

Supporting Authoring of Adaptive Hypermedia

By

Maurice Hendrix

A thesis submitted in partial fulfilment of the requirements
for the degree of
Doctor of Philosophy in Computer Science

Supervisor: Dr A.I. Cristea

University of Warwick, Department of Computer Science

March 2010

THE UNIVERSITY OF
WARWICK

Table of Contents

Table of Contents	II
List of Figures	IX
List of Tables	XIII
Acknowledgements	XIV
Declarations	XV
Abstract	XIX
List of Abbreviations and Acronyms	XX
1 Introduction	1
1.1 The Problem	2
1.2 Objectives	2
1.2.1 Research Questions	3
1.3 Thesis Outline	4
2 Background Research	6
2.1 E-Learning and standards	7
2.1.1 Metadata standards IEEE Learning Object Metadata: LOM	7
2.1.2 Dublin Core	7
2.1.3 IMS Learner Information Packaging (IMS-LIP)	8
2.1.4 IMS Content Packaging (IMS-CP)	8
2.1.5 Sharable Content Object Reference Model (SCORM)	9
2.1.6 IMS Vocabulary Definition Exchange (IMS-VDEX)	10
2.1.7 IMS Question and Test Interoperability (IMS-QTI)	11
2.1.8 IMS-Learning Design (IMS-LD)	12
2.1.9 Learning Management Systems (LMSs)	12
2.2 Semantic Web/Desktop	13
2.2.1 Resource Description Framework and Resource Description Framework (RDF and RDFS)	13
2.2.2 Web Ontology Language (OWL)	15
2.3 Adaptive Hypermedia: Brusilovsky's Taxonomy	16

2.4 Models for Adaptive (Educational) Hypermedia.....	18
2.4.1 The Dexter Hypertext Reference Model.....	18
2.4.2 AHAM.....	19
2.4.3 Munich Adaptive Hypermedia Reference Model	20
2.4.4 Web Modelling Language (WebML)	21
2.4.5 The Goldsmiths Adaptive Hypermedia Model (GAHM)	21
2.4.6 The XML Adaptive Hypermedia Model (XAHM).....	22
2.4.7 The Generic Adaptivity Model (GAM)	22
2.4.8 Layered WWW AHS Authoring Models and their Corresponding Operators (LAOS)	23
2.4.9 The Layers of Adaptation Granularity (LAG) Framework.....	25
2.4.10 Comparison.....	26
2.5 Authoring Adaptation Strategies	32
2.5.1 The LAG Language	33
2.5.2 LAG-XLS	37
2.5.3 GRAPPLE Adaptation Language (GAL)	38
2.6 Authoring Content for Adaptation.....	39
2.6.1 Common Adaptation Format (CAF).....	39
2.6.2 Ontology Based Content Description	43
2.7 Adaptive Hypermedia Authoring Systems	43
2.7.1 My Online Teacher (MOT).....	44
2.7.2 Adaptive Hypermedia for All (AHA!).....	48
2.7.3 Learning Activity Management System (LAMS).....	50
2.7.4 Activity Course Construction Toolkit (ACCT)	51
2.8 Related Adaptive Hypermedia and E-Learning Projects	51
2.8.1 ProLearn	51
2.8.2 Adaptive Learning Space (ALS).....	52
2.8.3 Generic Responsive Adaptive Personalised Learning Environment (GRAPPLE)	53
3 Automatic Content Addition	55

3.1 Authoring and Learning Scenario.....	56
3.2 The Approach.....	58
3.2.1 Metadata Schemas	59
3.2.2 Enriching the Domain Model and Goal Model with Metadata.....	60
3.2.3 Concept-oriented Relevance Ranking Method	60
3.2.4 Article-oriented Relevance Ranking Method	61
3.2.5 Sets versus Multisets in Computation of Relevance Ranking	62
3.2.6 Allowing Duplicates Among Sibling Concepts in the Concept Hierarchy	62
3.2.7 Adding Metadata as Separate Concepts or as Attributes	63
3.2.8 Handling Flexible Schemas	63
3.3 First Prototype	72
3.4 Evaluations	74
3.4.1 Small Scale Qualitative Analysis.....	74
3.4.2 Qualitative Analyses	75
3.5 Second Prototype.....	82
3.6 Evaluation of second prototype	86
3.6.1 Evaluation Setup.....	86
3.6.2 Hypotheses.....	86
3.6.3 Results and Discussion	87
3.7 Related Research	90
3.8 Conclusions	90
4 Adaptation Languages and Standards, a Comparative Analysis between IMS-Learning Design and Adaptive Educational Hypermedia	93
4.1 Representation and Semantics	94
4.2 Comparison Purpose and Criteria.....	95
4.2.1 IMS-Learning Design	95
4.2.2 Comparison of high-level features of LAOS and IMS-LD.....	96
4.2.3 Comparison of low-level Features of LAOS and IMS-LD.....	98
4.3 Adaptation	99
4.3.1 Adaptation Engineering Taxonomy	99

4.3.2 Pedagogy-based Adaptation Classification.....	103
4.4 Results Summary according to Comparison Criteria	108
4.5 Related Research	109
4.6 Conclusions	111
5 Improved Reuse Capabilities of Adaptation Strategies	113
5.1 Adaptation Engine Issues and Limitations	114
5.2 A Case Study	115
5.3 Solutions to Adaptation Engines Issues and Limitations.....	116
5.4 Meta-level Addition to LAG	118
5.5 Related Research	121
5.6 Conclusions	121
6 Graphical Authoring with CAM.....	123
6.1 Motivating Scenario	124
6.2 Pedagogical Strategies in the GRAPPLE Authoring Tool (GAT).....	127
6.2.1 Rollout	129
6.2.2 Depth First	131
6.2.3 Breadth First	132
6.2.4 Visual – Verbal.....	132
6.2.5 Beginner – Intermediate – Advanced	133
6.2.6 Relatedness	135
6.2.7 Conclusions	135
6.3 Design of the Concept Adaptation Model (CAM) Model	136
6.3.1 CAM.....	137
6.3.2 Link with LMSs.....	138
6.3.3 Relations between the Layers of the CAM model	139
6.3.4 Relations between the CAM and the Adaptation Engine.....	139
6.3.5 Model Verification.....	139
6.3.6 CRT and DM Combination Problems.....	141
6.4 Design of the Adaptation Languages.....	142
6.4.1 DM Language	143

6.4.2 CRT Language.....	147
6.4.3 CAM Languages.....	152
6.5 Design of the GRAPPLE Authoring Tool Shell and Web Services	158
6.6 Design of the CAM Component.....	160
6.7 Evaluation of GRAPPLE Authoring Tool.....	161
6.7.1 Quantitative Evaluation	163
6.7.2 Results of Quantitative Evaluation	166
6.7.3 Qualitative Evaluations.....	172
6.7.4 Results of qualitative evaluations	175
6.7.5 Qualitative Evaluation Findings	180
6.7.6 Summary of Evaluation Findings	184
6.8 Related Research	185
6.8.1 Competence Based Learning Tools	186
6.8.2 Mash-ups of Learning Environments: ROLE	188
6.8.3 My Online Teacher 2.0 (MOT2.0).....	188
6.8.4 My Online Teacher 3.0 (MOT3.0).....	189
6.8.5 GRAPPLE Simulation Authoring Tool	189
6.8.6 GRAPPLE Virtual Reality (VR) Authoring Tool.....	191
6.9 Conclusions	193
7 Summary and Conclusions.....	194
7.1 Introduction	194
7.2 Automatic Content Addition	197
7.2.1 Summary of Contributions.....	197
7.2.2 Limitations.....	197
7.2.3 Future Directions	198
7.3 E-Learning Standards	199
7.3.1 Summary of Contributions.....	199
7.3.2 Limitations.....	200
7.3.3 Future Directions	200
7.4 Improved reuse of adaptive strategies	200

7.4.1 Summary of Contributions.....	200
7.4.2 Limitations.....	201
7.4.3 Future Directions	201
7.5 Graphical Authoring: CAM.....	202
7.5.1 Summary of Contributions.....	202
7.5.2 Limitations.....	203
7.5.3 Future Directions	203
7.6 Final Conclusions	204
Appendix I. GAT User Guide	207
I.I. Domain Model Component.....	209
I.II. Concept Relationship Type Component	220
I.III. Concept Adaptation Model Component	227
I.III.I. Inserting CRTs	228
I.III.II. Filling Sockets with Concepts.....	231
I.III.III. Editing CRTs and sockets	234
I.III.IV. Deleting Concepts and CRTs.....	235
I.IV. GRAPPLE Authoring Tool Tutorial.....	236
I.IV.I. Creating a Domain Model – DM component.....	236
I.IV.II. Creating a Teaching Strategy – CRT component.....	241
I.IV.III. Creating an Adaptive Course – CAM component	247
Appendix II. XML Schema of CAM Internal Language.....	252
Appendix III. CAM Example	258
Appendix IV. XML Schema of CAM External Language	272
Appendix V. Information for Integrating GAT Component Into the GAT Shell	277
V.I. assets.....	277
V.II. net	277
V.III. org.grapple_authoring_tool.authoring.events	278
V.III.I. onLoadModel.....	278
V.III.II. onSaveModel	278
V.III.III. onNewModel	278

V.III.IV.	ononMenuItemClick	278
V.III.V.	onUpdateCurrentModel	279
V.III.VI.	onSetToolbar	279
V.IV.	org.grapple_authoring_tool.authoring.ui	279
V.V.	org.grapple_authoring_tool.authoring.utils	282
V.V.I.	DrawingUtils	282
V.V.II.	KeyUtils	282
V.VI.	org.grapple_authoring_tool.authoring.CopyPaste	282
V.VI.I.	Concepts	283
V.VI.II.	CRTs	283
V.VII.	org.grapple_authoring_tool.authoring.GrappleVgraph	283
V.VIII.	org.grapple_authoring_tool.authoring.crtshapes	283
V.IX.	org.grapple_authoring_tool.authoring.socketshapes	283
V.X.	org.un	283
Appendix VI.	GAT Qualitative Online Survey	284
VI.I.	Questions asked	284
VI.II.	Full data collected	284
Appendix VII.	GAT Quntitative Online Survey	293
VII.I.	Questions asked	293
VII.II.	Full data collected	293
VII.II.I.	Internal respondents	294
VII.II.II.	External respondents	295
References	296

List of Figures

Figure 1 Schematic description of the IMS-VDEX format [65].	11
Figure 2 Brusilovsky's taxonomy of Adaptive Hypermedia technologies [21].	17
Figure 3 The five level AHS authoring model and three layers of adaptation	26
Figure 4 Adaptation content example: questions and answers	42
Figure 5 Attribute editing and reading	44
Figure 6 Use of weights and labels in MOT as metadata	45
Figure 7 UML diagram of MOT database	46
Figure 8 The AJAX-based layout of MOT3.0.	46
Figure 9 Programming Environment for Adaptation Language (PEAL).	48
Figure 10 AHA! Graph Author Tool.	49
Figure 11 An example of a course in LAMS, image source: http://wikipedia.org	50
Figure 12 System overview of Enricher.	58
Figure 13 RDF schema of Semantic Desktop metadata in Beagle++	60
Figure 14 Right side, result with 'add metadata as separate concepts', left: without	63
Figure 15 RDF malleable schema of directory data	65
Figure 16 CAF loading; options and ranking method selection screen	73
Figure 17 Resource selection screen	74
Figure 18 SUS score for the three systems.	81
Figure 19 New prototype is accessed via enrich link in Goal Model view	83
Figure 20 Resource selection	84
Figure 21 Options and ranking method selecting screen.	85
Figure 22 Deployment screen	86
Figure 23 A manifest example of a UOL	96
Figure 24 System setup of template LAG Pre-processor.	118
Figure 25 Instantiated CRTs in the CAM for Planet and Star and two of the concrete planets and stars.	126
Figure 26 Generic relation between the general concepts Star and Planet and the concrete star and planet Sun and Jupiter.	126

