or rating systems. Furthermore, in the Preference Learning Component, this chapter suggests three user preference learning approaches: learning from individual's past experience, learning from public opinion and learning from neighbourhood. In addition to the proposed framework of TV content recommendation system, this chapter also discusses several design issues, for instance; diversity, novelty, explanation and group recommendations, and provides several corresponding solutions respectively.

13.2 Related Work

Much work has been done in the area of personalized TV recommendations. This chapter gives a comprehensive survey of previous research. Table 13.1 presents the main contents and recommendation techniques proposed in these research.

Table 13.1 The main contents of related research

Articles	Main contents	Techniques
Ardissono et al. [3]	A user model that stores the estimates of the individual user preferences for TV program categories and channels based on explicit user information, prediction on the user references inferred from prior information about TV viewer categories, the estimates on the user preferences inferred by observing her viewing behaviour	Classification, Bayesian Network
Cotter and Smyth [4]	Explicitly obtain users preference about channel, genre, and viewing time, and infer users preference based on their feedback on TV shows	Case-Based Reasoning, Collaborative Filtering
Das and ter Host [1]	Addressed main research questions of TV program recommendations: profiling methods, methods of eliciting users profile, method of information filtering, group recommendations	None
Engelbert et al. [5]	Built two types of user profiles: initial user profile defined by user during the beginning of the system, and adaptive user profile created continuously on the basis of the users viewing behavior	Bayesian Classifier
Hsu et al. [6]	Proposed a hybrid TV program recommender system based on user properties such as Activities, Interests, Moods, Experiences, and Demographic information	Artificial Neural Network
Martínez et al. [7]	Introduced a personalized TV program recommendation system, which combines content-filtering techniques with those based on collaborative filtering, and also provides all typical advantages of any social network as comments, tagging and ratings	Vector Space Model, Singular Value Decomposition, and Item-Based Filtering

(continued)

Table 13.1 (continued)

Articles	Main contents	Techniques
Smyth and Cotter [8]	Provided an Internet-based personalized TV listings service, which contains five components: profile database and profiler, program case-base, schedule database, recommender, guide compiler	Case-Based Reasoning, User profiling and Collaborative Filtering
Srinivas et al. [9]	The explicit recommender agent generates program recommendations based on users explicit profile; The implicit agents generate the likelihood that the viewer will like or dislike a particular TV program; The feedback agent fine-tunes recommendation quality	Bayesian Classifier, Decision Tree
Uberall et al. [10]	Generated Recommendation Index for genres, sub genres and events by using implicit and explicit profiles	Content-Based Filtering
Yu et al. [11]	Presented recommendation strategy for multiple viewers based on user profile merging. They firstly merged all user profiles to construct a common user profile, and then used a recommendation approach to generate a common program recommendation list for the group according to the merged user profile	None
Zimmerman et al. [12]	The implicit recommender generates profiles based on users viewing histories and explicit recommender generates recommendations based on users inputting profile and feedback	Bayesian Classifier, Decision Tree, Artificial Neural Network

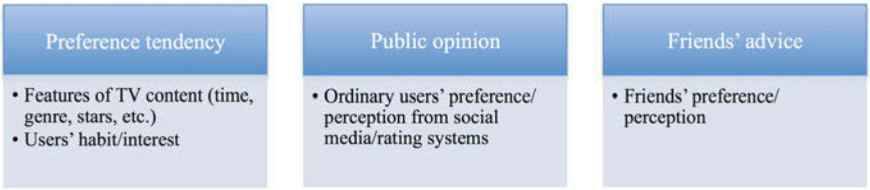


Fig. 13.1 The main points of assumed scenario

13.3 Framework of TV Content Recommendation System

Before introducing the proposed recommendation system for TV content, this chapter shows an assumed scenario. A user Jim wants to decide whether to watch a movie *Hancock* this Saturday night, or go shopping. He firstly checks the basic information of the movie, such as director, stars, genres, and make his initial knowledge about the movie. For example, if Jim likes movie star Will Smith and fantasy movie, he is likely to prefer this movie and would like to know more information about it. To confirm his first perception, then Jim tries to know about other users' opinion on the movie from some famous rating systems such as IMDb[1] and Yahoo movies.[2] He finds the average rating on IMDb about the movie is 6.4 out of 10, and average rating on Yahoo movies is 4.5 out of 5. The rating is relatively high on Yahoo movies, but is normal on IMDb. Then Jim finds that there are 225,488 users who gave ratings to the movie on IMDb, but there are just 2,375 users of that on Yahoo movies. After detecting this, Jim is a little confused by the inconsistent rating information. Then he decides to seek advice from his friends who have watched the movie. The result is that most of his friends think the movie is worth to watch. Finally, Jim decides to watch the movie in the weekend. From this scenario, the following assumptions can be made, see also Fig. 13.1.

- A user's preference for TV content is influenced by his/her own preference tendency, public opinion and friends' advice;
- A user's preference tendency is related to the characteristics of TV content and the user's habits;
- Public opinion indicates the preference/perception of ordinary user; and
- Friends' advice indicates the influence of a user' friends who have the similar preference/interest.

On the basis of the survey of TV recommendation systems and the assumptions provided, this chapter proposes a hybrid recommendation system for TV content. Figure 13.2 depicts the framework of the proposed recommendation system and identifies three components:

[1] http://www.imdb.com/.

[2] http://movies.yahoo.com/.

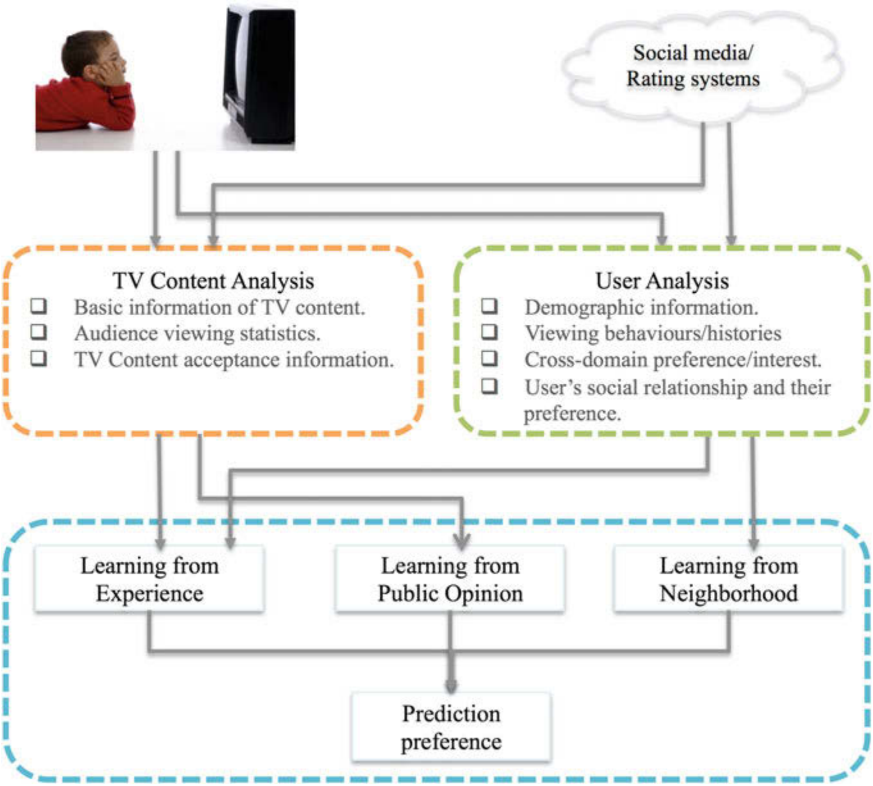


Fig. 13.2 The framework of proposed TV content recommendation system

- Content Analysis Component.
- User Analysis Component.
- Preference Learning Component.

Each of these components has its specific tasks and aims. The following sections address the specific definition of each component.

13.3.1 The Content Analysis Component

The main task of the Content Analysis Component is to process and analyze information from content providers and social media/rating systems, obtain basic information of TV content, audience viewing statistics and TV content acceptance information, and provide to generate structural and useful data for the Preference Learning Component.

Basic information of TV content: the basic information of TV content could be title, genre/sub-genre, keywords, actors. Among them, keywords mean the main terms, which are used to describe the main features of TV content. Keywords, genre/sub-genre, actors are the important characteristics of TV content, because this information could be used to distinguish the TV content from others and describe users' interests/habits. This type of information can be extracted from the description of the TV content provided by broadcast/content provider.

Audience viewing statistics: this type of information can be collected from related companies, research institutes, and TV content providers. Audience viewing statistics reflects the macro characteristics and trend of TV watching behaviours of users. This information could be important factors in learning and predicting individual's preference.

TV content acceptance information: this type of information means public's perceptions, feelings or opinions for certain TV content and can be extracted by text/data mining algorithms/tools from on-line social media, rating systems or other sources. There are many online social media/rating systems, such as IMDb, Yahoo movies, and AngiesList.[3] Most of these social media and rating systems enable users to give ratings or reviews/comments for specific products or service. From the rating distribution and reviews, the acceptance of TV content can be inferred.

13.3.2 The User Analysis Component

The main task of the User Analysis Component is to collect and extract users demographic information, viewing behaviours/histories, cross-domain preference/interest and social relationships' preferences/habits. This information could be used to provide user profile information to the Preference Learning Component.

Demographic information: this type of information means users' name, age, sex, occupation, income and so on. Demographic information is proved to be a factor can influence users' preferences. Currently, many TV recommendation systems use this factor to predict users' preferences [7, 12]. Demographic information also can be used to classify users into separate preference sets and then maps them to appropriate styles [3].

Viewing behaviours/histories: users' past viewing behaviours/histories may reflect their interests and watching habits. This type of information could tell that a user watched what TV content at what time and how long the user watched them and how he/she prefers. The time user spent watching the content indicates the user's watching habit. How long the user watched the content and rating indicates the users' preference for it.

[3] http://www.angieslist.com/.

Cross-domain preference/interest: users' preference in one domain may be used to predict their preference in other domain. Hence, it is important to explore users' cross-domain preference/interest. With the development of Smart TV, users could have fun with TV programs, movies, music through relevant applications on the same TV device. Users' preference in one or two of these domains like movies and music, could be used to predict users' preference in the other domain. For example, if a user like watching comedy movies, then he/she may like comedy TV programs as well.

Social relationships and their preference: Users' preferences tend to be influenced by their friends/family/colleagues, so it is reasonable to take users' social relationships' preference into account. Users' social relationships could be explicit relationship, which may be extracted from relevant social network such as FaceBook or Twitter, and implicit relationship, which can be extracted based on users' past or current preference and habits.