Figure 27 Showafter relationship	130
Figure 28 'Countaccess' relationship	130
Figure 29 The main relation implementing the 'Depth First'	131
Figure 30 The main relation implementing the 'Breadth First'	132
Figure 31 Visual vs. verbal	133
Figure 32 Beginner - intermediate - advanced	134
Figure 33 A socket with one concept	153
Figure 34 A socket with multiple concepts	153
Figure 35 A CRT.....	154
Figure 36 Authoring tool.....	161
Figure 37 SUS score GAT	167
Figure 38 Knowledge Domain Model.....	186
Figure 39 GRAPPLE authoring tool.	208
Figure 40 Opening the GAT tool.	209
Figure 41 Save a new Domain Model.	210
Figure 42 Create a new Domain Model.	210
Figure 43 Definition of a new Domain Model.	211
Figure 44 Adding a new concept.....	212
Figure 45 Creation of a new concept.....	212
Figure 46 Resource addition.	213
Figure 47 Resource window filling.....	213
Figure 48 Astronomy Domain Model with only one concept Star and a related resource.	214
Figure 49 Astronomy Domain Model with four concepts, Star, Planet, Satellite and System.	215
Figure 50 Relationship creation.	216
Figure 51 Relationship creation by using the New button.	216
Figure 52 Relationship definition.....	217
Figure 53 Concepts with relationships.	218
Figure 54 How to modify or delete relationships.....	218
Figure 55 New concepts and relationships.....	219

Figure 56 Creation of a new type of relationship.	219
Figure 57 Final version of the Astronomy Domain Model.	220
Figure 58 General-Tab: General properties can be specified.	222
Figure 59 Meta-Info Tab: Meta information can be specified here.....	223
Figure 60 GAL Code Tab: The GAL code can be input here.	224
Figure 61 User Model Tab: User model variables used in the GAL code can be specified here.....	224
Figure 62 Constraints Tab: Constraints can be specified here.	226
Figure 63 DM-Relations Tab: Relations to Domain Model can be specified.	227
Figure 64 Opening the GAT tool.	227
Figure 65 Creating a new CAM.	228
Figure 66 Inserting a CRT instance into a CAM.....	228
Figure 67. Choosing a CRT to insert.....	229
Figure 68 The prerequisite CRT, which has 2 sockets, inserted into the CAM.	229
Figure 69 Copy a CRT instance. Other actions are delete and cut (copy then delete).	230
Figure 70 Past CRT is now enabled.	230
Figure 71 Copy concepts from a socket in a CAM window.....	231
Figure 72 Pasting Concepts into sockets.	231
Figure 73 Socket editing screen, providing an alternate place to paste concepts and an interface to insert concepts.	232
Figure 74 Selection interface for the Domain Model the concept comes from.	232
Figure 75 Selection of the Concept.	233
Figure 76 Insert External Location.....	233
Figure 77 Two sockets containing the concepts Star	234
Figure 78 CRT editing screen.	235
Figure 79 Deleting all concepts from a socket.	235
Figure 80 Opening the GAT tool.	236
Figure 81 Save a new Domain Model.	237
Figure 82 Definition of a new Domain Model.	237
Figure 83 Adding a new concept.....	238

Figure 84 Creation of a new concept.....	238
Figure 85 Resource addition.	239
Figure 86 Resource window filling.	239
Figure 87 Inserting is-a relation between Sun and Star.....	240
Figure 88 Relationship definition.....	241
Figure 89 Solar System DM.....	241
Figure 90 Creating new prereq CRT.	242
Figure 91 CRT general tab.	242
Figure 92 CRT is based on the Knowledge variable.	243
Figure 93 Deselect loops allowed.	244
Figure 94 GAL code for prereq.....	244
Figure 95 Creating the start CRT.	245
Figure 96 The knowledge variable is also used by the start CRT.	245
Figure 97 The GAL code for the start CRT assigns 100 to the knowledge for all concepts in the start_socket.....	246
Figure 98 The knowledge propagation CRT.	246
Figure 99 Also this CRT uses the knowledge variable.	247
Figure 100 GAL code for the knowledge propagation CRT.	247
Figure 101 Creating a new CAM.	248
Figure 102 Creating Solar System CAM.	248
Figure 103 Inserting a CRT instance into a CAM.....	249
Figure 104 Choosing a CRT to insert.....	249
Figure 105 The CAM with the CRT instances inserted.	249
Figure 106 Copying the Solar System concept.	250
Figure 107 Inserting the Solar System concept.....	250
Figure 108 Completed CAM.....	251

List of Tables

Table 1 Comparison between Models of Adaptive (Educational) Hypermedia	31
Table 2 Sesame2MOT Conversion hypotheses results	78
Table 3 Final evaluation results, web-based Sesame2MOT Enricher v2 application	88
Table 4 Comparison of types of adaptation.....	108
Table 5 GAT hypotheses results internal respondents	169
Table 6 GAT hypotheses results external respondents.....	170
Table 7 Difference between internal and external respondents.....	172

Acknowledgements

First, I would like to thank my PhD supervisor, Dr. Alexandra I. Cristea for her mentorship, guidance, support, and positive criticism throughout. I am also grateful for her organisation of funding for my research through a scholarship, without this work would not have been possible.

Furthermore I wish to thank all collaborators in the PROLEARN and ALS projects and especially the GRAPPLE¹ project. The work in these projects has helped me gain valuable skills in team working and organising, and has been very rewarding.

I also wish to thank the members of the Intelligent and Adaptive Systems Group, and my lab mates, for their interest in and valuable feedback on my research. I would like to specifically mention Joshua Scotton and Fawaz Ghali, who kindly gave up their spare time to proofread drafts of my thesis.

I am also thankful to my friends, in and out of university for their friendship and support.

I express my gratitude to my parents for their continued support.

Special thanks to Sarah for her support, patience and proof reading of this thesis.

¹ <http://www.grapple-project.org/>

Declarations

Much of the material in this thesis derives from papers that have previously been published, as part of my PhD work. These papers are the work of a collaborative effort and expressing my exact contribution is not always easy. In the rest of this declaration I will outline for each sub-chapter if and where the work was previously published and give an indication of how much of the work was done by me.

After the introduction in the first chapter, chapter 2 outlines related research conducted by others, as do individual related research sections within chapters. Chapter 3 is my own work. The ideas were formed during an MSc dissertation, however the finalisation of the ideas as well as both prototypes and evaluation rounds were conducted during the PhD degree. As such the ideas are closely linked to the rest of my work; they cannot be taken apart in an appendix. The work has been previously published in the following papers in which the co-authors have acted as mentors:

- **Finalised representation of the whole cycle from idea to prototypes to evaluations:**
 - [89] Hendrix, M., Cristea, A.I., A Spiral Model for Adding Automatic Adaptive Authoring to Adaptive Hypermedia, Journal of Universal Computer Science (J.UCS), volume 14, 2008.
- **Evaluation results:**
 - [90] Hendrix, M., Cristea, A.I., Joy, M., Evaluating the automatic and manual creation process of adaptive lessons, International Conference on Advanced Learning Technologies (ICALT), 2007.
 - [92] Hendrix, M., Cristea, A., A qualitative and quantitative evaluation of Adaptive authoring of Adaptive Hypermedia, European Conference on Technology Enhanced Learning (ECTEL), 2007.
 - [91] Hendrix, M., Cristea, A.I., Evaluating Adaptive authoring of Adaptive Hypermedia, Adaptive and Adaptable Educational Hypermedia (AEH), Workshop at User Modelling (UM), 2007.

- **Finalised ideas:**
 - [93] Hendrix, M., Cristea, A.I., Nejd, W., Authoring Adaptive Educational Hypermedia on the Semantic Desktop, International Journal of Learning Technology (IJLT), volume 3, 2007.
 - [94] Hendrix, M., Cristea, A.I., Nejd, W., Automatic and Manual Annotation Using Flexible Schemas for Adaptation on the Semantic Desktop, European Conference on Technology Enhanced Learning (ECTEL), 2006
- **Initial ideas:**
 - [95] Hendrix, M., Cristea, A.I., Nejd, W., Authoring Adaptive Learning Material on the Semantic Desktop, Authoring of Adaptive and Adaptable Educational Hypermedia (A3EH), Workshop at Adaptive Hypermedia and Adaptive Web-based Systems, 2006.

Chapter 4 is mainly my own work. However the observation that IMS-LD and AEH have common goals and there may be lessons that the communities can learn from each other was made previously by the collaborators in the following paper:

- [39] Cristea, A.I., Burgos, D., Authoring Adaptive Hypermedia and IMS Learning Design: A Possible Understanding, Conference on Advanced Learning Technologies (ICALT), 2006.

The work was previously published in:

- [85] Hendrix, M., Cristea, A.I., Burgos, D., Comparative Analysis of Adaptation in Adaptive Educational Hypermedia and IMS-Learning Design, Technical Report RR443, 2009.

Chapter 5 is again entirely my own work and ideas. The following papers have been previously published about it:

- [86] Hendrix, M., Cristea, A.I., Reuse Patterns in Adaptation Languages: Creating a meta-level for the LAG adaptation language, Adaptive Hypermedia and Adaptive web-based systems (AH), 2008.

- [87] Hendrix, M., Cristea, A.I., A meta level for LAG Adaptation Language re-use in education, *Adaptable Hypermedia Workshop (A3H), workshop at Adaptive Hypermedia and Adaptive Web-Based Systems (AH)*, 2008..
- [88] Hendrix, M., Cristea, A.I., Meta-Levels of Adaptation in Education, *Computers and Advanced Technology in Education (CATE)*, 2008.

The research presented in chapter 6 has been conducted collaboratively with several partners in the EU GRAPPLE² [125] project. The work as reflected in section 6.4.1 was mainly the responsibility of L. Oneto and M. Dicerto, the work in section 6.4.2 was mainly the responsibility of D. Albert, A. Nussbaumer and C. M. Steiner [5] although I was heavily involved in the ideas, brainstorming, design, support and discussions. The evaluation in section 6.7 was mainly conducted by Work Package 9 partners C. Glahn, C. Steiner, D. Verpoorten, F. Docq, and L. Mazzola [75] in the GRAPPLE project, however I was involved in the discussion as to what should be evaluated and, created manuals for authors, questionnaires, etc. Moreover, I conducted my own two demonstration sessions, where I gathered additional feedback, and extended the analyses of the data carried out by Work Package 9 researchers. As my work is part of a whole, these evaluations reflect directly in it and have to be presented together in this part of the thesis.