13.3.3 The Preference Learning Component

The Preference Learning Component is the core function of the proposed recommender systems. The main tasks of this component include: learning from individual's past experience, learning from public opinion and learning form neighbourhood.

Learning from Experience: users preference tendency can reflect users preference based on their past watching histories and basic information of TV content. For example, if a user often watches comedy drama in the night, it can be inferred that he/she likes comedy drama and his/her TV content watching habit is at night. Then when there is a new comedy drama, the recommender could recommend it to the user. With respect to the learning algorithms, content-based filtering methods [13], such as Bayesian Classifier [5, 9], Case-Based Reasoning [14, 15], and Vector Space Model [16, 17], are good choices and recommended.

Learning from Public Opinion: public opinion also can influence users perception for TV content. With the development of social media and rating systems, most users tend to want to know the public's perception/opinion on certain TV content before they making decisions. There may be several social media and rating systems related to TV content, so it is necessary to predict public opinion on the TV content by integrating several sources. If the public ratings for TV content are high, then it is possible that users want to watch it.

Learning from Neighbourhood: traditional Collaborative Filtering algorithm is the most popular approach in recommendation systems [18]. A user's preference for TV content may be influenced by the opinion of his/her social relationships such as friends, colleague, and family members. Social relationships could be explicit or implicit. Explicit social relationship means users are connected through real relationship (friends/family/colleague) or social network. For example, a user has watched *The Big bang Theory* and likes it, and then he/she may recommend the drama to his/her friends or colleague. On the other hand, implicit social relationship is based on users' interests and preferences. Users may have similar preference for TV content even though they don't know each other. For example, there is a group of users, who don't know each other, used to watch comedy drama and gave high ratings to comedy drama, and then it can be predicted the preference of one of them based on the other users' preferences. Collaborative Filtering methods, such as User or Item-based Filtering [19–21], and Matrix Decomposition method [22, 23], are recommended in this learning task.

13.4 Important Design Issues

Considering the situation that most of previous research related to TV content recommendation systems is focused on the improvement of prediction accuracy, this chapter presents several other important design issues, i.e. diversity, novelty, explanation and group recommendations. Each of these issues is indispensable in designing/building recommendation systems for TV content and should be take into consideration by designers/developers.

13.4.1 Diversity

Diversity refers to how different the TV content in a recommendation list are with respect to each other, and it is one important metric to evaluate the quality of a TV recommendation system. McNee et al. [24] argued that providing accurate recommendations is not enough, the other aspect such as diversity of recommendations should be taken into account. For instance, recommending a list of very similar TV content, such as a list of comedy movies, may be not attractive to a user, even when the list's average accuracy might be high. Because the user may want many different choices, not only one kind of movies.

To provide diverse recommendations for TV content, the main approach is selecting the TV content that has maximum similarity with users' preference and, at the same time, has minimum similarity with other TV programs in the recommendation list [25]. For example, Ziegler et al. [25] presented a novel method called topic diversification which is designed to balance and diversify personalized recommendation lists in order to reflect the user's complete spectrum of interests. Sheth et al. [26] proposed an approach to diversity in recommendations called Social Diversity, which utilizes social networks in recommendation systems to leverage the diverse of underlying preferences of different user communities to introduce diversity into recommendations.

13.4.2 Novelty

Novelty of recommendations is defined as the proportion of known and unknown relevant TV content in the recommended list. The core concept of novelty is related to a recommender's ability to educate users and help them discover new TV content [27]. With the development of recommendation systems and consumers requirements, there is an increasing realization that novelty and diversity are fundamental qualities of recommendation effectiveness and added-value [28].

Several researches have mentioned novelty issues in recommendation systems. Celma and Herrera [29] presented item-centric and user centric methods to evaluate the quality of novel recommendations. Weng et al. [30] proposed a taxonomy-based recommendation system improve its recommendation quality and novelty. The proposed recommender utilizes techniques from association rule mining to find how different topics are associated with each other in a given user cluster. Based on the discovered topic associations, the recommender suggests items with topics that are strongly linked to the taxonomy profile of the target user.

13.4.3 Explanation

Explanation of recommendations means reasons for recommending particular TV content to particular users and help users to know why they get the recommendations and decide whether to adopt the recommendations or not. Generating explanations for specific recommendations is import for a recommendation system. For instance, Swearingen and Sinha [31] found that a good algorithm that generates accurate recommendations is not enough to constitute a useful system from users perspective and the system needs to convey to users its inner logic and why a particular recommendation is suitable for them. Hu and Pu [32] found that some users were curious to know more about how the system was achieving such good recommendations.

There are several recommendation systems provide explanations for their suggestions in the form of similar items the consumer has rated highly, like Amazon, or keywords describing the item caused it to be recommended [33]. Friedrich and Zanker [34] summarized several explanation approaches in related articles. For instance, Bilgic and Mooney [35] presented content-based explanation styles, which provides argumentation based on keywords describing the recommendation item, and collaborative explanation styles, which disclose the behaviour of the most similar peers.

13.4.4 Group Recommendations

As entertainment devices, TV are often viewed by a group of users, such as a family or viewers at public places. For example, people in one family may have different preferences and habits for TV content. The housewife may often watch TV in the afternoon, children may often watch cartoon in the evening and the husband may often watch news in the night. But when they watch TV content in the weekend, they want to watch the preferable TV content for everyone. So the TV recommendation systems should not only provide personalized content for individuals, but also be able to recommend content to multiple viewers taking care of the preferences of the majority of viewers, in the case where the viewers are watching TV at the same time, and in the same spot [27].

There are several approaches to solve group recommendations for TV content. For instance, Masthoff [36] discussed different strategies for combining individual user models to select TV items to suit groups of viewers and arrive at a recommendation decision to a group of users through combining individual user ratings on whole programs rather than features. Jameson [37] proposed a user interest aggregation method for group recommendation by allowing the current member optionally to view the preferences already specified by other members. Yu et al. [38] suggested using a user profile merging algorithm, which merges individual profiles so as to form a common user profile that reflects most and consistent preferences of the group.

13.5 Final Remarks

This chapter proposes a hybrid recommendation system for Smart TV. The proposed recommendation system consists of TV Content Analysis Component, User Analysis Component and Preference Learning Component. These components work together to generate personalized TV content for users. In addition to the framework, this chapter also addresses the importance of diversity, novelty, explanation and group recommendation in designing/building of a TV program recommendation system, and shows several corresponding solutions for these issues.

The proposed framework of recommendation system will be developed and implemented in the future work including three components presented in Sect. 13.3. Furthermore, suitable algorithms will also be integrated in the system considering the important design issues discussed in Sect. 13.4.

References

1. Das, D., ter Horst, H.: Recommender systems for TV. Technical Report WS-98-08, AAAI, California, USA (1998)
2. Asabere, N.Y.: A survey of personalized television and video recommender systems and techniques. Int. J. Inf. Commun. Technol. Res. **2**(7), 602–608 (2012)
3. Ardissono, L., Gena, C., Torasso, P., Bellifemine, F., Difino, A., Negro, B.: User modeling and recommendation techniques for personalized electronic program guides. In: Personalized Digital Television. Human-Computer Interaction Series, vol. 6, pp. 3–26. Springer, Netherlands (2004)

4. Cotter, P., Smyth, B.: Ptv: Intelligent personalised tv guides. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, AAAI Press (2000), pp. 957–964
5. Engelbert, B., Blanken, M., Kruthoff-Bruwer, R., Morisse, K.: A user supporting personal video recorder by implementing a generic bayesian classifier based recommendation system. In: 2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops). (March 2011), pp. 567–571
6. Hsu, S.H., Wen, M.H., Lin, H.C., Lee, C.C., Lee, C.H.: Aimed- a personalized tv recommendation system. In: Cesar, P., Chorianopoulos, K., Jensen, J. (eds.) Interactive TV: a Shared Experience. Lecture Notes in Computer Science, vol. 4471, pp. 166–174. Springer, Berlin (2007)
7. Martínez, A.: Pazos Arias, J.J., Vilas, A.F., Duque, J.G., Nores, M.L.: What's on tv tonight? an efficient and effective personalized recommender system of tv programs. IEEE Trans. Consum. Electron. **55**(1), 286–294 (2009)
8. Smyth, B., Cotter, P.: A personalised tv listings service for the digital tv age. Knowl. Based Syst. **13**(2–3), 53–59 (2000)
9. Srinivas, K.K., Gutta, S., Schaffer, D., Martino, J., Zimmerman, J.: A multi-agent tv recommender. In: Workshop on Personalization in Future TV, Sonthofen (2001)
10. Uberall, C., Muttukrishnan, R., Rakocevic, V., Jager, R., Kohnen, C.: Recommendation index for dvb content using service information. In: Proceedings of the 2009 IEEE International Conference on Multimedia and Expo, pp. 1178–1181. IEEE Press, Piscataway, NJ, USA (2009)
11. Yu, Z., Zhou, X., Hao, Y., Gu, J.: Tv program recommendation for multiple viewers based on user profile merging. User Model. User-Adap. Inter. **16**(1), 63–82 (2006)
12. Zimmerman, J., Kauapati, K., Buczak, A., Schaffer, D., Gutta, S., Martino, J.: Tv personalization system. In: Personalized Digital Television. Human-Computer Interaction Series, vol. 6, pp. 27–51. Springer, Netherlands (2004)
13. Lops, P., Gemmis, M., Semeraro, G.: Content-based recommender systems: State of the art and trends. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 73–105. Springer, US (2011)
14. Bridge, D., Goker, M.H., McGinty, L., Smyth, B.: Case-based recommender systems. Knowl. Eng. Rev. **20**(3), 315–320 (2005)
15. Smyth, B.: Case-based recommendation. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web, pp. 342–376. Springer, Berlin (2007)
16. Bin Ghauth, K., Abdullah, N.: Building an e-learning recommender system using vector space model and good learners average rating. In: Ninth IEEE International Conference on Advanced Learning Technologies (July 2009), pp. 194–196
17. Musto, C.: Enhanced vector space models for content-based recommender systems. In: Proceedings of the Fourth ACM Conference on Recommender Systems. RecSys' 10, pp. 361–364. ACM, New York, NY, USA (2010)
18. Ekstrand, M.D., Riedl, J.T., Konstan, J.A.: Collaborative filtering recommender systems. Found. Trends Hum. Comput. Interact. **4**(2), 81–173 (2010)
19. Desrosiers, C., Karypis, G.: A comprehensive survey of neighborhood-based recommendation methods. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 107–144. Springer, US (2011)
20. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Comput. **7**(1), 76–80 (2003)
21. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web. WWW' 01, pp. 285–295. ACM, New York, NY, USA (2001)
22. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
23. Paterek, A.: Improving regularized singular value decomposition for collaborative filtering. In: Proceedings of KDD Cup Workshop at SIGKDD'07, pp. 39–42. ACM, New York (2007)