All other sections are my work, except for the background research in section 6.8. The research has been previously published in the following papers:

- [82] Hendrix, M., Nussbaumer, A., Dicerto, M., Oneto, L., Cristea, A.I., GAT: The FP7 GRAPPLE Authoring Tool Set, *European Conference on Technology Enhanced Learning (ECTEL)*, 2009.
- [83] Hendrix, M., Cristea, A.I., Stewart, C., Adaptation languages for learning: the CAM meta-model, *International Conference on Advanced Learning Technologies (ICALT)*, 2009.
- [84] Hendrix, M., De Bra, P., Pechenizkiy, M., Smits, D., Cristea, A., Defining adaptation in a generic multi layer model: CAM: The GRAPPLE Conceptual

² <http://www.grapple-project.org/>

Adaptation Model, European Conference on Technology Enhanced Learning (ECTEL), 2008.

Chapter 7 contains conclusions. It is my own work and draws on all other chapters.

This thesis has not been previously submitted in any form for a degree at The University of Warwick or any other institution. The initial idea behind the automatic content addition in chapter 4 was formed during an MSc dissertation. The ideas were finalised, implemented and evaluated during the PhD and therefore it is not possible to single out the research conducted during the MSc dissertation and place them in an appendix.

Abstract

It is well-known that students benefit from personalised attention. However, frequently teachers are unable to provide this, most often due to time constraints. An Adaptive Hypermedia (AH) system can offer a richer learning experience, by giving personalised attention to students. The authoring process, however, is time consuming and cumbersome. Our research explores the two main aspects to authoring of AH: authoring of content and adaptive behaviour. The research proposes possible solutions, to overcome the hurdles towards acceptance of AH in education.

Automation methods can help authors, for example, teachers could create linear lessons and our prototype can add content alternatives for adaptation.

Creating adaptive behaviour is more complex. Rule-based systems, XML-based conditional inclusion, Semantic Web reasoning and reusable, portable scripting in a programming language have been proposed. These methods all require specialised knowledge. Hence authoring of adaptive behaviour is difficult and teachers cannot be expected to create such strategies. We investigate three ways to address this issue.

1. *Reusability*: We investigate limitations regarding adaptation engines, which influence the authoring and reuse of adaptation strategies. We propose a meta-language, as a supplement to the existing LAG adaptation language, showing how it can overcome such limitations.
2. *Standardisation*: There are no widely accepted standards for AH. The IMS-Learning Design (IMS-LD) specification has similar goals to Adaptive Educational Hypermedia (AEH). Investigation shows that IMS-LD is more limited in terms of adaptive behaviour, but the authoring process focuses more on learning sequences and outcomes.
3. *Visualisation*: Another way is to simplify the authoring process of strategies using a visual tool. We define a reference model and a tool, the Conceptual Adaptation Model (CAM) and GRAPPLE Authoring Tool (GAT), which allow specification of an adaptive course in a graphical way. A key feature is the separation between content, strategy and adaptive course, which increases reusability compared to approaches that combine all factors in one model.

List of Abbreviations and Acronyms

AEH	Adaptive Educational Hypermedia
AH	Adaptive Hypermedia
AHA!	Adaptive Hypermedia for All!
AHAM	Adaptive Hypermedia Application Model
AHS	Adaptive Hypermedia Systems
ALE	Adaptive Learning Environment
AM	Adaptation Model
CAF	Common Adaptation Format
CAM	Concept Adaptation Model
CDM	Concept Domain Model
CRT	Concept Relationship Type
CSRT	Concept-Service Relationship Type
DM	Domain Model
DTD	Domain Type Definition
EML	Educational Modelling Language
GAHM	Goldsmiths Adaptive Hypermedia Model
GAL	GRAPPLE Adaptation Language
GAT	GRAPPLE Authoring Tool
GM	Goal and Constraints Model
IEEE	Institute of Electrical and Electronics Engineers
IMS-CP	IMS Content Packaging
IMS-LD	IMS Learning Design
IMS-LIP	IMS Learner Information Packaging
IMS-QTI	IMS Question and Test Interoperability
IMS-VDEX	IMS Vocabulary Definition Exchange
LAG	Layers of Adaptation Granularity
LAG-XLS	LAG XML Learning Styles

LAMS	Learning Activity Management System
LAOS	Layered WWW AH Authoring Model and their corresponding Algebraic Operators
LMS	Learning Management System(s)
LOM	IEEE Learning Object Metadata
MOT	My Online Teacher Adaptive Hypermedia Authoring System
MOT2.0	My Online Teacher 2.0 social web adaptive hypermedia authoring and delivery system
MOT3.0	My Online Teacher 3.0 Adaptive Hypermedia Authoring System
PM	Presentation Model
PRT	Pedagogical Relationship Types
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
RM	Resource Model
SCORM	Sharable Content Object Reference Model
SDM	Service Domain Model
SUS	System Usability Scale
SWM	Service Workflow Model
UM	User/learner Model
UoL	Units of Learning
VR	Virtual Reality
W3C	World Wide Web Consortium
WYSIWYG	What You See Is What You Get
WHURLE	Web-based Hierarchical Universal Reactive Learning Environment
XAHM	XML-based Adaptive Hypermedia Model
XHTML	Extensible Hypertext Mark-up Language
XML	EXtensible Mark-up Language
XSL	EXtensible Style-sheet Language

1 Introduction

Adaptive Hypermedia (AH) [21] is a generalised paradigm, which finds its most common application area as that of Adaptive Web-based systems. The latter includes web pages which can adapt their content towards the users' needs. Personalised Learning is considered an important improvement upon a one-size-fits-all approach by teachers, students, researchers and policy makers alike [38], [115], [143], [155]. Unfortunately, due to the unfavourable ratio between students and teachers, often teachers are unable to provide as much personalised attention to each student as they might like.

A solution to this is Adaptive (Educational) Hypermedia, which can offer a personalised learning experience to each student. This can be used by itself, or as a supplement to regular teaching and learning. Therefore, Adaptive (Educational) Hypermedia can, by delivering a personalised learning experience, offer an improved learning experience compared to classical one-size-fits-all approaches.

Despite its obvious advantages, as briefly summarised above, the acceptance of Adaptive (Educational) Hypermedia³ has not yet been as widespread as one would expect on the basis of its perceived benefits in education. At the same time, on-line courses using so called Learning Management Systems (LMS) are becoming increasingly popular and widespread. A major factor in this is the authoring process of AH. In this thesis we will concentrate on the authoring process and investigate how this can be improved in order to promote wider uptake of AH.

³ In this thesis, as the main application is learning, we use the terms of Adaptive Hypermedia and Adaptive (Educational) Hypermedia interchangeably, but in general, Adaptive Hypermedia can apply to many other domains, such as e-government, e-commerce, etc.

1.1 The Problem

Despite advantages mentioned in the previous section, Adaptive Hypermedia is not as widespread as one would expect. We can identify several reasons for this lack of uptake.

- Creating static web-pages, which allow students to see everything, including concepts which they cannot yet understand, is quite easy, compared to creating AH. However, there is a serious risk for students using such static web-pages, to become 'lost in hyperspace' [63] [38].
- Creating a static web-page, or indeed an LMS course, in which students are only able to see what they should be able to grasp is, however, very difficult [84].
- The lack of standards (compared, for example, to LMSs, which have become very popular for a large part due to the use of e-learning standards, allowing for a very high degree of reusability and compatibility) also hinders the uptake. Due to the lack of standards for adaptation, nearly every system has its own unique way of creating adaptive courses and sharing courses between systems is often impossible or very difficult [39], [85].
- As a result of the lack of standards many systems have their own unique formats and there is very little content readily available for teachers to reuse [39], [85].

Hence, whilst AH-based learning may be beneficial, the authoring process of Adaptive Hypermedia has always been an important bottleneck.

1.2 Objectives

The aim of this research is to improve the authoring process of Adaptive Hypermedia and to facilitate Adaptive Hypermedia in reaching its potential in improving learning outcomes of students taking online courses [38], [115], [143], [155]. Therefore in this thesis we will investigate ways to improve the current state of the authoring process of AH, in such a way that ultimately teachers without a computer science background will be able to create adaptive courses [84]. In addition to this, an important aspect of the current issue is the fact that nearly every AH system employs its own tools and formats for authoring. Hence in

order to move an adaptive course authored from one system to another, a teacher will effectively have to recreate it [39], [85]. Therefore this thesis will compare current standards for learning specifications with Adaptive Hypermedia to see how AH and current learning based standards can benefit from each other.

1.2.1 Research Questions

As we have seen in previous sections, Adaptive Hypermedia can provide a personalised learning experience and thereby provide an improvement on one-size-fits-all approaches. We have seen that the Adaptive Hypermedia has not received as wide an uptake as we would expect and have identified the authoring process as an important bottleneck.

Hence our main research question: *How can we improve the authoring process of Adaptive Hypermedia in such a way that it allows technically experienced adaptive course authors to successfully specify adaptive courses?* The eventual aspiration of this research is supporting a teacher with non technical background, but this thesis focuses on providing support for technically experienced adaptive course authors.

Given the two major components of authoring (the content authoring and the adaptive strategy authoring), the question can be broken down into the following questions:

- *Can tools be developed which harvest multiple alternative content (i.e. supplementing the available content for the course under construction, with content already stored or accessible on the author's desktop) which is both highly relevant for the course and is aligned with the teaching strategy for the course under construction.*
- *Can IMS-LD, which is emerging as de-facto e-learning standard, improve the interoperability of arbitrary Adaptive Hypermedia systems and in particular, can AH be expressed in IMS-LD and vice-versa?*
- *Can a meta-language be designed and can a tool be built, which increases the reusability of adaptation strategies?*

- *Can tools be developed that allow the specification of an adaptive course including the adaptation strategies? Can technically experienced authors successfully create courses with these tools and can they successfully create adaptation strategies?*

Our hypotheses regarding these research questions are the following:

- *Tools can be developed which harvest multiple alternative content. Course authors feel the content both highly relevant for the course and is aligned with the teaching strategy for the course.*
- *IMS-LD can be used to express AH, and therefore provide a level of interoperability between arbitrary Adaptive Hypermedia systems. In general AH can also express the IMS-LD standard.*
- *A meta-language can be designed and a tool can be built, which increases the reusability of adaptation strategies.*
- *Technically competent course authors feel they can create adaptive courses, including adaptation strategies, using an authoring tool based upon a novel graphical authoring model.*

1.3 Thesis Outline

The rest of this thesis is organised as follows. Chapter 2 features a review of the state of the art in Adaptive Hypermedia. The field of e-learning has goals which are highly related to the field of Adaptive Educational Hypermedia. In contrast to Adaptive Educational Hypermedia, however there exists an abundance of standards for e-learning. Therefore chapter 2 also reviews the state of the art in e-learning standards. Chapter 3 describes our approach to address the difficulties authors have in creating content for Adaptive Hypermedia. The approach can relieve some of the content creation burden, specifically the creation of content alternatives for adaptivity, based on the assumption that teachers will have a lot of highly related material on their own desktop. Standards are largely missing at present from Adaptive Hypermedia. Chapter 4 presents a comparative overview of the IMS-Learning Design specification and Adaptive Hypermedia. Improving re-usability of adaptive strategies, as described in chapter 5, could help authors create adaptive courses

more effectively. Chapter 6 shows a novel graphical approach to authoring of Adaptive Hypermedia. An authoring model and tool are described, which allow authors to more easily specify an adaptive course in a visual way without the need for programming knowledge. Finally in chapter 7 we discuss our results and recommend directions for further work.