24. McNee, S.M., Riedl, J., Konstan, J.A.: Being accurate is not enough: How accuracy metrics have hurt recommender systems. In: CHI '06 Extended Abstracts on Human Factors in Computing Systems. CHI EA '06, pp. 1097–1101. ACM, New York, NY, USA (2006)
25. Ziegler, C.N., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In: Proceedings of the 14th International Conference on World Wide Web. WWW '05, pp. 22–32. New York, NY, USA (2005)
26. Sheth, S.K., Bell, J.S., Arora, N., Kaiser, G.E.: Towards diversity in recommendations using social networks. Computer science technical reports, Columbia University (2011)
27. Pu, P., Chen, L., Hu, R.: A user-centric evaluation framework for recommender systems. In: Proceedings of the Fifth ACM Conference on Recommender Systems, pp. 157–164. ACM, New York, NY, USA (2011)
28. Castells, P., Vargas, S., Wang, J.: Novelty and diversity metrics for recommender systems: Choice, discovery and relevance. In: International Workshop on Diversity in Document Retrieval (DDR 2011) at the 33rd European Conference on Information Retrieval (2011)
29. Celma, O., Herrera, P.: A new approach to evaluating novel recommendations. In: Proceedings of the 2008 ACM Conference on Recommender Systems. RecSys '08, pp. 179–186. ACM, New York, NY, USA (2008)
30. Weng, L.T., Xu, Y., Li, Y., Nayak, R.: Improving recommendation novelty based on topic taxonomy. In: 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology Workshops. (Nov 2007), pp. 115–118
31. Swearingen, K., Sinha, R.: Interaction design for recommender systems. Designing Interact. Syst. **6**(12), 312–334 (2002)
32. Hu, R., Pu, P.: Acceptance issues of personality-based recommender systems. In: Proceedings of the Third ACM Conference on Recommender Systems. RecSys' 09, pp. 221–224. ACM, New York, NY, USA (2009)
33. Mooney, R.J., Roy, L.: Content-based book recommending using learning for text categorization. In: Proceedings of the Fifth ACM Conference on Digital Libraries. DL '00, pp. 195–204. ACM, New York, NY, USA (2000)
34. Friedrich, G., Zanker, M.: A taxonomy for generating explanations in recommender systems. AI Mag. **32**(3), 90–98 (2011)
35. Bilgic, M., Mooney, R.: Explaining recommendations: Satisfaction vs. promotion. In: Proceedings of Beyond Personalization 2005: A Workshop on the Next Stage of Recommender Systems Research at the 2005 International Conference on Intelligent User Interfaces, San Diego, CA (January 2005)
36. Masthoff, J.: Group modeling: selecting a sequence of television items to suit a group of viewers. User Model. User-Adap. Inter. **14**(1), 37–85 (2004)
37. Jameson, A.: More than the sum of its members: Challenges for group recommender systems. In: Proceedings of the Working Conference on Advanced Visual Interfaces. AVI '04, pp. 48–54. ACM, New York, NY, USA (2004)
38. Yu, Z., Zhou, X., Hao, Y., Gu, J.: User profile merging based on total distance minimization. In: Proceedings of the 2nd International Conference on Smart homes and health Telematics, IOS Press (2004), pp. 25–32

Chapter 14
Incompleteness and Fragmentation: Possible Formal Cues to Cognitive Processes Behind Spoken Utterances

L. Hunyadi, H. Kiss and I. Szekrenyes

Abstract What may eventually connect engineers and linguists most is their common interest in language, more specifically language technology: engineers build more and more intelligent robots desirably communicating with humans through language. Linguists wish to verify their theoretical understanding of language and speech through practical implementations. Robotics is then a place for the two to meet. However, speech, especially within spontaneous communication seems to often withstand usual generalizations: the sounds you hear are not the sounds you describe in a laboratory, the words you read in a written text may be hard to identify by speech segmentation, the sequences of words that make up a sentence are often too fragmented to be considered a "real" sentence from a grammar book. Yet, humans communicate, and this is most often, successful. Typically this is achieved through cognition, where people not only use words, these are used in context. People also use words in semantic context, by combining voices and gestures, in a dynamically changing, multimodal situational context. Each individual does not simply pick out words from the flow of a verbal interaction, but also observes and reacts to other, using multimodal cues as a point of reference and inference making navigation in communication. It is reasonable to believe that participants in a multimodal communication event follow a set of general, partly innate rules based on a general model of communication. The model presented below interperate numerous forms of dialogue by uncovering their syntax, prosody and overall multimodality within the HuComTech corpus of Hungarian. The research aims at improving the robustness of the spoken form of natural language technology.

Keywords Syntax · Prosody · Gestures · Multimodality · HuComTech

---

L. Hunyadi (✉) · H. Kiss · I. Szekrenyes
University of Debrecen, Egyetem Tér 1, Debrecen 4012, Hungary
e-mail: hunyadi@unideb.hu

J.W. Tweedale et al. (eds.), *Intelligent Decision Technology Support in Practice*, Smart Innovation, Systems and Technologies 42,
DOI 10.1007/978-3-319-21209-8_14

14.1 A Generative Model of Human–Human Communication

Robotic development for human-machine interaction essentially differs from the development of industrial robots in that, in addition to aiming at technical and technological perfection it has special emphasis on providing more human-like naturalness to this interaction. In order to achieve this goal one has to develop a profound understanding of human-human communication and implement it in a given application. The collections of large amounts of data about various particular aspects of human communication undoubtedly serve this goal by offering the basis for statistical models to target specific control tasks [1–3]. However, the main challenge is that, on the surface, in other words, in terms of exact sequences and alignments of verbal and nonverbal events human communication never repeats itself. As a result, one can always come across with sequences and alignments on which an application trained on a statistical model will fail. Human communication essentially follows the psychological and neuro-physiological modularity and variability of speech acts [4]. Accordingly, one and the same communicative function, such as a simple greeting, can be realised in an unlimited number of speech acts, each varying by the presence or absence, sequence or alignment of certain modalities, all depending on the "here" and "now" of the respective setting the function is performed in. Accordingly, one needs to treat these individual variations as representatives of some more general, holistic paradigms of communicative behaviour. Accordingly, in order to build such a system, one needs to learn the basic, general, individually non-variable ingredients of these events and build variations upon them. The framework for this is a theoretical model having two main arms: one has to describe the invariant properties of human communication and the ways how they are realised through individual variation, the other in strong association with the first one has to describe how these human actions are technologically implemented on the surface [5].

Such a model needs to have three essential properties: it needs to be multimodal, generative and modular. As for multimodality, it has to capture all the verbal and non-verbal properties of a communicative event relevant for technology, including their sequential ordering, temporal alignment and possible hierarchies. Importantly, its multimodal character is required to theoretically represent the holistic property of human cognitive action and perception, but, not in contradiction with this requirement, it presupposes unimodality: it has to individually handle every event in every single modality for technological implementation. As for generativity, the model has to handle the two-way character of communication: since communication involves both acting and reacting (most often speaking and listening), in other words, participants in an event alternate in their roles, this bidirectional property of communication needs to be reflected in the model as well. Following generative activity, the model has to handle both synthesis and analysis and, importantly, based on the same rules or procedures. Ideally, these bidirectional rules make it then possible for a robot to "learn" certain behavioural actions to both perform and perceive. Finally, an examination of modularity should indicate that a communicative event is built of the hierarchy and temporal sequence of sub-events with their own specific function and degree of abstraction as building blocks of a larger whole. Each of the modules is responsible for a particular stage in the process culminating in the construction of a technologically realized event (generation) or, in the opposite direction, in deciphering (interpreting) the message of such an event (analysis).

The goal then is twofold: first, to provide a model that breaks down communication to the required details by defining its minimal ingredients across all possible modalities, and second, to establish a structured, hierarchical dependency of these ingredients responsible both for the synthesis (generation) and analysis (interpretation) of the event. (For a more detailed description of the model [6].)

14.1.1 The Scheme of the Generative Theoretical-Technological Model

The scheme of this generative theoretical-technological model is shown in Fig. 14.1. Each of the modules is based on the same underlying structure: they build a structure to their own specificity with a certain abstraction. Accordingly, the module Basic structure serves to determine the very general formal, the structural frames of a given communicative event independent of its actual content; the module Functional extension supplies functions to these “naked” frames enriching them with communicative functions generally required for certain types of events irrespective of the actual context it will be performed within; the module Pragmatic extension further specifies the event with particular, actual properties to accommodate it to the context “here and now”. The Technological extension is the robotic aspect of the application, the module which, appearing on the surface, incorporates all inputs of the previous modules constituting to the highest degree of complexity of form and communicative information.

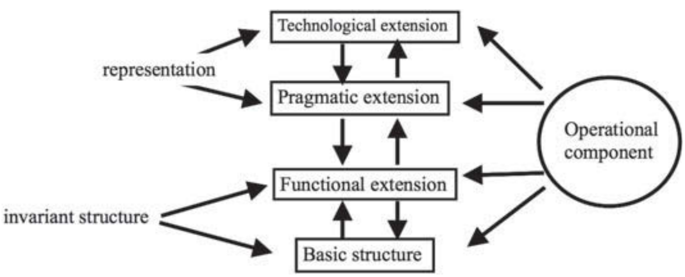


Fig. 14.1 The scheme of the two-way generative theoretical-technological model of communication

14.1.2 Structural Primitives as Minimal Building Blocks

From the modularity of the model it follows that each of the modules is different for being responsible for a function of their own, depending on the position it occupies in the generative hierarchy. What connects them, however, is their structural similarity: they are built of minimal building blocks, the structural primitives specific to their function in constructing the given communicative event, and the same basic operations apply to them to generate further structural variations.

Thus, the Basic module includes the primitives of *start*, *end*, and, in a bottom-up approach, their recursive application generates a virtually infinite hierarchy of embedded and co-ordinated sub-events; the Functional extension includes such primitives as *question*, *answer*, *imperative command*, *exclamation* applied to the recursively generated underlying communicative structure, generating the overall modus of an interaction. Built on top of these structures, the Pragmatic extension actualizes the event by using such primitives as *turn-take*, *turn-keep*, *turn-give*, *topic initiation*, *agreement*, *disagreement*. The Technological extension includes primitives related to the technological representation of an actual event and involve the setting of control parameters.

14.1.3 Primitives Realized as Markers

The purpose of the primitives is to establish the modality-independent skeleton of an event. This is important, because one should recognise certain events, such as greetings as similar even if they are realised on the surface in different modalities: actually, one can greet the other verbally and non-verbally, formally and less formally alike, yet, the event is equally recognised as greeting. However, if one wants to make a further step towards some unique event realisation, one needs to make a choice which modality or modalities to select. The model introduces markers as modality dependent realisations of certain moments in the skeleton of the event, through which the event is to be realised. Accordingly, people can start an interaction verbally by using such verbal markers as saying "Hello!" or, alternatively, by simply using the non-verbal marker of nodding towards the partner. Alternatively, humans can use a combination of these verbal and non-verbal markers, for example, nodding towards the partner while saying "Hello!" Both actions convey the same communicative function. The proper selection of markers to represent the primitives and structures generated from them, is a crucial step in determining the overall modality in which the given communicative event is to take place.

14.1.4 Noisy Markers with Cognitive Bias in the Decision

One of the important properties of human communication is that it is inherently noisy. This noisiness becomes obvious when considering verbal communication: as previously mentioned, one does not always perceive every speech sound or perceives it "incorrectly" due to environmental noise proper, or due to a change in the focus of attention. Noise, however, need not be verbal alone: sometimes one "expects" a gesture to accompany a word or a phrase or any other gesture to support online interpretation, when it is actually missing. If to correctly predict this missing element is important for the proper selection of the next action in human communication, the more it is crucial in human-machine interaction: the machine should make a proper decision even if not all "expected" markers are present, or if not all available verbal or nonverbal data can be assigned a proper functional role in the given event.

It may often happen that at a given point of time input data are scarce for full interpretation and decision making. Even though statistical methods could again give an acceptable probability for the next step to make at that point, it might appear useful to follow human communication in such a case: it suspends decision making for a while, waiting for some "reasonable time" to probably capture some more data. Until such a decision may be made possible, the communicative event (both by humans and the machine) could be redirected along a complementary, side path and return to decision making when additional data have cleaned the noise.

Noise in the verbal modality can be especially misleading: fragmented speech often results in broken syntactic structure, and assuming a missing word incorrectly just to make the structure complete may lead to serious misunderstandings. In such cases, in order to support a suitable decision, it is especially important to consider other, nonverbal markers around the incomplete syntax.

The HuComTech project aims at identifying and describing through annotation all relevant markers of the corresponding primitives in the constituting hierarchical modules across all verbal and non-verbal modalities so as to determine their alignments and other kinds of co-occurrences for the expression of certain communicative functions and pragmatic roles. Researchers are restricted by the scarcity of data capable of easily leading to cognitive bias in human perception. The more it does so in determining the interactive reaction of a machine, the less engineers expect to build a machine that can perform cognition better humans. Scientists hope that the data obtained from human multimodal databases will prove useful in improving the existing decision-making mechanisms. Data and multimodal analyses from spoken syntax, prosody and gestures presented below are intended to point in this direction.

14.2 The Annotation of Spoken Syntax of Hungarian in the HuComTech Corpus

The aim of the annotation of spoken syntax within the HuComTech corpus is threefold: first, to give the very first formal description of Hungarian spoken syntax with the aim to understand its inner organization by comparing it with the syntax of written texts, second, to obtain data regarding the sequential organization of clauses in the flow of text and third, to find out how clausal (in)completeness is related to other modalities, including prosody and gesturing [7]. Whereas the first goal mainly meets the interests of linguistics proper, the latter two are especially relevant to automatic language processing and human machine interaction: by discovering regularities in the sequence of clauses as well as in the alignment of multimodal markers with syntactic structure people may gain important cues for missing linguistic components as well as the cognitive processes behind them.

Cognitive science has long acknowledged the importance of the study of language as a way to understand how the mind works. It has also been gaining importance to observe in this connection that there is a systematic way language is associated with various kinds of gestures leading to the discovery of some cognitive functions of this relation [8–12] and to the far reaching suggestion that language and gestures constitute a self-contained unified cognitive system [13]. Researchers are following this line of thought by assuming that there are cognitive processes behind the apparent fragmentation and incompleteness of spoken linguistic form that can be studied and possibly understood within this unified complex of thought, language and gestures. The system of the syntactic annotation was designed with this background and goals in mind.

14.2.1 Principles of Syntactic Annotation

One often makes a difference between "spoken language" and "written language". Of course, there are many observations to support this distinction. Apart from the difference between spelling and pronunciation the clearest difference is between the structural organisation of what people write and how they speak. Still, considering the language independent, universal cognitive foundations of human language, one can assume that what is written and what is spoken is actually based on the same grammar, on the same syntax, and what only makes a difference is that, due to online processing in speech, the organisation of the sequence of words in spoken utterances reflects certain cognitive processes more overtly, than in writing. Accordingly, in order to describe "spoken syntax" one does not need a syntax different from the one that is manifest in writing, on the contrary, one needs to follow the same grammar in spoken utterances as well. What will make such a spoken syntax nevertheless special is that it will identify its specific fragmentation and incompleteness as compared to the (written baseline) structure. The syntactic annotation scheme of the HuComTech corpus to be presented below follows these assumptions [14].

The annotation focuses on the overall organisation of the spoken material across a whole communicative event. For this reason, and due to the fact that fragmentation does not allow to unambiguously assume the missing elements, phrase level dependencies, the usual focus of systems of automatic parsing are not annotated. On the other hand, whereas the latter systems only recognize syntactically complete structures and fail to capture the structural and conceptual connection between sequences of sentences, the present scheme aims at going beyond the clause to gain a holistic understanding of the syntax of a communicative event together with some of its pragmatic and cognitive properties.

Following these considerations the structural primitive of syntactic analysis will be the clause defined as the maximum surface sequence of grammatically connected words. The sentence will be defined as the maximum surface sequence of grammatically connected clauses. These definitions allow for analysing large chunks of utterances as a sentence, whereas a single word can also be analysed as a sentence or a clause. These are exactly the cases of fragmented speech which can pose serious problems for automatic parsing but which can be, in this scheme of annotation, highly interesting and informative as an insight into the pragmatic and/or cognitive basis of such spoken fragmentation.

14.2.2 The Taxonomy of Syntactic Annotation

Table 14.1 presents the categories of the syntactic annotation. Accordingly, the main focus is on annotating the kind of syntactic relation between clauses and the linear arrangement of syntactically connected clauses. Since discontinuity is one of the characteristic features of spoken utterances rather hard to capture without semantic

Table 14.1 The taxonomy of the syntactic annotation (pattern: n.n.n.n.n.n.n. (where n is a one or two digit number))

Digit#	Definition	Remarks
1	Clause ID	The place of the given clause within the sequence of clauses within the sentence (“1” = the first clause of the sentence)
2	Subordinating	Shows which clause is subordinated to the given clause (“2” = the clause with “2” as its first digit within the given sentence is subordinated to it)
3	Coordination	Shows which clause is coordinated with it (“3” = it is coordinated with the clause with “3” as its first digit; “3,4” = multiple coordination with clauses “3” and “4”)
4	Subordinated to	Shows which clause the given clause is subordinated to (“1” = it is subordinated to the clause with “1” as its first digit)
5	Embedding/Inserting	The clause contains an embedded/inserted clause (“3” = the clause embedded/inserted in the given clause has the ID number “3”)
6	Embedded/Inserted	The given clause is related to another clause without a grammatical marker (“3a,3b” = the given clause is embedded/inserted in clause “3” where “a” and “b” refer to the two halves of clause “3”)
7	Missing categories	Categories missing from the surface; implicit categories specified below in Table 14.2

analysis using automatic parsing especially in fragmentation, the scheme intends to capture discontinuity between clauses while at the same time indicating the structural relation between them, such as *coordination*, *subordination* (distinguishing whether the clause is subordinated or is in fact subordinating another clause), *embedding/inserting* (distinguishing whether the clause is embedded/inserted or is in fact embedding/inserting another clause).

Since different digit places correspond to the indication of being main or subordinate clause, following the numerical identifiers of clauses it is straight forward to track syntactic dependencies between clauses across their discontinuous sequences in either forward or backward direction. In addition to denoting syntactic relations between clauses, the scheme also indicates kinds of incompleteness within clauses in terms of missing syntactic elements. Remarks in Table 14.1 further clarify how missing elements are determined. (One remark: the scheme was designed to follow some sort of descriptive approach so as to be as theory independent as possible while offering useful data for a possibly highest number of theoretical approaches).

As suggested above, Table 14.2 displays a list of “Missing Categories”.

14.2.3 What the Syntactic Data Suggest

Annotation is still ongoing with about 15 h of spoken material annotated so far, resulting in 9,435 sentences (7,344 in the informal and 2,091 in the formal dialogues). The number of clauses found is 21,246 (17,204 informal and 4,042 formal). As a preliminary observation, the sentence/clause ratio shows a significant difference between the two kinds of dialogues: 2.34 (informal) and 1.93 (formal) suggesting that different communicative events may reflect different cognitive processes underlying language production: even though both event types were spontaneous, informal dialogues included shorter sentences (fewer clauses within a sentence) than in formal dialogues (short, one-word questions and answers representing a single clause as a complete sentence).

The two forms of dialogues also had different average duration (informal: 2435.4 ms, formal: 4000.73 ms). In addition to the obvious effect in event dynamics, formal dialogues with their significantly longer duration allow space for more complex or more complete structure generation.

Table 14.3 shows the features subcategorising incomplete clauses.