2 Background Research

As introduced in the first chapter, our aim is to improve the authoring process of AH, especially for teachers without a computer science background. This improves the acceptance of AH, which can potentially lead to improved learning outcomes for students taking courses in AH systems. In this chapter we will introduce important related research and indicate how it relates to our own research.

The rest of this chapter is organised as follows. In section 2.1 we introduce e-learning systems and standards which have similar goals to Adaptive Educational Hypermedia. Some of these standards have some features for adaptive e-learning and, while these standards may not be able to express everything needed for authoring an AH course, they have some commonality with AH objectives and, more importantly, are in widespread use. Section 2.2 introduces the key concepts in the Semantic Web. The Semantic Web paradigms are closely related to the research on the Semantic Desktop, which features in our approach for automatic content addition, in chapter 3. The system implemented in the GRAPPLE project⁴, of which the new graphical authoring model CAM, described in chapter 6, is a part, uses Semantic Web technology to keep track of information about the user in its user modelling framework. Sections 2.3 and 2.4 introduce the state of the art in AH. Section 2.3 discusses Brusilovsky's taxonomy [21], which is the most widely known comprehensive survey of AH technologies. Section 2.4 gives an overview of the state of the art in models for Adaptive (Educational) Hypermedia, describing both the authoring of content as well as adaptation. Section 2.5 expands upon section 2.4 by introducing languages for adaptation in AH. Finally in section 2.6, a selection of well known adaptive authoring systems is introduced and we show how in these systems both content and adaptation are created.

⁴ <http://www.grapple-project.org/>

2.1 E-Learning and standards

2.1.1 Metadata standards IEEE Learning Object Metadata: LOM

In order to create adaptive educational material, it is logical to start with existing standards for creating educational and e-learning material, in order to see to what extent they can support the definition and creation of personalised learning. A well-known standard for learning object definition is the IEEE Learning Object Metadata (LOM) [98] specification.

An adaptive system uses resources such as text, audio and video. Therefore, a functioning system will need to capture the resources it uses either in a separate resource model or, as is often the case, integrated into the Domain Model, which captures the domain conceptually. To be able to make decisions about adaptation, each resource must have metadata specifying attributes such as difficulty, language and media format. LOM defines a number of vocabularies for describing pedagogical attributes of resources. It includes, for example, attributes like resource type, level of interactivity, semantic density, difficulty and description. In addition, LOM encapsulates the Dublin Core [61] elements, see also section 2.1.2. Importantly, LOM describes learning objects. It does not itself have any adaptive functionality. LOM describes whole learning objects by themselves, but does not make statements about the smaller building blocks individually. LOM can be used to select learning objects in an adaptive way, but adapting the learning objects themselves is not supported.

2.1.2 Dublin Core

As we have seen in the previous section, LOM encapsulates the Dublin Core elements. The Dublin Core metadata element set is not a dedicated e-learning standard, but a cross-disciplinary standard for describing a wide range of networked resources [97]. Dublin Core is mentioned here because of its use in the Semantic Web and in the LOM standard. There are two levels: *Simple* and *Qualified*. Simple Dublin Core describes objects using 15 simple elements [61] like the Title, Creator and Subject. The Qualified Dublin Core adds three additional elements to that: *Audience*, *Provenance* and *RightsHolder*. It also adds a group

of *element refinements* or *qualifiers*. Qualifiers further refine the elements, mainly in ways that are seen as useful for resource discovery. The elements describing objects could be used in AH to discover suitable learning objects. For example, the title and keywords could be used to relate learning objects in a repository to a course in the AH system and either shown to students or offered to the teacher for inclusion. The qualifiers could be especially useful in this respect.

2.1.3 IMS Learner Information Packaging (IMS-LIP)

IMS Learner Information Packaging⁵ contains a collection of information about a *Learner* (or group of Learners) or a *Producer* of learning content (creator, provider or vendor). The aim of the specification is to provide interoperability for exchanging learner information on the Internet. Learner data can be imported into and extracted from IMS-LIP compliant learner information servers. Learner information systems may exchange data with this standard, or even integrate it into delivery systems. The owner of an information server should be able to specify what information may be shared with other systems. The following entities form the core structures of IMS-LIP: *accessibilities*, *activities*, *affiliations*, *competencies*, *goals*, *interests*, *qualifications*, *certifications* and *licences*, *relationship*, *security keys* and *transcripts*. The IMS-LIP specification is for exchanging information about learners. Information about learners is highly relevant in Adaptive Hypermedia, often in the form of a user- model. However, an important feature of user-models in AH, missing in IMS-LIP, is the overlay model. For each Domain Model topic, the overlay model keeps information about the user, such as his knowledge of the topic. IMS-LIP is also used for example to pass information between the delivery engine and the User Modelling framework [1] in the GRAPPLE⁶ project.

2.1.4 IMS Content Packaging (IMS-CP)

IMS Content Packaging [99] is an IMS standard for packaging content and metadata together. It can, for example, package some content or an assessment that can be delivered,

⁵ <http://www.imsglobal.org/>

⁶ <http://www.grapple-project.org/>

for instance, by a Learning Management System. IMS-CP consists of a ZIP file, containing a manifest file *imsmanifest.xml*. The learning content is usually included in the ZIP, hence the package forms a self-contained unit. However it is technically possible to refer to URLs on the Internet, which effectively breaks with the self-containment. IMS-CP is used by the Sharable Content Object Reference Model (SCORM) [2], [103] (see section 2.1.5). Typically, each SCORM Content Object (CO) is defined by a content package. While this standard is useful for interoperability and being able to easily transport content in one package, it does not provide any features for capturing adaptation.

2.1.5 Sharable Content Object Reference Model (SCORM)

The SCORM [2], [103] standard was created in order to bring together existing e-learning standards into one single package. It is intended to form a structured design pattern around the creation of interchangeable educational material. SCORM consists of the following parts.

- *The Content Aggregation Model* defines the components used to build a learning experience from learning resources, and how they are aggregated and organised in units of instruction.
- *Assets* represent a collection of media. If this collection represents a single learning resource, it is called a *Sharable Content Object* (SCO).
- An *activity* is a unit of learning, which may provide an asset or be composed of sub-activities.

A course is created by composing activities in a *Content Organisation*. The *Content Aggregation* is used to deliver both the structure and resources belonging to a *Content Package*. The content is described in a content package, which contains a file called *imsmanifest.xml* describing the structure of the content package and physical resources. The runtime communications consist of commands for communicating information to and from the SCORM compliant system (for example a Learning Management System). The commands use student metadata for storing information on individual students. The course

metadata in SCORM is based upon LOM [98] (see section 2.1.1). It contains a dictionary of metadata terms that can be used to describe a course for the purpose of archiving and sharing the course. The *imsmanifest.xml* file also contains information on what is called ‘*content sequencing*’. This specification allows a limited amount of adaptive behaviour to be specified. However, compared to AH, the adaptation that can be specified is very limited [131], despite the fact that SCORM was designed with adaptation in mind. SCORM allows several content organisations that could be intended, for example, for different learning styles, backgrounds and devices. However, the choice of which content alternative to display is left up to the delivery environment and most either show the default content alternative or leave the selection to the learner. Sequencing information allows the definition of a set of rules for deciding the next activity to be shown. However this information is only based on parameters corresponding to the particular course, and there is no model of persistence between the courses or a way to use external information.

2.1.6 IMS Vocabulary Definition Exchange (IMS-VDEX)

The IMS Vocabulary Definition Exchange [102] specification was designed for the exchange and expression of simple machine-readable lists of human language terms, along with information that may assist a human in understanding the meaning of the various terms [65]. IMS-VDEX can be used to encode different types of vocabularies. It can be used to encode a simple flat list of terms, a glossary or dictionary and a thesaurus or a hierarchy of terms. Although IMS-VDEX is not meant to accommodate all possible complex vocabularies, there is a limited provision for complex vocabularies. IMS-VDEX was specifically designed to supplement other IMS specifications and the IEEE LOM standard by giving additional semantic control to tool developers. Figure 1 describes the IMS-VDEX format schematically. The IMS-VDEX format depicts a vocabulary of terms, but does not itself have any adaptive features. It could however be used to specify the content in an adaptive system. As we can see in the figure, the *Vocabulary* contains an *Identifier*, *Name*, *Default Language*, *Profile Type*, *Registration Status* and *Order Significance*, indicating whether the order of terms is important. It also contains a number of *Terms*, *Relationships* and freeform *Metadata*. The *Terms* contain an *Identifier*, *Caption*

(the title), *Description*, *Valid for Index*, a Media Descriptor, possibly a number of sub-terms and freeform *Metadata*. The Media Descriptor locates the related media file and adds a *Note* for information. The relationships have a *Type* name, *Source* term and *Target* term.

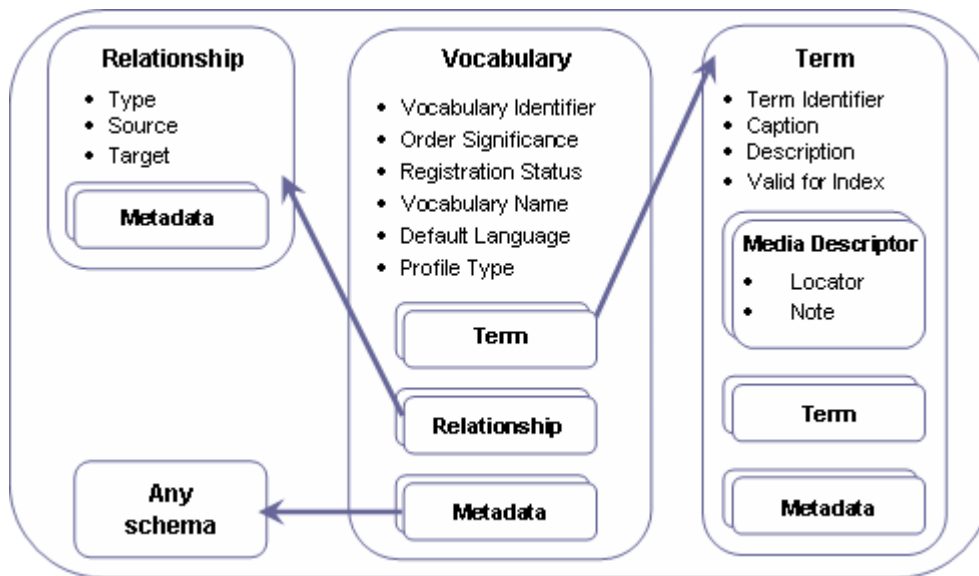


Figure 1 Schematic description of the IMS-VDEX format [65].

The IMS-VDEX standard is intended for describing vocabularies. It can be used to describe Domain Models, as a conceptual domain can be seen as a special vocabulary. Thus, in the new graphical authoring model CAM, described in chapter 6, the IMS-VDEX standard was chosen to represent the Domain Model.

2.1.7 IMS Question and Test Interoperability (IMS-QTI)

The IMS Question and Test Interoperability [101] specification is a standard designed for interoperability between systems. It consists of a set of questions, assessments and results, bundled together by using an XML data binding. The interchange of questions is a widely used feature of IMS-QTI. The standard provides an essential part of learning systems, the assessment. Similarly to IMS-CP however, IMS-QTI does not provide features for adaptation. However, it is still very useful for AH systems, as, to be able to support IMS-QTI assessments, is clearly relevant to its goals.

2.1.8 IMS-Learning Design (IMS-LD)

The IMS Learning Design [100] specification is an IMS specification for learning, with strong pedagogical roots. It models pedagogical scenarios and is, to an extent, emerging as a de-facto standard. It was originally developed by the Open University of the Netherlands in 2003. It aims at covering various learning situations and roles. Key to the IMS-LD specification, are the different *roles* people can have. Each role has specific *objectives* that can be reached by a person by performing (learning or support) *activities*. These activities are conducted in an environment consisting of *learning objects* and *services*. On the technical side, the specification is made up of three levels, which model what is called the Units of Learning (UoL).

- *Level A* provides method, activities and roles.
- *Level B* adds properties, conditions, calculations, global elements and a monitoring service.
- *Level C* rounds it off with notifications [105], [142].

The goal of IMS-LD is very similar to that of Adaptive Educational Hypermedia: to improve students' learning outcomes by giving them a tailored learning experience. In chapter 4, the similarities and differences, especially concerning adaptation, are investigated in more detail.

2.1.9 Learning Management Systems (LMSs)

Learning management systems are systems that can combine a large variety of information on courses, both in terms of content, as well as in course delivery tools (such as time scheduling tools and chat tools). Learning management systems offer different interfaces for different roles. Teachers and students get different functionality, messaging functions, notification tools and so forth. Popular open source learning management systems are for example Sakai⁷ and Moodle⁸. Some examples of commercial learning management systems are CLIX⁹ of IMC and LearnExact/Hive¹⁰ of GIUNTI labs. The advantage of learning

⁷ <http://sakaiproject.org/>

⁸ <http://moodle.org/>

⁹ <http://www.im-c.de/en/products/learning-management-system/product-overview/overview/>

management systems is that they support many e-learning standards and that they are widely in use in academia as well as schools around the world. The disadvantage however is that they do not support as yet personalisation to the learner (or author) and customisation of a very limited range of parameters, usually based on adaptability (user-tuning) and not adaptivity (automatic system tuning).

2.2 Semantic Web/Desktop

'*Semantic Web*' [14] is a term first coined by Sir Tim Berners-Lee, the inventor of the World Wide Web, as the successor of the World Wide Web. In the Semantic Web, data and their representation are separated from each other. The semantics of the data is stored in a machine readable format. The idea behind this is that it will be easier for machines to 'understand' the semantics and automatically process the information, which would lead to a universal medium of knowledge exchange.

The *Semantic Desktop* [132] is a vision of a user's computer, where all information is semantically stored. This makes it easier to retrieve and reason about information on one's computer. This vision is brought to life via a set of languages and tools. Semantic Desktop and Semantic Web are based on a set of shared concepts and standards. The rest of this section will introduce these key concepts and standards that drive the Semantic Web and Semantic Desktop.

2.2.1 Resource Description Framework and Resource Description Framework (RDF and RDFS)

The Resource Description Framework (RDF) [129] is a formal specification, which is a W3C standard for representing metadata on the web. It is intended to provide a formal description of concepts, terms and relationships in a given knowledge domain. The format is also heavily used on the Semantic Desktop. RDF enables the encoding of structured metadata in a machine-processable way. This enables the exchange of information and it

¹⁰ <http://www.giuntilabs.com/>

also means that the information can be made available to other applications it was not originally intended for.

RDF is built out of *subject-predicate-object triples* forming a graph that can be represented visually. There are a number of different serialisations of RDF. RDF/XML is an XML serialisation of the triplets, Notation 3 (N3) and Turtle (a subset of N3) are non-XML serialisations of the RDF graphs.

RDF allows any object from one triple to be the subject of another and therefore supports chaining of relationships as well as supporting *reification*. For example [130], consider a person *person:p1*, who works at the *company:c1*. This can be described using the following RDF:

```
person:p1    company:worksAt    company:c1
```

Using the reification vocabulary we can now describe being an employee of company c1 as follows:

```
company:employeeE1    rdf:type          rdf:Statement
company:employeeE1    rdf:subject       person:p1
company:employeeE1    rdf:predicate     company:worksAt
company:employeeE1    rdf:object        company:c1
```

Using these statements about employee E1, we can express who hired person p1:

```
person:p2    company:hired    company:employeeE1
```