Fragmentation of spoken utterances is clearly demonstrated by the data: only about every fifth clause was found complete with NP-VP pair on the surface and no obligatory constituent missing. Interestingly, no difference was observed between formal and informal types of dialogues. Another kind of completeness was attributed to clauses with no NP-VP pair. These are mostly cases of short questions or one-word answers, such as “Where?”, “Me?”, another characteristic feature of spoken

Table 14.2 Missing categories

Values	Definition	Remarks
1	Nothing is missing	The clause is a complete clause
2	Head clause is missing	No clause can be identified as its head clause
3	Missing clause of coordination before or after the clause	The presence of a conjunction in the clause suggests coordination but no additional clause can be identified as its coordinated clause
4	Missing reference word	Missing relative pronoun
5	Missing conjunction	The interpretation of two adjacent clauses suggests that they are related either by coordination or subordination, but the appropriate conjunction is missing from the surface. (Perceived coherence is unmarked.) (It started raining. I went home.)
6	Missing grammatical subject	The interpretation of the clause points to a grammatical subject which is missing from the surface
7	Missing logical subject	The interpretation of the clause points to a logical subject which is missing from the surface
8	Missing predicate	The presence of a free adverbial with no further structural relations suggests a missing verb and its arguments
9	Missing object	A clause containing a verb whose mandatory object complement is missing from the surface
10	Missing adverbial complement	A clause containing a verb whose mandatory adverbial complement is missing from the surface
11	Missing attribute	A clause with a nominal phrase missing its mandatory attribute from the surface
12	Missing verb	One or more nominal phrases are interpreted as constituting a clause as arguments of a missing verb
13	Unfinished clause	In general, the case of corrections or aborted start: more than one obligatory structural unit (specified as 2 through 12 above) is missing
14	Complete, but no NP-VP structure	Sentential words: Szia! Hi! Affirmative answer: Igen. Yes. Jó. Good. Negative answer: Nem. No. Egyáltalán nem. Not at all. Uncertain answer: Talán. Perhaps. Single word question: Hol? Where? Clause starts with the conjunction mint such as Politeness marker: Conjunction/reference word: Sőt! Moreover!, És?, And?, Mert! (Just) Because!
15	Grammatical relationship inherently unmarked	Insertion

Table 14.3 Frequency of the annotated syntactic categories

Type of clause	Type of dialogue					
	All dialogues		Informal dialogues		Formal dialogues	
	Total	%	Total	%	Total	%
Grammatical subject is missing	4345	31.49	3178	29.51	1167	37.14
Complete, but no NP-VP structure	4144	30.03	3375	31.34	769	24.47
Nothing is missing	3422	24.8	2664	24.74	758	24.12
Unfinished clause	895	6.49	728	6.76	167	5.32
Logical subject is missing	387	2.8	274	2.54	113	3.6
Predicate is missing	286	2.07	214	1.99	72	2.29
Object is missing	147	1.07	102	0.95	45	1.43
Missing reference word	111	0.8	89	0.83	22	0.7
Missing coordinated clause	64	0.46	58	0.54	6	0.19
Head clause is missing	52	0.38	37	0.34	15	0.48
Missing conjunction	26	0.19	22	0.2	4	0.13
Missing adverbial complement	15	0.11	11	0.1	4	0.13
Verb is missing	10	0.07	10	0.09	0	0
Insertion	6	0.04	6	0.06	0	0
Missing attribute	0	0	0	0	0	0

utterances: short but incomplete is preferred to longer but complete. (Close to one-third of informal and one-fourth of formal clauses were of this kind, again showing event-type specific distribution.)

The grammatical subject was missing in more cases in the formal dialogues (in more than one third of formal clauses), but to no surprise: in Hungarian, the morphological marking of the subject is on the verb, and the additional marking of the subject with a noun/pronoun usually appears as redundant or communicatively marked. Informal dialogues often require less cognitive load, this may be a possible reason for its somewhat fewer cases in informal clauses.

It is interesting to see a sequence of how many clauses make a sentence: about 95 % of all sentences are made up of no more than 3 clauses both in formal and informal dialogues, suggesting that the cognitive task of building sequences of clauses may override the formal/informal distinction. Table 14.4 displays the longest single sequence detected for either formal or informal dialogues studied. Here 21 classes were found in an of the informal dialogue examined.

These data suggest that completeness/incompleteness, the most obvious feature of the fragmentation of spoken utterances is a meaningful marker of certain important cognitive processes within a dialogue. The nature of these processes can, however,

Table 14.4 The distribution of sequence of clauses within a sentence in formal and informal dialogues

# Clauses within the sentence	Informal dialogues		Formal dialogues	
	Total	%	Total	%
1	2933	80.18	688	81.52
2	289	7.9	82	9.72
3	163	4.46	24	2.84
4	92	2.52	26	3.08
5	44	1.2	9	1.07
6	37	1.01	9	1.07
7	24	0.66	2	0.24
8	24	0.66	2	0.24
9	17	0.46	0	0
10	7	0.19	0	0
11	6	0.16	1	0.12
12	4	0.11	0	0
13	4	0.11	0	0
14	2	0.05	0	0
15	1	0.03	0	0
16	2	0.05	0	0
17	2	0.05	1	0.12
18	3	0.08	0	0
19	0	0	0	0
20	3	0.08	0	0
21	1	0.03	0	0

vary, ranging from superficial topic management to rapid exchanges of turns to cognitive task management having priority over the substantial load of the syntactic requirement of sequential ordering of hierarchically ordered elements.

In any case, even if these data point to the importance of carrying out further investigations into the cognitive processes behind spoken fragmentation and syntactic incompleteness, analysis of spoken syntax cannot serve as the lone basis of reconstructing the missing elements either in linguistic structure or its functional interpretation. Researchers can expect to extend the possibilities of properly interpreting incomplete verbal expressions by considering additional nonverbal, multimodal data (including prosody and gesturing). This is what the subject of the next two sections will be.

14.3 Spoken Syntax Versus the Non-verbal Modalities of Prosody

Since due to the prevalent incompleteness of syntactic structure in spoken language the proper formal and contextual analysis of speech events is largely hindered, such an analysis needs to largely rely on further multimodal cues as well. Among the modalities it is prosody with its close relation to linguistic form that may appear to be especially helpful, but gestures usually accompanying a communicative event in one way or another are also important candidates to be considered.

14.3.1 Spoken Syntax Versus Prosody

The term speech prosody essentially covers the three independent modalities of duration (expressed by time), intonation (expressed by frequency) and intensity (expressed by energy). These modalities jointly, in an inseparable way contribute to the concept of prosody and, accordingly, to our perception and production of the nonverbal "overlay" of speech communication. Prosody conveys a number of important functions: it plays a structural role in the sequential and hierarchical organisation of elements within and across utterances, expresses semantic and pragmatic relations in the context of an interaction, and it reflects the cognitive state of the participants.

It is well known that the proper choice of prosody has its important function in communication to the extent that it may even override the default meaning of a verbal expression. It is well known that someone responding to a request by saying "Yes" does not necessarily mean consent or approval: one has to "listen" to how this "Yes" was pronounced in order to properly evaluate the intended meaning behind it. Prosody does not, however, convey intensions and emotions alone; being associated with words and groups of word (phrases) it is straightforward to try to associate prosody with syntactic structure as well. Due to the fact that syntax is incomplete and fragmented in many instances in spoken language, it might be useful to consider prosody as a cue to a more complete understanding of syntactic structure behind the given often fragmented sequence of words. This was one of the primary aims behind the annotation of the prosody of the HuComTech corpus as well. Based on the existing syntactic annotation researchers may identify the main prosodic characteristics of each of the clauses, including F0, intensity and emphasis. Although this research is evolving, the authors are able to report the initial results.

Based on Martens *Prosogram*[1] [15] an automatic annotation scheme of F0 was developed with essential contribution to design and programming by Szekrényes [16], who subsequently extended the script to handle intensity annotation as well. Initially the algorithm identified only three pitch trends, rising, falling and stagnant assigned to the annotated clauses, but the most recent version already includes five

[1] http://bach.arts.kuleuven.be/pmertens/prosogram/.

distinct F0-styles giving space for more subtle distinctions. The annotation labels contain these classification results of the stylized F0 curve (created with Praat [17] and Prosogram) calculating two main parameters: the duration and the amplitude of F0 trends.

Even though speech is primarily based on verbal utterances consisting of words and their linguistically structured sequences (sentences), they, being spoken realisations of mental entities are also characterized by properties pertaining to speech, in particular by prosody.

Prosody, of course, does not act alone: it fulfills its functions in relation to other modalities, both verbal (the linguistic content) and nonverbal (gestures). What makes this multimodality of human expression unique is that none of them can assume a single leading role: it always depends on many (often difficult if at all possible to predict) conditions. Therefore what science need consider is to first collect individual descriptive properties (what modality in what form and under what conditions appears where) and then try to consolidate such facts about all these modalities around well defined functions (such as those included in the theoretical generative model of communication outlined above). Finally, statistical and machine learning methodologies incorporating models and rules should undoubtedly play an important role in arriving at a broader understanding and practical implementation of the complex relation of bits and pieces of multimodal markers in eventual real life applications. In what follows the authors describe the elements regarding speech prosody as manifested in their ongoing work with the HuComTech multimodal corpus.

14.3.2 Duration

A given sound wave of speech is spread across time that is captured at various levels perceived and categorised in a binary way as either short or long, categories that can have a significance at various distinct levels. Accordingly, for a sound to be short or long usually has linguistic (phonological in particular) significance whose variation contributes to the semantics of a given word. The duration of a syllable, especially the sequence of syllables of similar or different, alternating durations constitutes to speech rhythm, a phenomenon that can be characteristic of both a given language and a given individual, speaker specific performance. Duration is characteristic of the non-sounding constituents of speech, pauses as well. The even/uneven distribution of pauses across words in an utterance can again have both a linguistic and performative significance, the latter being the cognitive offprint of a speaker's attitude to the wider environment beyond the words. A special kind of pauses appears between larger chunks of utterances, especially between clauses and sentences. It will be assumed that these silent pauses occur at periods of time when some cognitive processes of certain importance take place: they can be processes of forward planning, backward reflection or repair, processes whose results, in turn, can be reflected in the actual linguistic form of the given chunk (clause or sentence) of the utterance as well.

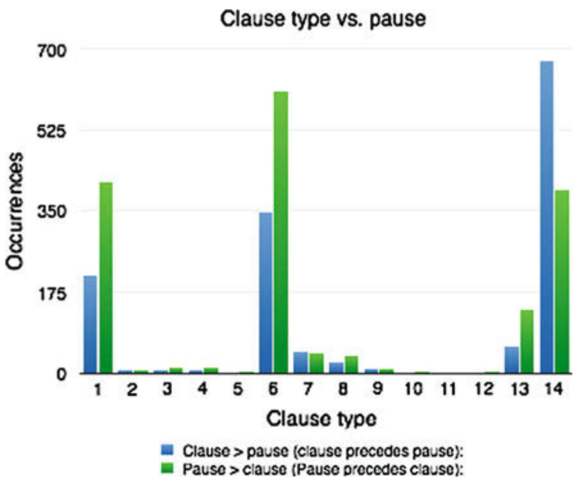


Fig. 14.2 The frequency of clause types associated with pauses

Therefore, when describing and studying the durational properties of speech prosody it is expected that one will be able to reach beyond the spoken words and grasp some important information about the cognitive state of the speaker, apparently hidden from the “naked ears”.

The annotation and analysis of silent pauses immediately preceding or following a clause as a component of a complex of multimodal markers serves this purpose. Given that as a result of linguistic analysis 14 different types of syntactic incompleteness were identified in clauses (see Table 14.1), the annotation aimed at finding out if any of these types are correlated with specific pausal durations, namely, whether these pauses can be considered as reflections of the cognitive processes as a result of which the given clause was formed as syntactically incomplete.

The authors have observed that not all clauses are preceded/followed by a pause: in most cases the first clause is preceded and the last clause is followed by a pause within a sentence, and the rest of the clauses may or may not have a pausal boundary.