This means that any RDF statement can itself be the subject or object of a triple. Hence graphs can be nested, as well as chained. RDF uses Uniform Resource Identifiers (URIs) to describe resources. Below, an example RDF fragment is given [130]. In this example, the popular – and now accepted standard - RDF/XML notation is used to describe the email, name and title of a person called Dr. Miller.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">
  <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Dr. Miller</contact:fullName>
    <contact:mailbox rdf:resource="mailto:em@w3.org"/>
    <contact:personalTitle>Dr.</contact:personalTitle>
  </contact:Person>
</rdf:RDF>
```

RDF does not pre-define any terms for data modelling. RDF Schema (RDFS) however does predefine some terms such as *Class*, *subClassOf* and *Property* and allows developers to define their own simple ontologies for their RDF data. These can be used for simple reasoning, such as whether a resource belongs to a certain class.

2.2.2 Web Ontology Language (OWL)

The Web Ontology Language [112] is a W3C standard designed for applications with semantic information in the form of ontologies. It is defined as an extension to RDF. It adds formal semantics to RDF by using pre-defined constructs that go beyond the RDF Schema. The OWL has an RDF/XML syntax. Formal agreements on semantics enable reasoning over the data. OWL is sub-divided into three different versions, increasing by expression power:

- *OWL Lite* supports a simple set of constraints, computable in polynomial time.
- *OWL DL (Description Logic)* has more expressive power. Computability of results is guaranteed, but it does not guarantee that results can be computed in polynomial time.
- *OWL Full* is the full OWL language, it relaxes the constraints of OWL DL and allows rules that violate Description Logic restrictions. It allows free mixing of OWL and RDF Schema and does not enforce a strict separation of classes, properties, individuals and data values. As a result computability is not guaranteed.

Similarly to RDFS, reasoning [12] can be done in OWL, in order to infer information over semantic data. OWL reasoning can go beyond the reasoning capabilities of RDFS, by applying instance inferences and defining rules. Reasoning in OWL is based on semantics. Using the semantics, inferences about ontologies or ontology-members can be made. Reasoning can be done in the following ways:

- *Class inferences*

Inferences based on classes are based on *subclasses* or on *sub-properties*. As an example [130], let us take four classes: a driver, a bus driver, a vehicle and a bus. Bus is a subclass of vehicle and a bus driver drives a bus, a driver drives a vehicle, so we can conclude that a

bus driver is a driver. These kinds of inferences are also possible in RDFS schema inferences and can be used in our prototypes, described in chapter 3 in more detail.

- *Instance inferences*

Inferences can also be based on instances. When a property takes a certain subtype and there is an instance which has this property, the instance must be of that subtype. For example, we have two classes: a bus driver and a bus. The *bus driver drives* an instance called *coach*. The *drives* property of *bus driver* takes busses, so the *coach* must be a *bus*.

- *Rule Set*

In OWL reasoning can also be done by a set of rules to the OWL ontology. Here OWL really goes beyond the reasoning capabilities of RDFS.

The prototypes we developed for automatic content addition, as described in chapter 3, optionally make use of RDFS for inference. The reason for choosing RDFS over OWL was that the Semantic Desktop setup at the time used a Sesame¹¹ [18] store and used RDFS. This was mainly for performance reasons. Moreover, many OWL light features can be expressed with RDFS.

2.3 Adaptive Hypermedia: Brusilovsky's Taxonomy

Brusilovsky's taxonomy [21] is widely seen as the comprehensive survey of Adaptive Hypermedia technologies. The taxonomy outlines the main areas of adaptation in AH. Thus, its understanding is crucial in understanding the makings of an adaptive hypermedia system.

In Figure 2, the taxonomy is schematically given. In chapter 4 we use the taxonomy to compare AH and IMS-Learning Design. The taxonomy outlines what can be adapted. It does not specify how this should be done. The main components are *adaptive presentation* and *adaptive navigation support*. Adaptive presentation branches out into *adaptive*

¹¹ Sesame is an open source RDF framework, see <http://www.openrdf.org/>

multimedia presentation, *adaptation of modality* (which media is used) and *adaptive text presentation*. Adaptive text presentation is the most widely used. Systems like MOT [37] and AHA! [54] include other media, but it is the selection of media files which is adapted, by producing different hypertext, not the media files themselves. The Adaptive text presentation is split up in *Natural Language generation* [157], which involves natural language processing techniques, and the more popular *Canned text adaptation*. The canned text adaptation is further split, based on the various processing ways that are possible with text that is divided into fine-grained fragments. The reason why this type of adaptation is most detailed in the taxonomy is that is it the most widely used.

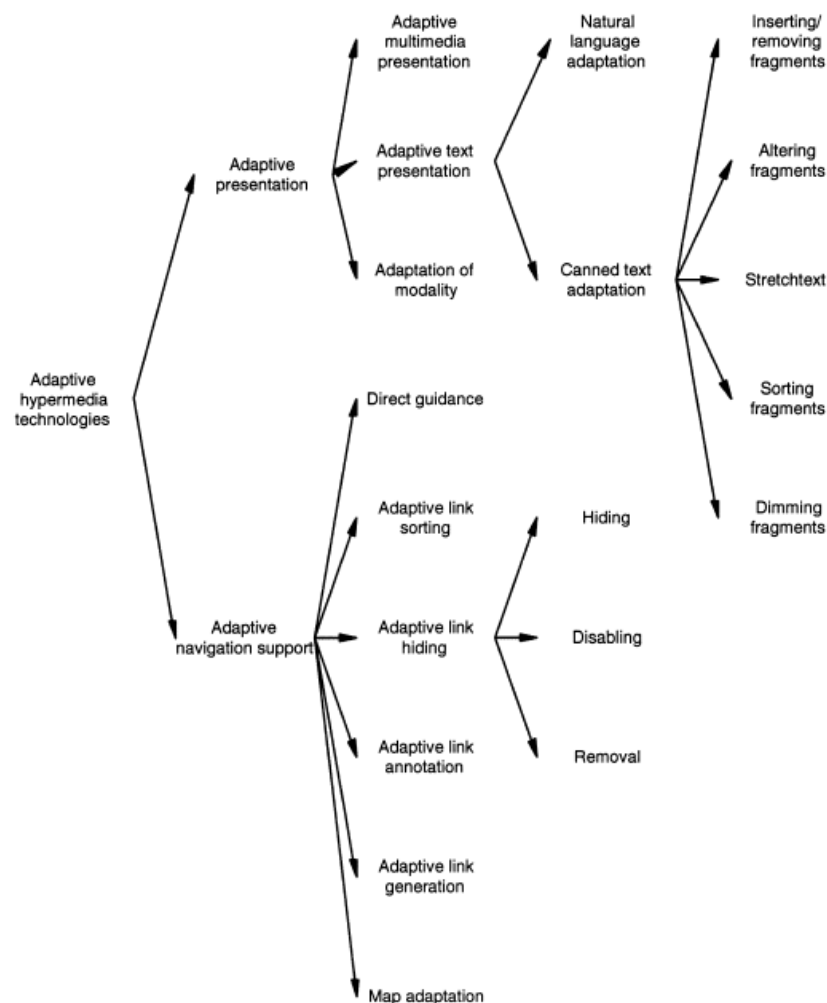


Figure 2 Brusilovsky's taxonomy of Adaptive Hypermedia technologies [21]

Along with adapting the actual content, the navigation can also be adapted using adaptive navigation support. Adaptive navigation support has clear benefits over websites which

allow users to navigate everywhere. The navigation support provides users with fewer but more relevant options to choose from, thus eliminating or reducing the ‘lost in hyperspace’ syndrome [63]. This can be in the form of *direct guidance*, removing all choice, *sorting of links* in order of relevance, *hiding* or *disabling links* considered not relevant, *annotating links* and *maps* as well as *generating new links*.

2.4 Models for Adaptive (Educational) Hypermedia

Creating Adaptive Hypermedia is a complex endeavour, especially compared to creating static web-pages. Authoring of Adaptive Hypermedia is a problem that has received a lot of attention recently, as the usability of the authoring process directly impacts upon the wider acceptance of AH.

Many adaptive (educational) hypermedia systems have been launched since the early 1990s. The Adaptive Hypermedia Application Model (AHAM) [56], [153] was one of the first models, followed by the Web Modelling Language (WebML) [30], the Goldsmiths Adaptive Hypermedia Model (GAHM) [122], the Munich reference model [104], the XML Adaptive Hypermedia Model (XAHM) [29], the LAOS framework [41] and the Generic Adaptivity Model (GAM) [58]. The goal of each of these models is to record important concepts in current Adaptive (Educational) Hypermedia systems, such as the node/link structure, user model, adaptation patterns and presentation settings. In this section we investigate various frameworks for personalisation on the web. We consider different aspects of adaptation and personalisation to inform the design of our new graphical authoring model, described in chapter 6, called the Concept Adaptation Model (CAM).

2.4.1 The Dexter Hypertext Reference Model

The Dexter Hypertext Reference Model [79] is a model that aims to capture the important abstractions found in a wide range of hypertext systems. The model was introduced in 1990 and is aimed at hypertext systems in general and not at Adaptive Hypermedia in particular. However the comprehensiveness of the model has made it the basis for the AHAM [59],

[156], Munich [105] and LAOS [43] models described later in sections 2.4.2, 2.4.3 and 2.4.4. The model consists of three main layers, the *storage* layer, the *runtime* layer and the *within-component* layer. The *storage layer* describes the nodes and links as a network that forms the hypertext. The *runtime layer* describes mechanisms for user-interaction with the hypertext, and finally the *within-component* layer describes the content and structures within the nodes. The model is formally specified in the Z specification language [141].

2.4.2 AHAM

The AHAM [56], [153] reference model for Adaptive Hypermedia Systems (AHS) describes adaptive applications as consisting of three main layers:

- The *Domain Model* (DM) describes concepts, then groups them in a hierarchical structure and defines arbitrary concept relationships, possibly of a special domain *concept relationship type* (domain CRT). In principle, a DM can be “imported” from a simple subject domain ontology, except for the concept relationships that have a (pedagogical) meaning.
- The *User Model* (UM) also defines concepts, but with user specific attributes, for example, knowledge level, learning style and preferences. Typically, the UM is an overlay model of the DM, meaning that for every concept in a DM there is a corresponding concept in the UM.
- The *Adaptation Model* (AM) defines the adaptation behaviour. It typically consists of *condition-action rules* or *event-condition-action rules* that define how user actions are transformed into UM updates and into the generation of presentation specifications. There are two types of rules.
 - *Generic adaptation rules* are connected to CRTs; this for instance allows the definition of a knowledge update rule for visiting pages and a prerequisite rule for determining the suitability of concepts, depending on whether all prerequisites are satisfied. An author only has to specify concept relationships and an authoring tool can then generate the corresponding adaptation rules automatically.

- *Specific adaptation rules* apply to specific concepts of the Domain Model and can be used for a very rare adaptation rule or for defining an exception to a generic adaptation rule. Authoring such specific adaptation rules requires knowledge of the language in which such rules are defined (and which is system-dependent).

In the AHA! System [54], a graphical authoring tool, the Graph Author see section 2.7.2, is used to define the DM and draw a graph of concept relationships (of different types). As the name “graph” suggests, concept relationships are binary (or in some cases unary), whilst in AHAM there is no restriction to the number of concepts that together may form a relationship.

2.4.3 Munich Adaptive Hypermedia Reference Model

The Munich Adaptive Hypermedia Reference Model [104] is based on the Dexter hypertext reference model. The Munich model is similar to AHAM, it features an object-oriented specification, written in UML ¹² and offers a visual representation and formal specification in the Object Constraint Language (OCL) [121]. The model extends each layer of the Dexter hypertext reference model to include adaptation and user modelling information as follows.

- The *run-time layer* is responsible for the user interaction. It is extended with acquisition of user behaviour and management of the sessions needed in AH systems.
- The *Storage layer*, instead of just storing the nodes and links of a hypermedia, as in the Dexter model, is divided into three sub models.
 - The *Domain Meta-Model* manages the basic structure and content of the hypermedia system in a set of nodes and links and the mechanisms with which they are navigated. The nodes are treated as data containers.
 - The *User Meta-Model* manages sets of attributes about users, with the intention to personalise the hypermedia based upon these attributes.

¹² Unified Modeling Language, www.uml.org/

- The *Adaptation Meta-Model* consists of a set of adaptation rules that perform the personalisation based upon the information in the User Meta-Model.
- The content and structure defined in the *within-component layer* depends on the application and is not detailed by the Munich AH reference model.

2.4.4 Web Modelling Language (WebML)

WebML [30] is a visual language based on UML, but designed for describing the content structure of web applications. The specification of a website in WebML has four perspectives.

- The *structural model* describes content in terms of the relevant entities and relationships.
- The *hypertext model* describes the navigation through the content [30]. It consists of the Composition and Navigation models, which describe the composition and navigation through the content.
- The *presentation model* uses an abstract XML syntax to describe the appearance and layout of pages, separated from the device and from the rendition language.
- The *personalisation model* has the entities *user* and *group* to describe the users and the groups they form. Personalisation is defined based on the data stored in these entities.

2.4.5 The Goldsmiths Adaptive Hypermedia Model (GAHM)

The Goldsmiths Adaptive Hypermedia Model [122] is a conceptual model. It divides functionality into three areas as follows.

- The *H-Region* models hypermedia-based interaction that cannot be personalised. Hyper-pages are represented as formal specifications, of which the semantics are given with reference to an abstract machine. Hence it produces rendered hyper-pages.
- The *P-Region* models user-initiated personalisation of hypermedia content. Hence this adaptation is user-driven, or adaptable.

- The *A-Region* models system-initiated personalisation of hypermedia content. The adaptivity is modelled as a process, allowing the system to initiate personalisation. Hence the adaptation is system-driven, or adaptive.

The model features a detailed low-level description of adaptation specification, but bestows less attention on the purpose of the adaptation (for example, the pedagogical approach, in the case of educational systems).

2.4.6 The XML Adaptive Hypermedia Model (XAHM)

The XML Adaptive Hypermedia Model [29] does not build on previous models, but starts off fresh. As the basis it uses graph theory and a probabilistic interpretation of the weights in the application domain. XAHM is aimed at the delivery process and supports system variables needed in adaptation and captures implicit and explicit user behaviour.

XAHM is XML-based and contains an *application domain*, a *user model* and an adaptation model. It consists of a *graph-based layered model* for describing the logical structure of the hypermedia and *XML-based models* for describing the metadata for basic information fragments and pages to be adapted. Adaptation is mapped over three dimensions: the *user's behaviour*, the *technology* and the *external environment*. The data model comprises *information fragments* (comparable to data in the within-component layer in AHAM and Munich), *presentation descriptions* (XML formatting information), *elementary abstract concepts* (larger units comprising a number of adaptively linked presentation descriptions) and an *application domain* (a set of linked and weighted elementary abstract concepts). The adaptation is mainly around the user's behaviour.

2.4.7 The Generic Adaptivity Model (GAM)

The Generic Adaptivity Model (GAM) [58] is a model that is based on *state-machines*. Each interaction generates an event that triggers a state change in the system. The events can be parameterised by external values, for example, stored in a user model. GAM is more low-level compared to AHAM, as it does not provide hypermedia-specific concepts. The differences between AHAM and GAM are as follows.

- AHAM is specific for AH systems, whereas GAM aims to provide a generic adaptivity model.
- In GAM, *push* and *pull adaptation* have been added, whereas AHAM only performs push adaptation. In push adaptation, the incoming event generates a set of follow-up events that cause the model to update. Pull adaptation, on the other hand, means that when the value of a certain property is requested, the adaptive engine needs to use other properties in order to calculate that value.

2.4.8 Layered WWW AHS Authoring Models and their Corresponding Operators (LAOS)

The LAOS framework [41] is a five-layered extension of AHAM. It responds to and improves upon some of the limitations of AHAM, with a special focus on authoring. The major difference between LAOS and AHAM is a higher level of reuse, due to the clear separation of *primitive information* (content) and *presentation-goal* related information, such as pedagogical information in educational systems and prerequisites. In the following, more differences are analysed. In LAOS, different aspects of the adaptation model are distributed over multiple layers [84] in the model, in particular the following:

- The *Domain Model*, is similar to the AHAM model DM, with the exception of allowing only domain-specific information (concepts and links), unlike in AHAM. Hence no information is hard-wired in the definition of the concepts, but it is kept separate. Therefore the concepts and indeed the whole DM can be reused with different strategies;