Figure 14.2 shows the frequency of clause types associated with pauses:

According to the Figure, the following types are most frequently associated with pauses: Type 14 (complete, but no NP-VP structure, usually sentence words), Type 6 (grammatical subject missing), Type 1 (complete clause), and Type 13 (unfinished, more than one obligatory structural element missing).

As for Type 14, a pause significantly more often follows than precedes the clause: it confirms the intuition that sentence words like “Hi!”, “Good!”, “Not at all!” more often have the function of concluding some phase of an interaction rather than of starting it.

In contrast, Type 6 clauses are found to have a preference to be preceded rather than to be followed by a pause. A possible reason for this is that a significant share, 41.35 % of all Type 6 clauses occur as the first clause of a sentence, and as such, they are structurally “entitled” to be preceded by a pause. This number is only increased as a result of additional contextual-stylistic conditions. This twofold, structural and contextual effect on the placement of pauses preceding the clause cannot be counterbalanced by the sheer fact that, in principle, a clause with its grammatical subject missing on the surface (clause Type 6) should otherwise be closely associated with its preceding clause without any connecting pause. As an effect, such clauses should rather have a pause following the clause. The abstract, syntax based requirement placed on a clause with no surface grammatical subject is overridden by the cognitive shaping of an utterance: if a clause with a missing subject does occur at the beginning of a sentence (for example the first clause in a sequence of constituting clauses), the pause preceding it reflects the cognitive process of planning a chain of clauses as part of a longer utterance.

The case of clauses of Type 1 is fairly simple: being structurally complete, a preceding pause can be considered as a reflection of a proper and successful planning process. The fewer occurrences of pauses following this type of clause, on the other hand, can be due to the fact that complete clauses tend to be followed by another clause rather than be the final one of a sentence (that in turn would require a certain amount of pause): Type 1 clauses as the final clause of a sentence are found in 1,197 cases, only nearly one third of the 3,133 cases of clauses other than Type 1 as the final clause of a sentence.

Finally, Type 13 is the case of corrections or aborted start with more than one structurally obligatory syntactic element missing. The predominance of cases with a pause preceding the clause rather than following it is a reflection of the very nature of this clause type: what is behind the pause itself is the cognitive process of reflection or repair. That the actual process is incomplete is suggested by the very incompleteness of the clause itself.

That the position of a pause preceding versus following a given clause can indeed be indicative of some different functions is shown in Fig. 14.3.

The results of principal component analysis shown in Table 14.5 indicate that pauses can actually be classified as belonging to two groups based on whether they

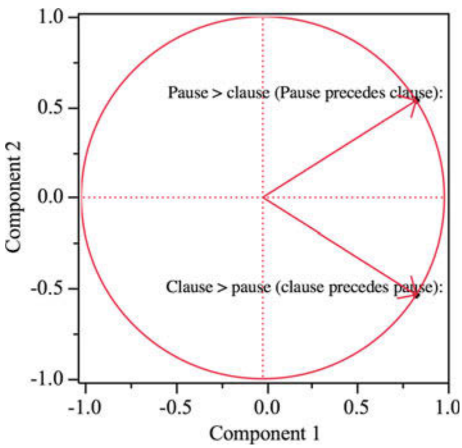


Fig. 14.3 The position of a pause preceding versus following a given clause

Table 14.5 The results of principal component analysis

Number	Eigenvalue	Percent	Cum percent
1	1,4261	71,305	71,305
2	0,5739	28,695	100,000

Table 14.6 The duration of the pause tends to be related to the type of the associated clause

Clause type	Clause > Pause (Clause Precedes Pause)				Pause > Clause (Pause Precedes Clause)			
	Duration	StDev	Min.	Max.	Duration	StDev	Min.	Max.
1	2166.24	1495.74	169	11608	2850.47	1666.68	228	14503
2	185.49	712.05	723	4149	265.11	1027.26	1191	5635
3	185.05	748.6	758	4455	261.5	699.96	593	4591
4	1241	831.37	1242	4167	2531.8	770.23	1541	4167
5	1464	1464	1464	1464	77.51	384.83	943	2667
6	2058.49	1394.87	179	12372	2566.28	1297.41	247	12344
7	1322.5	1628.36	320	6461	1389.35	1754.01	795	6303
8	1542.36	993.46	291	6576	723.15	1321.92	379	6708
9	1959.9	877.48	639	5027	753.11	1391.73	754	7618
10	748.5	502.75	393	1104	1939.5	999.84	393	2817
11	N.A.				N.A.			
12	1694				104.8	481.59	1747	3707
13	737.25	1048.87	219	4585	862.05	834.8	165	6627
14	656.61	289.82	133	4912	869.7	437.56	160	4352

occur before or after the clause. Following a discussion of the relation between clause type and pauses can lead us to hypothesise that this division of these pauses according to position have cognitive basis: planning and reflection.

Table 14.6 displays the duration of the pause that tends to be related to the type of the associated clause.

Due to the restricted amount of data available, the authors only consider here types 1, 6, 13 and 14. It can be noticed that, following average durations, they can be divided into two groups: Types 1 and 6 are associated with pauses with similar and rather long duration, whereas types 13 and 14 have significantly shorter pauses. This difference can be attributed to their functional difference: it is assumed that clauses of Types 1 and 6 are based on more complex cognitive processes: the planning of a longer stretch of sequences of clauses (Type 1) and the reflection (and linking) to a subject in a preceding clause (Type 6). Types 13 and 14 share in common the reflection of a different cognitive process: correction or aborted start (Type 13) practically refers to just part of what has been said before and does not require a total rethinking/replanning hence its shorter duration. Finally, a pause associated with a clause Type 14 just accommodates a short feedback without any more complex reflection or planning. It can also be noticed that in all these types of clauses the pause preceding the clause appears to be somewhat longer than the one following it. This may be assumed to be due to the cognitive component common in all these types: both forward planning and backward reflection may require more cognitive processing than the planning of other types of subsequent clauses.[2]

14.3.3 Intonation and Intensity

Intonation constitutes the perception of speech melody as a concatenation of subsequent F0-contours. Similarly to duration, changes of the F0-contour can affect the perception of sounds, syllables, words as well as whole utterances, they can show both linguistic and individual variation and express linguistic, semantic/pragmatic and cognitive functions. What is perceived is not identical with what is actually measured: it is already the result of certain cognitive classification, generalisation and filtering, and all this is governed by our overall experience often reducing our perception to "what we want or do not want to hear". Thus the challenge is how to process the highly complex sounding material (the waveform) so as to extract information relevant to perception. The authors decided to model at least the physical side of this perception by reducing the measured F0-values to stylised trends of melody along a predefined set of tonal spaces. The results were tested and implemented in the HuComTech corpus and are used in the present analysis as well. Since the original intention was to contribute to the study of the syntax-prosody interface in general and, through this to the description of the multimodality of communication, prosodic analysis has so far been done on material already having syntactic annotation. Therefore the annotated prosodic corpus at this point consists of about 10 h of sounding material.

The analyser aims at offering a relatively close approximation of the perception of speech melody. Regardless of the absolute pitch value or voice quality, people perceive the voice as rising, falling, or remaining at a certain level. This kind of stylisation is aimed at setting specific sub-ranges that are relative to the overall vocal range of the given speaker. This concept also accommodates sequences of pitch values as trend lines within or across individual sub-ranges. The following conventions are used.

Stylisation is preceded by (1) speaker selection from the dialogue (simultaneous speech is not included), (2) automatic silence detection, (3) removal of outlier pitch values. The definition of subranges are determined using the following threshold values (T1 through T4):

T1: Mean Stdev
T2: Mean Stdev/3
T3: Mean + Stdev/3

[2] The analysis did not specifically consider clauses of such complex syntactic relations as subordination, coordination and embedding. Therefore it is left open whether the cognitive processing of these types is reflected by pause durations corresponding to their complexity.

Fig. 14.4 Automatic F0 and intensity annotation (an example from ELAN)

T4: Mean + Stdev

where Mean = mean of F0 values, Stdev = normal distribution of F0 values.

The subranges are defined as:

L2: from smallest measured value to T1
L1: from T1 to T2
M: from T2 to T3
H1: from T3 to T4
H2: from T4 to greatest measured value

Automatic intensity annotation was carried out using essentially the same script adjusted to intensity values. An example of an automatic frequency and intensity annotation is shown as a screenshot from the ELAN annotation environment (Fig. 14.4).

The picture shows data for the clause *hát* 'well'. Its syntactic code is 1.0.0.0.0.0.14. meaning it is of Type 14, a clause consisting of a single sentence word. It is the single clause of the sentence s131 (the 131 sentence of the speaker at shortly after minute 10 of the recording. The voice range for the utterance is L2 throughout, consisting of two smaller stylised segments both within the voice range L2, with the given actual F0-values for the start and the end of the given segment. The verbal interpretation of the trend lines within the given ranges is *rise* (a steep increase of F0) followed by *upward* (a moderate increase of F0). According to the data energy was detected earlier than F0 resulting in three intensity segments with their subsequent stylised interpretation *rise* (a steep increase of energy), *fall* (a steep decrease of energy) and *descending* (a modest decrease of energy), all moving around the middle range (M) in both directions.

This detailed annotation of prosody can be especially useful for automatic speech synthesis, since the implementation of these stylised labels (useful for abstract, style based as compared to concrete, pattern based synthesis) can vary according to speaker or even pragmatic content. At the same time this labelling may also enhance speech recognition since the description of possible (correct) sequences of labels may assist in excluding eventual false positives, too. Of course, in order to do so one needs a database of matching words/phrases and F0, intensity and syntactic text data based on a larger set of similar occurrences. The annotation in the HuComTech corpus may serve as a possible source of such structured information in the future.

14.3.4 Establishing a More Comprehensive Relation Between Spoken Syntax and Prosody Beyond Duration

This collection of data can contribute to a more extended study of the relation between spoken syntax and prosody by including intonation and intensity as well. Accordingly, the question can also be asked how the intonation and intensity of certain kinds of utterances are realised at certain syntactic positions and how a variation of these parameters reflect a variation in syntax, semantics, pragmatics, even (in synchrony with other modalities) to the extent they offer an insight in the cognitive state of the speaker. A simple example is shown in Fig. 14.5.

The two sentences s51 and s52 read as “Well”, “I was still quite lazy in the elementary school.”, respectively, and the single word expression “Well,” of sentence s51 is followed without a pause by the second sentence s52 and a continuous intonation connects the two with a single continuous descending intonation pattern ranging over the border between the two. It can also be seen, however, that the relative increase of intensity at the same border is used to identify a syntactic boundary, a prosodic segmentation necessary for proper semantic processing.

Figure 14.6 represents the prosodic analysis of a more complex syntactic construction involving multiple cases of embedding.

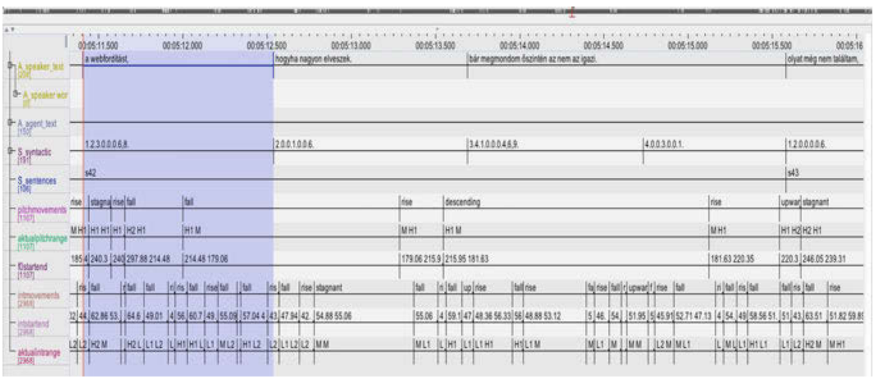


Fig. 14.5 Prosodic and syntactic annotation (an example from ELAN)

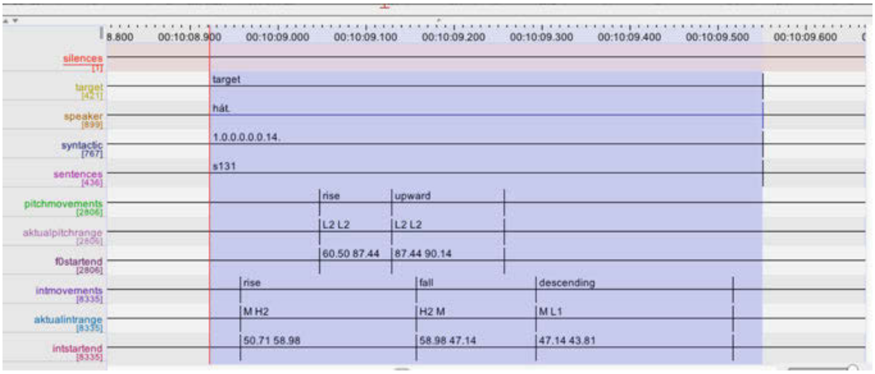


Fig. 14.6 Prosodic analysis of a more complex syntactic construction

The example consists of a single sentence s42 having four clauses, reading: "(I use) web translation", "if I am lost very much", "but I tell you honestly", "it is not the real thing". The syntactic annotation of the first clause, 1.2.3.0.0.0.6,8. indicates, among others, that the next clause is subordinated to it, similarly, the third clause, 3.4.1.0.0.0.4,6,9. indicates, among others, that the next clause is subordinated to it. The pitch movement, however, does not reflect this multiple subordination: a "fall" F0 segment ranges over the boundary between the first two clauses and a "descending" F0 segment does over the boundary between the second and the third. Instead, syntactic phrasing is again (at least partly) accompanied by intensity variation. This example draws again our attention to the fact that:

- prosodic structure does not necessarily follow syntactic structure, and
- a more realistic text-to-speech system should take it into account and build the actual prosodic shape of the flow of speech following nonlinguistic (pragmatic, intentional, emotional) criteria effective across sentence boundaries.

A richly annotated multimodal corpus should prove to be instrumental in obtaining the necessary amount of data supporting such generalisations.

14.4 Spoken Syntax Versus the Non-verbal Modality of Gesturing

The HuComTech corpus is extensively annotated for gestures with respect to physical attributes, emotions and pragmatic functions of communicative events [18] and can serve as a reliable source for the study of the relation between spoken syntax and gestures as well.

Multimodality of communication involves the complex interaction of verbal and nonverbal channels (modalities) of communication. Whereas it is generally true that its verbal, linguistic component carries an essential bulk of the message, it is also widely observed that the nonverbal component plays an important role, too. Certainly, people experience that nonverbal markers, especially those of gesturing often accompany speech with additional information about the context as well as the background knowledge, emotions and other attitudes of the participants. The authors also observe that these multimodal markers do not simply enhance the verbal content but can sometimes even contradict to the default meaning of a word or a whole utterance: by choosing the appropriate prosody and selecting specific gestures while answering to a question as “Yes”, the answer may convey either of two, logically contradictory senses: *affirmative* or *negative*. However, in addition to the obvious stylistic, semantic or pragmatic functions of gesturing in the multimodal realisation of a communicative act, one can go further by suggesting that gesturing does not only accompany the verbal component to support or modify its content, but it can also be the direct reflection of certain cognitive processes as realised in the given linguistic form. Namely, it is assumed that the obvious fragmentation and incompleteness of the syntax of a number of spoken utterances is the result of certain cognitive processes involved in planning, correction and back-channelling that are reflected in the involuntary gesturing accompanying speech.

14.4.1 Gestures and Pauses

The HuComTech corpus is (being) annotated for a wide variety of gestures with the overall aim to capture the essential underlying semantic and pragmatic components of a communicative event. By pairing gestures with specific syntactic annotation, however, the authors wish to learn about the cognitive functions of gestures as reflected in linguistic form. A short introduction to ongoing work on this issue below will show the possibilities offered by this approach. In particular, one will ask the question how and to what extent facial expressions aligning with various syntactic clause types can actually be the reflection of certain cognitive processes behind the given linguistic form. One will consider the alignment with gestures of the same clause types with syntactic incompleteness already discussed in relation to prosody earlier. The gestures to be studied in their alignment with linguistic form will be related to the category of head movement, namely *nodding*, *raising*, *lowering*, *shaking*, *turning* as well as *sideways*.

The following two configurations were considered: (a) a pause is surrounded by a gesture (the pause falls within the duration of the given gesture) which, in turn is left aligned with a clause (the begin and end time of the gesture precedes the begin and end time of the clause: the gesture starts before the clause but ends before the clause ends), and (b) a pause is surrounded by a gesture (the pause falls again within the duration of the given gesture) which, in turn is right aligned with a clause (the begin and end time of the gesture follows the begin and end time of the clause: the gesture starts after the beginning of the clause but extends over the right boundary of the same clause).

The first general observation to be made refers to the relation between pause and gesturing. It appears that, confirming earlier findings, gestures tend largely to be associated with the verbal content: the number of gestures inside pauses is quite small, regardless of the relative sequence of the gesture and the pause, shown in Figs. 14.7 and 14.8 (the character ">" indicating surface precedence).

The general role of aligning pauses and clauses can also be extended to include the absolute pitch value or voice. Similarly gestures can align with pauses and tend to reflect the cognitive processes of planning, replanning or correction. Alternatively gestures directly associated with clauses without pauses are considered reflections of cognitive processes associated with the performance of proper linguistic form.

14.4.2 The Temporal Alignment of Gestures and Clauses

The temporal alignment of head gestures with clauses, in other words, whether the start of a gesture precedes or follows the start of a clause may also offer the chance to make interesting assumptions about the function of gestures beyond rhythmically

Gesture > clause	type 1		type 2		type 3		type 4		type 5		type 6		type 7		type 8		type 9		type 10		type 11		type 12		type 13		type 14		type 15	
gesture aligned with	C	P	C	P	C	P	C	P	C	P	C	P	C	P	C	P	C	P	C	P	C	P	C	P	C	P	C	P	C	P
nod	70	11	1	0	1	0	2	0	0	0	95	20	12	0	5	2	4	0	0	0	0	0	0	0	4	0	54	11	0	0
raising	96	31	5	1	0	0	4	0	3	0	159	79	9	3	12	3	5	3	0	0	0	0	0	0	10	3	40	9	0	0
lower	121	79	1	0	5	5	5	3	1	0	120	52	11	22	9	3	5	1	1	0	0	0	1	0	12	4	42	21	1	0
sideways	7	1	0	0	0	0	1	0	0	0	10	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	2	0	0
shake	33	8	0	0	0	0	0	0	0	0	67	5	7	0	1	0	4	1	1	0	0	0	0	0	5	1	22	8	0	0
turn	1	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	9	

Fig. 14.7 Gestures >clause type: gestures aligned with clauses (C) and pauses (P)

Clause > gesture	type 1		type 2		type 3		type 4		type 5		type 6		type 7		type 8		type 9		type 10		type 11		type 12		type 13		type 14		type 15	
gesture aligned with	C	P	C	P	C	P	C	P	C	P	C	P	C	P	C	P	C	P	C	P	C	P	C	P	C	P	C	P	C	P
nod	59	19	1	0	0	0	0	0	0	0	81	23	6	0	7	4	5	1	0	0	0	0	0	0	15	0	126	21	0	0
raising	110	44	2	0	2	1	2	2	0	0	162	62	17	17	12	2	5	0	1	0	0	0	0	0	14	3	57	12	1	1
lower	98	55	3	1	1	0	2	0	2	0	124	99	14	6	4	3	2	0	0	0	0	0	0	0	14	8	36	5	0	0
sideways	5	0	0	0	0	0	0	0	1	0	9	5	0	0	2	0	0	0	0	0	0	0	0	0	0	0	2	0	0	
shake	61	10	1	1	2	1	0	0	0	0	72	9	9	2	5	2	1	1	0	0	0	0	0	0	5	0	39	9	0	0
turn	4	0	0	0	0	0	0	0	0	0	1	1	0	0	5	0	0	0	0	0	0	0	0	0	0	0	1	0	0	

Fig. 14.8 Clause type >gestures: gestures aligned with clauses (C) and pauses (P)

following the flow of speech. Consider Figs. 14.9, 14.10, 14.11, and 14.12 with precedence defined as $\Delta t_{begin} < 400$ ms:

By examining the above tables, the observer will notice that Type 1 clauses (those with no missing obligatory syntactic element), tend to be rather preceded than followed by head gestures. Alternatively the other types investigated here appear to make little difference. There can be at least three reasons for that: first, a syntactically complete clause usually represents the first clause of an utterance and the given head gesture can reflect the effort of making such a start. Second, the utterance often also ends with this complete clause and thus is followed by a pause, and, as pointed out earlier, gestures predominantly do not occur within pauses. Third, such a complete clause can be followed by a sequence of further clauses with no pause between them but also with no apparent F0-segmentation at clause boundaries. This is the case when as observed in the authors discussion of intonation and intensity speech prosody does not directly follow syntactic structure (in fact is proves to be more "rudimentary" in this respect).

In contrast, a roughly similar distribution of head gestures is shown preceding and following the beginning of clauses of types 6, 13 and 14. These are characteristic of the cognitively more marked syntactic types (Type 6: grammatical subject missing, Type 13: more than one obligatory syntactic constituent missing, Type 14: the overall NP-VP structure missing). The authors suggest that this similar distribution of gestures before/after the beginning of a clause can be the gestural indication of this cognitive complexity stretching across the whole of the clause and extending even beyond. The complexity of Type 6 clauses is further supported by the fact that roughly one fourth, one third of all head gestures used in all syntactic types belong to this type: the lack of the grammatical subject in a clause is a quite strong call for the need for some other (nonlinguistic) means to connect the given clause with the rest of the utterance in order to keep the bits of information conveyed by the individual clauses connected. The frequent use of head gestures at both boundaries of classes of this type can effectively serve this purpose.

Finally the reader may notice that around half of all instances of nodding are associated with clause Type 14. This high frequency can be justified by the communicative functions of his type: being short clauses consisting of mostly sentence words they express affirmation, negation, greetings, short questions or answers with the pragmatic function of turn-give often requiring a response from the interlocutor. As such, this dynamics of the conversation is effectively supported by the intensive use of head gestures.

The above tables also indicate to what extent a gesture is included in a clause or, on the other way, a clause is included in the gesture, practically showing the functional scope of a given gesture. Whereas leaving at this point these data for further analysis by the reader, the authors only mention one observation: nodding strongly tends to include the clause in its duration (nodding starting before and ending after the clause) in clause Type 14 but not in other types. This difference is again due to the communicative functions of Type 14 clauses reviewed above.

The authors summarise their observations by concluding the relation between syntactic structure and head gesturing using the following statements. Gestures in

clause type 14						
gesture: headshift	gesture > clause (% across all clause types)	gesture > clause # cases)	clause inside gesture (# cases)	clause > gesture (% across all clause types)	clause > gesture (# cases)	gesture inside clause (# cases)
nod	45.57	72	70	50	102	0
raise	17.11	58	27	23.21	68	7
lower	21.07	71	40	20	68	16
sideways	13.2	52	30	18.64	74	15
shake	25	24	23	25.51	25	1
turn	16.29	86	57	18.91	104	5

Fig. 14.9 Clause type 14

clause type 13						
gesture: headshift	gesture > clause (% across all clause types)	gesture > clause (# cases)	clause inside gesture (# cases)	clause > gesture (% across all clause types)	clause > gesture (# cases)	gesture inside clause (# cases)
nod	4.43	7	4	4.41	9	0
raise	7.08	24	8	7.51	22	6
lower	4.15	14	4	8.53	29	8
sideways	5.84	23	6	6.8	27	7
shake	4.17	4	2	3.06	3	0
turn	7.01	37	10	7.45	41	11

Fig. 14.10 Clause type 13

clause type 6						
gesture: headshift	gesture > clause (% across all clause types)	gesture > clause (# cases)	clause inside gesture (# cases)	clause > gesture (% across all clause types)	clause > gesture (# cases)	gesture inside clause (# cases)
nod	25.95	41	11	19.61	40	4
raise	31.56	107	10	32.42	95	36
lower	29.67	100	3	33.82	115	34
sideways	31.22	123	3	36.52	145	42
shake	29.17	28	3	28.57	28	0
turn	34.47	182	13	32.73	180	65

Fig. 14.11 Clause type 6

clause type 1						
gesture: headshift	gesture > clause (% across all clause types)	gesture > clause (# cases)	clause inside gesture (# cases)	clause > gesture (% across all clause types)	clause > gesture (# cases)	gesture inside clause (# cases)
nod	10.13	16	4	5.26	24	2
raise	18.58	63	5	8.46	74	30
lower	21.96	74	6	12.23	81	29
sideways	25.63	101	8	15.94	82	21
shake	17.71	17	0	0	16	1
turn	20.45	108	11	18.25	121	32

Fig. 14.12 Clause type 1

general, also including head gestures go beyond just following linguistic form. They make up an independent modality that interacts with other modalities such as the verbal content and speech prosody teamed up in order to effectively convey various kinds of information. This information is related to the communicative event itself and is specific to its formal structure (linguistic, semantic-logical) and to its communicative (pragmatic) function. Further, this information is also related to the participants of the given communicative event and is specific to their cognitive, psychological and social state. Gestures alone do not have exclusivity in expressing any of these functions, but forming part of a multimodal complex they are in fact indispensable in making an effective communicative event. In order to understand their individual contribution to delivering a given pragmatic information this complexity needs to be studied to determine all possible connections (not as isolated incidences). When completed in all its envisaged aspects of multimodality, the HuComTech corpus is expected to effectively promote research and development along the way to achieving this goal.

14.5 Conclusions

Participants of various events of human-human communication perceive these events holistically, noticing, observing and interpreting their verbal and non-verbal components as inseparable ingredients of a larger picture. However, in order to follow the technological requirements of human-machine interaction these components need to be treated individually. The challenge then is how to reconstruct this holistic view of an event from the individually treated single components. A novel theoretical-technological model of communication is proposed for this purpose: any communicative event is generated from primitives at distinct modular levels and modality specific markers are mapped onto these primitives as surface realization of an abstract communicative structure. It is shown how the study of certain markers contributes to a better understanding of communicative primitives and to the multimodal understanding of an event as a whole: it is found that the understanding of an incomplete syntactic structure can be enhanced in combination with aligning prosodic markers and gestures of facial expression. The conclusion is, however, that the combined consideration of aligning multimodal markers only increases the probability of a successful interpretation, whereas such calculations will always leave room for online corrections: communication is an act of cooperation in which several factors are left undefined at the outset requiring clarifications through constant negotiations between the parties all along the communicative event. A final note: even if technologically challenging, some of the results of multimodal alignment discussed in this chapter are suitable in their present form already to be implemented in a multimodal application of synthesis/generation and could enhance the effectiveness of a TTS system. Analysis/parsing, however, is even more challenging than synthesis/generation: even if some of the authors findings can already bring us closer to a more successful parsing of spoken material, in order to better understand the fragmentation and incompleteness of spoken linguistic form additional steps are needed to be taken to further integrate emerging knowledge about spoken syntax with the various other verbal and nonverbal aspects of speech communication.

Future plans of the HuComTech project wish to address the core of these challenges: even though temporal alignments and overlaps of markers across modalities offer invaluable cues to existing machine learning techniques, a more complete insight into the real nature of multimodal communication can essentially be gained by considering the sequences of co-occuring (aligning or overlapping) markers as temporal patterns across longer stretches of communicative events and by mapping these formal temporal patterns onto the corresponding patterns of communicative functions associated with them. By capturing these temporal patterns of sequences of pairs of markers and their functions one can eventually capture the dynamics of human communication, the feature probably most importantly underlying enhanced future systems of human-machine interaction. Public access[3] to HuComTech's still expanding data is aimed at further promoting these goals.

Acknowledgments The research presented in this chapter was partly supported by project TÁMOP 4.2.2-C/11/1/KONV-2012-0002. Further support was received from NeDiMAH (Network for Digital Methods in the Arts and Humanities), a cross-European project of the European Science Foundation.

References

1. Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G.: Robotics, Modelling, Planning and Control. Springer, New York (2010)
2. Markou, M., Singh, S.: Novelty detection: a review–part1: statistical approaches. Signal Process. **83**, 2481–2497 (2003)

[3]The HuComTech corpus is available from two locations: https://corpus1.mpi.nl/ds/imdi_browser?openpath=MPI1761660%23 (Nijmegen) and https://clarin.nytud.hu/ds/imdi_browser?openpath=MPI13%23 (Budapest).

3. Kube, C.R., Zhang, H.: Task modelling in collective robotics. Auton. Robots **4**, 53–72 (1997)
4. Kasher, A.: Modular speech act theory: research program and results [collection on the foundations of speech act theory]. In: Tsohatzidis, S.L.T. (ed.) Foundations of Speech Act Theory Philosophical and Linguistic Perspectives, pp. 312–322. Routledge, New York (1994)
5. Hunyadi, L.: Multimodal human-computer interaction technologies. Theoretical modeling and application in speech processing. Argumentum **7** 313–329 (2011)
6. Hunyadi, L., Földesi, A., Szekrényes, I., Staudt, A., Kiss, H., Abuczki, Á., Bódog, A.: Az ember-gép kommunikáció elméleti-technológiai modellje és nyelvtechnológiai vonatkozásai [a theoretical-technological model of human-machine communication and its relation to language technology]. Általános Nyelvészeti Tanulmányok **XXIV** (2012) 265–310
7. Hunyadi, L., Szekrényes, I., Borbély, A., Kiss, H.: Annotation of spoken syntax in relation to prosody and multimodal pragmatics. Cognitive Infocommunications (CogInfoCom), pp. 537–541 (2012)
8. Rizzolatti, G., Arbib, M.A.: Language within our grasp. Trends Neurosci. **21**(5), 188–194 (1998)
9. Gallese, V., Fadiga, L., Fogassi, L., Rizzolatti, G.: Action recognition in the premotor cortex. Brain **119**(Pt 2), 593–609 (1996)
10. Willems, R.M., Hagoort, P.: Neural evidence for the interplay between language, gesture, and action: a review. Brain Lang. **101**(3), 278–289 (2007)
11. Willems, R.M., Ozyurek, A., Hagoort, P.: Seeing and hearing meaning: ERP and fMRI evidence of word versus picture integration into a sentence context. J. Cogn. Neurosci. **20**, 1235–1249 (2008)
12. Willems, R.M., Hagoort, P.: Hand preference influences neural correlates of action observation. Brain Res. **1269**, 90–104 (2009)
13. McNeill, D.: Hand and Mind: What Gestures Reveal about Thought. University of Chicago Press, Chicago (1992)
14. Kiss, H.: A hucomtech audio adatbázis szintaktikai szintjének multimodális vizsgálata [the multimodal study of the syntactic level of the hucomtech audio database]. MSZNY **2014**, pp. 27–38 (2014)
15. Alessandro, C., Mertens, P.: Prosogram: semi-automatic transcription of prosody based on a tonal perception model. In: Proceedings of the 2nd International Conference of Speech Prosody pp. 23–26 (2004)
16. Szekrényes, I.: Annotation and interpretation of prosodic data in the hucomtech corpus for multimodal user interfaces. J. Multimodal User Interfaces **8**, 143–150 (2014)
17. Boersma, P.: Praat, a system for doing phonetics by computer. Glot Int. **5**(9/10), 341–345 (2001)
18. Abuczki, Á.: A multimodal analysis of the sequential organization of verbal and nonverbal interaction. Argumentum **7**, 261–279 (2011)

Index

J.W. Tweedale et al. (eds.), *Intelligent Decision Technology Support in Practice*, Smart Innovation, Systems and Technologies 42,
DOI 10.1007/978-3-319-21209-8