- The *Goal and constraints model* (GM) extracts and concentrates all goal-related (for example, pedagogical, for educational applications) information previously intermingled with domain information in the AHAM model. This improved separation of concerns greatly increases the reuse potential of the DM.
- The *Presentation model* (PM) extracts and concentrates all presentation information previously intermingled with domain information in AHAM. This additional separation of concerns again promotes reuse of the DM, as well as of the PM itself.

- The *User model* is similar to the one in AHAM.
- The *Adaptation model* extends the ideas of generality and specificity.
 - *Generic adaptation rules*: instead of connected to a type of event (such as visit), allowing for specification of the combination of *type of event*, *type of concept*, *type of relation* and determining, based on this ternary combination, the UM or PM updates. Such rules can be applied to *any* domain map.
 - *Specific adaptation rules*: similar to the generic rules, but applied to a specific concept, identified from a given domain map (this is mainly used for backward compatibility with previous types of adaptive hypermedia models, like AHAM, where concepts were connected only in this explicit, specific way and only reusable with one DM).

In this way, pedagogical information, for example, can be expressed in the GM alone and kept apart from other information. Also, the PM describes the final look and feel of the presentation as well as the quality of service parameters (for example for mobile devices).

Concept relationship types in LAOS are also of different types, depending on the model they belong to. For example,

- the domain CRTs only describe domain relations (such as *part-of*, *is-a* or *relatedness* relations as available in the MOT authoring tool [40], [42] built on LAOS); similarly,
- CRTs in the GM describe only goal-related relations (pedagogic relations for the educational domain, such as AND-OR relations with pedagogical labels, as available in the MOT authoring tool).

Moreover, the adaptation model allows for different levels (and thus degrees) of reuse of adaptation, by conforming to the LAG framework. This type of layered structure allows reuse of each layer separately, beside the type of reuse described above, and thus is more

flexible and more appropriate for authoring, where the ‘write-once use-many’ paradigm is most relevant¹³.

LAOS does not prescribe a specific representation for each layer, unlike models such as XAHM or GAM, but merely specifies its contents. Hence, each layer can be represented for example by databases, XML files or state machines. The theoretical adaptation framework, LAG, only specifies different levels for reuse (strategy, high level patterns, low level adaptation ‘*assembly*’ language patterns for example if-then rules). It does not enforce a specific language¹⁴.

2.4.9 The Layers of Adaptation Granularity (LAG) Framework

The *Layers of Adaptation Granularity framework* [43] divides the Adaptation Model of LAOS into three parts:

- direct adaptation techniques, such as (event-)-condition-action rules, ((E)CA) as in AHAM, that are the building blocks for adaptation and also thus forming the assembly-language type of adaptation specifications;
- more sophisticated adaptation languages [37] (such as LAG [43] or LAG-XLS [136]);
- adaptation strategies, comprising reusable, annotated storylines of adaptation and personalisation that can be applied to various Domain Models.

The direct adaptation techniques mainly consist of Event Condition Actions (ECA). ECAs are active rules, consisting of three parts.

- the event, which specifies, what triggers the invocation of the rule;
- the condition, which has to be satisfied before the action can be performed;
- the action, the updates of local data, usually the user model. This does not always result in directly noticeably adaptation of the AH.

¹³ Note that previous models were not aimed especially at authoring and thus reuse was not considered as a priority. From an adaptation engine perspective, the speed of delivery is more relevant than the reuse, thus a compact, quick-response storage mechanism is preferable to a distributed, highly-layered one.

¹⁴ The LAG language, see section 2.5.1, is not to be confused with the LAG framework. The LAG language is a language that fits in with the LAG framework, but others can be used.

Many AH systems use Condition Action (CA) rules. CA rules are similar to ECAs, where the condition acts as event. Hence the rules are activated when the condition is satisfied, and deactivated when it no longer satisfied.

The framework standardises adaptation techniques at the different levels and aims to enable exchange of adaptive techniques between different applications and to give authors higher-level handles of low-level adaptation techniques. Hence authors can define adaptation at the level of adaptation strategies and leave the details to the system. These strategies, contrary to adaptation descriptions at the lowest level, can than be reused in other systems or with other content. Figure 3 shows the three layers schematically.

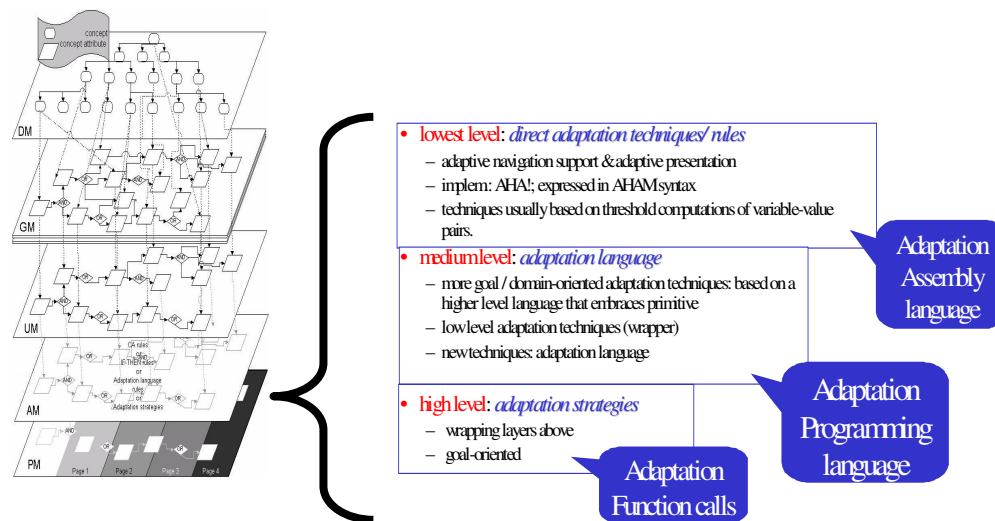


Figure 3 The five level AHS authoring model and three layers of adaptation

2.4.10 Comparison

In [44] the authors compare various models for authoring AH. In this section we expand upon this comparison, summarising and comparing the models described in the previous sections. There is currently no standardised way to compare such frameworks. We have come up with a number of factors, based on research around the previous models: the Dexter Hypertext Reference Model [79], Munich [104], WebML [30], AHAM [56], [153], LAOS, GAHM [122] and GAM [58]. We hypothesise that the following criteria are important to enable the easy authoring of Adaptive Educational Hypermedia (AEH) courses by non-technical authors.

- Is there any *separation of concerns* between the different aspects of the authoring process?

If there is a separation of concerns between different aspects of the authoring process, authoring of these diverse aspects can be done by different authors independently. For example a subject expert can create the relevant content and use an adaptation strategy created by another author with more technical knowledge.

- Is the *content reusable*?

If the content is reusable, courses, or more importantly parts of courses, can be reused by the original author as well as by other authors. In this way, an author can create a new course which contains parts already made by others. This can both save the author time and increase the quality of the course created.

- Is the *adaptation reusable*?

While reusability of content is common, few models allow for reusability of adaptation strategies or adaptation constructs. The ability to reuse adaptation strategies increases the incentive for authors with the required knowledge to create such a strategy, and it decreases the number of strategies that are needed dramatically. The effect of this can be that authors without the required technical knowledge to write strategies by themselves can still use strategies which were created by other authors and may have been created either for their course or for a different course.

- Are the different *roles* for different users well defined?

Does the model clearly define the different roles, such as teachers and students, or authors and consumers, but also different types of authors? Different users will interact with the model in diverse ways and models that take this into account define the different roles clearly. Moreover,

delimiting the range of functionality a user has access to can be useful if users have different goals, such as differentiating:

- between authors, who wish to create courses, and students who wish to learn the course;
- between users who have different expertise, such as differentiating between teachers, who have as main responsibility teaching a group of students, and authors who have as main responsibility the creation of courses;
- differentiating between content authors, who are experts in the course topics and have the goal to create the best possible material for it, and strategy authors, who are pedagogical experts and can best express the personalised adaptation strategy to deliver the content in, but may not be experts in the specific content.

- Does the model provide *flexibility of formats*?

Does the model itself prescribe a specific format, or allow more freedom to system designers to fit in custom formats according to their own needs? Does the model come with a range of formats accepted for content and adaptation specification, or an easy way to import (or export) other formats, or is it restrictive?

- Does the model provide for *visual processing*?

Visual processing can be a way to enhance the authoring experience, especially for authors who are not technical experts and are unfamiliar with the formats and languages used. Depending on the expertise of a certain author, visual processing may arguably be the only way they would be able to tackle the challenging task of authoring for adaptive courseware.

- Is the adaptation that can be represented with the model *complete*, in other words, can the model represent all types of adaptation outlined in Brusilovsky's taxonomy [21]?

Brusilovsky's taxonomy gives an overview of the different types of adaptation considered important in AH. For the model to be as versatile as possible, it should allow for at least¹⁵ all these types to be described.

- Does the model use a *concept-oriented representation*?

According to the widely used book metaphor [53] the content can be divided into concepts, for example, along the lines of the division of a book in chapters and sub-chapters. This metaphor more closely matches the domain of a course conceptually than, for example, pages. Does the model use this or any other representation via concepts?

- Does the model use *standards*?

Standards allow for importing and exporting of courses. This can greatly increase the portability of the model and decrease the amount of work that needs to be done when moving to or from a specific system.

- For education applications, does the model have a *pedagogical layer*? Or more generally, is there a layer related to its goals?

Reuse of content and of adaptive strategy is very useful and can reduce the amount of work the authors need to do. However in order to be able to reuse content and adaptive strategies separately it is important that the content does not contain any descriptions that are specific to the course and the pedagogy the teacher wishes to use. If there is a separate layer related to the goals of the model, a pedagogical layer for educational applications, this information can be captured separately from the content; hence the content can be reused in other circumstances. For example, what is considered advanced knowledge for first year computer

¹⁵ There are different proposals for extending Brusilovsky's taxonomy and bringing it up to date with modern developments of adaptive systems. However, as this taxonomy is still widely quoted in literature, as well as taught in universities, we consider currently that a system can be complete (necessary condition) if and only if it at least allows for all the types of adaptation which are described by the taxonomy, thus leaving space for the possibility to extend this requirement with a more complete taxonomy in the future.

science students in a course on linear algebra might be basic knowledge to third year mathematics students.

Additionally, we investigate frameworks based on the following supplementary questions.

- What is the *data representation* approach?

Which format and /or approach are used to represent the data? Using this question we want to get an overview of some of the formats and approaches in use.

- What is the *target* system type?

In general we are comparing authoring systems for AEH, however WebML is targeted at general web applications. We add it to the comparison for its visual and UML based approach, so this is important to take into account when making a comparison between systems.

Table 1 shows the AHAM [56], [153], Munich [104], WebML [30], XAHM [29] and LAOS [41] models, and compares their key features side by side. AHAM features reusability of adaptation via CRTs. This means that the types of relationships between concepts can be reused. However in general this does not result in fully reusable strategies. In order to create a strategy, CRTs have to be inserted between domain concepts, creating specific instantiations, as the rules do not generalise over the content. We see that some models use UML or XML. These however are very general purpose standards. As we can see in the table, many models offer some form of visual processing and some form of reusability of the content authored. As we can see, the LAOS model has more of the desired features than the other models. It is the only model in our table to feature reusability of adaptation at the level of whole strategies, as well as parts of strategies, via the LAG language [45], [48]. The combination of visual processing and reusability of adaptation could combine the benefits of a visual authoring interface with the benefits of reusability of adaptation.

Table 1 Comparison between Models of Adaptive (Educational) Hypermedia

	AHAM	Munich	WebML	XAHM	LAOS
Separation of concerns	Yes	Yes	Yes	Yes	Yes
Content Reusability	No	No	Yes	No	Yes
Adaptation Reusability	To some extent at the level of CRTs	No	To an extent	No	Yes
Different user roles	No	No	Yes	No	Yes
Flexibility (different formats?)	Yes	No	No	No	Yes
Visual processing	Yes, via Graph author	Yes, visual representation in OCL	Yes	Graph-theory basis has potential for visual processing	No, but would be possible (it is a tree-based extension to AHAM)
Adaptation complete	Based around canned (hyper-)text adaptation, no explicit support for adaptive multimedia presentations and natural language adaptation	No explicit support for adaptive multimedia presentations and natural language adaptation	No, based around we application s not AH	No explicit support for adaptive multimedia presentations and natural language adaptation	Based around canned (hyper-)text adaptation, no explicit support for adaptive multimedia presentations and natural language adaptation

Table 1 Continued: Comparison between Models of Adaptive (Educational)

Hypermedia

	AHAM	Munich	WebML	XAHM	LAOS
Concept-oriented	Yes	Object-oriented	No	Graphs describe conceptual domain, this is comparable to a concept-oriented approach	Yes
Standards	No	To some extent via UML	To some extent via UML	To some extent via XML	To some extent via XML
Pedagogic layer	No	No	No	No	Yes (via Goal and constraints model)
Data representation	Database /XML	Object-oriented	UML	XML	Database /XML
Target	A(E)HS	A(E)HS	Web App.	AHS	A(E)HS

2.5 Authoring Adaptation Strategies

In the early days, AH adaptation was embedded in the content (for example, prerequisites in Interbook¹⁶ [64] or XML files describing both content and conditional inclusion of content fragment such as in AHA! [54]). While mixing of content and adaptation in delivery seems maybe justified for performance reasons at the authoring time it severely restricts reusability of both content and adaptation. In order for authors to be able to reuse

¹⁶ <http://www.contrib.andrew.cmu.edu/~plb/InterBook.html>

both the content and adaptive specification, a separation was needed. For this purpose, *adaptation languages* were introduced. In this section we will describe the most important adaptation languages so far, the first adaptation language, the *LAG language* [45], [48], the XML-based version which later followed it, the LAG-XLS [136] language and the recently developed GAL [145] language.

2.5.1 The LAG Language

The *LAG language* [45] is a high level adaptation language, which also includes some elements of an adaptation assembly language. It can be used to describe re-usable adaptation strategies. The LAG language has fewer constructs than general programming languages. This makes it both simpler to understand and work with (by humans), as well as easier to process and parse (by machines). The constructs it does have are targeted at adaptation, especially in an AH environment. Each LAG file consists of two parts: the *initialisation* and *implementation* parts.

- 1 The *initialisation* is executed only once, when a new user registers for a course. It handles all the start-up logic required and the setup of user-model variables. It defines what a user will see the first time he or she accesses the adaptive environment.
- 2 The *implementation* part is executed in a continuous loop, whilst the user is accessing the course. This execution phase is thus different from a regular programming language, in the sense that it uses a loop, rather than being executed only once. The adaptive system reacts to user input, such as clicks.

The LAG language allows writing more general rules for a whole course or for a set of concepts rather than just for specific concepts. For example [45] instead of:

```
IF 'The Night Watch' is visited THEN show 'Rembrandt'
```

In LAG one can write the following more general rule:

```
IF Painting is visited THEN show Painter
```

The above is expressed in a LAG strategy as follows:

```
if (UM.DM.Concept.access == true && DM.Concept.type == Painting)
then UM.DM.showing = Painter
if (UM.GM.showing == DM.Concept.type)
then PM.DM.Concept.show = true
```

Below we show how the main principles of adaptation, according to Brusilovsky's taxonomy [21], are supported by the LAG language.

(i) Hierarchy

The LAG language supports adaptation based upon the position of concepts in a hierarchy. The main constructs are the *parent* and *level* properties. The example below shows a concept, if its parent has been accessed:

```
if UM.GM.Concept.parent.access then (GM.Concept.show = true )
```

The example below shows all concepts which are at most of the level stored in the User Model for that course (UM.GM):

```
if GM.Concept.level<= UM.GM.level then (PM.GM.Concept.show=true)
```

(ii) Relations

The LAG language supports adaptation based upon relations between concepts. An example is the following fragment, which shows related concepts if a concept of the 'stereotype' *advanced* is accessed.

```
if (GM.Concept.access && UM.GM.stereotype == adv)
then PM.DM.Concept.Relatedness.Concept.show=true
```

(iii) Domain-concept type

Adaptation can also be based on the *type* of a domain concept. The example below shows all attributes other than the *conclusion* attributes for all concepts.

```
while PM.GM.Concept.type!= conclusion (PM.GM.Concept.show=true)
```

(iv) Metadata (Label/ Weight)

An important part of the LAG language is adaptation based upon metadata, which is found in the weights and labels. The example below shows a concept if the label is equal to the current knowledge level of the user.

```
if (GM.Concept.label == UM.GM.knowlvl)
then (PM.GM.Concept.show = true)
```

Looping through concepts

It is useful to walk through all the concepts, either to find out, for example, how many have a certain label:

```

while (GM.Concept.label == beginner)
  (UM.GM.beginnerLabelCounter +=1)

```

or to set user model overlay variables for all concepts in a course (for example, all concepts in a course have a variable knowledge, initialised to zero):

```

while (true)
  (UM.GM.Concept.knowledge = 0)

```

This type of loop is currently only supported in the *initialization* part of a LAG program.

The example below illustrates a simple strategy called ‘beginner – intermediate – advanced’. This strategy displays concepts to the user, depending on his experience level. The example uses the simpler labels ‘*beg*’, ‘*int*’ and ‘*adv*’ for concepts intended for beginner, intermediate and advanced users respectively (instead of the labels ‘*beginner_title*’, ‘*beginner_text*’, as in section 3.2). The example also uses a number of variables. The ‘show’ variable, which determines if the concept is to be shown, is one of the few core set variables of the LAG language. Other variables are used, for example, to record if a concept has been visited, or how many concepts of a particular group of concepts have been visited. It is more elegant to keep the set of variables as small as possible. Fewer variables make strategies smaller in terms of file size, thus easier to read, and in terms of memory usage, thus performing better.

The initialisation part (below) is performed only the first time the user enters the system; after that, every time the user selects a (lesson) concept, the implementation part (see part 6 below), describing the actual interaction loop, is performed.

```

initialisation(

```

1) General: make every general (unlabeled) concept readable; mark every concept as "not visited yet"

```

  (beenthere =0);
  while (true) (
    PM.GM.Concept.show = true
    UM.GM.Concept.beenthere = 0)

```

2) initialise the number of concepts for beginning to advanced students to 0;

```

  UM.GM.begnum=0 UM.GM.intnum=0 UM.GM.advnum= 0

```


3) count and store the actual number of concepts for beginner students;

```
while GM.Concept.label == beg ( UM.GM.begnum += 1 )  
while (GM.Concept.label == beg) ( PM.GM.begnum +=1 )
```

4) count and store the actual number of concepts for intermediate students;

```
while (GM.Concept.label=int) (   
    PM.GM.Concept.show = false  
    UM.GM.intnum +=1 )
```

5) count and store the actual number of concepts for advanced students;

```
while (GM.Concept.label == adv) (   
    PM.GM.Concept.show = false  
    UM.GM.advnum += 1 )
```

6) set the level of the student to beginner, for the first entry in the system;

```
UM.GM.knowlvl = beg )  
implementation(
```

7) *UM.GM.Concept.beenthere* computes the "number of times a Concept has been

accessed". The following keeps track of how many beginner, intermediate and advanced concepts still need to be visited. These rules are checked each time a concept is accessed.

One concept is not 'aware' of other concepts, however.

```
if (UM.GM.Concept.Access==true) then (   
    if (UM.GM.Concept.beenthere = 0) then  
        if (GM.Concept.label == beg) then(  
            UM.GM.begnum--1      )  
        if (GM.Concept.label == adv) then(  
            UM.GM.advnum--1      )  
        if (GM.Concept.label ==int) then(  
            UM.GM.intnum--1      )  
        UM.GM.Concept.beenthere+=1 )
```

8) Change the stereotype from beginner to intermediate; from intermediate to advanced,

when appropriate; make relevant concepts visible.

The strategy above illustrates a classical case of adaptation, to students of varying knowledge level¹⁷. In chapter 5 we will see more about these limitations and possible solutions to some of the problems that it causes.

¹⁷ For examples of strategies please visit: <http://prolearn.dcs.warwick.ac.uk/strategies.html>

2.5.2 LAG-XLS

LAG-XLS [136] retains the reuse advantages of the LAG framework. In addition it is specifically targeted at supporting frequently used instructional methods to support Learning Styles [136], [34]. LAG-XLS instantiates the Adaptation Language layer of the LAG framework, but with different goals compared to the LAG language. LAG-XLS expresses the following two methods: *selection of media items* (or files, of a particular type of information, according to Brusilovsky's taxonomy [21]) and *ordering information, or sorting of links*, according to Brusilovsky's taxonomy [21], in a simple and straightforward manner. The LAG-XLS language is based upon the LAG language and solves some of its issues, while also simplifying parts of it for its more specific goals of supporting instructional methods and specifically Learning Styles. LAG-XLS is an XML-based language, which aims to align itself with the Semantic Web (see section 2.2, for more information). The language, of which the DTD can be found in [136], consists of the following elements:

- *strategy*: the root element of a strategy file containing an attribute called *name*;
- *description*: the description of the strategy; for example, information about the learner model, intended use and specific requirements (such as specific labels in the content);
- *if*: specifies *if-then-else* rules;
- *condition*: a Boolean expression which may contain user-related information, for example the user's preferred Learning Style; the condition statements only appears within an if statement;
- *then*: the 'then' part of the *if-then-else* construct, defining the actions to be performed on the 'then' path;
- *else*: the 'else' part of the *if-then-else* construct, defining the actions to be performed on the 'else' path.

The following elements describe how the selection and sorting is achieved:

- *select*: chooses a representation to be included in the final presentation, from a set of existing concept representations.

- *sort*: arranges the sequence of concept representations in order of decreasing relevance, depending on the learner's user-model, for example depending on the user's Learning Style.

Both the *select* and *sort* elements have an *attributeName*, to determine which part the author wants to reorder or include in the final presentation. Media items can be ordered in the following ways:

- *showLink*: shows a link to the concept representation;
- *showContent*: shows the content of the concept representation;
- *showDefaultContent*: shows a default content specified by the author;
- *action*: specifies how the User Model is updated; attribute *UMupdate* shows whether this represents an absolute or relative update;
- *UMvariable*: is the UM variable to be updated;
- *expression*: is the expression determining the new value of the UM variable.

2.5.3 GRAPPLE Adaptation Language (GAL)

The GRAPPLE Adaptation Language [145] is an engine independent adaptation language. It is based upon lessons learnt from the LAG [43] adaptation language.

The GAL language is designed to describe the navigational structure of a web application and how this adapts to the actions of its users. Central in the GAL language is the *unit*, an abstract representation of a page, shown to a certain user after a certain request. The data from which units are generated should be accessible via a single RDF query endpoint. Updates are also applied via the endpoint. Below an example of the GAL language [145] is shown. The example adds an adaptive attribute that gives *beginner* or *advanced* information about the concept called *planet*, depending on whether the user has advanced knowledge about the concept.

```
gal:hasAttribute [
  gal:name planetInformation;
  gal:label "Information: ";
  gal:value [
    gal:if [ gal:query
      SELECT ?conceptPlanetInstance
```

```

WHERE { $User :hasConceptInstance ?conceptPlanetInstance
        :name $Planet.name;
        :visited ?Visited;
        :advancedUserValue ?Advanced.
        FILTER (?Visited >= ?Advanced)
    }};
gal:then [$Planet.AdvancedInfo];
gal:else [$Planet.BeginnerInfo];
]
]

```

2.6 Authoring Content for Adaptation

Authoring content for adaptation has been around for as long as AH systems have existed. However, as we have seen in section 2.5, in the past, content often contained embedded adaptation. However, in order to be able to reuse both the content and the adaptive specification, a *separation of concerns* between content and adaptive specification is essential. This section introduces some methods for authoring content for AH, rooted in the separation of concern paradigm. First we describe the CAF format [46], which is based on the LAOS model and describes the content and content-related metadata, populating the Domain Model and Goal and Constraints Model layers, free from adaptive specification. Authoring the content, free from adaptation, is closely related to modelling the specific subject domain conceptually. Therefore this represents an ontology-based method of describing the domain content.

2.6.1 Common Adaptation Format (CAF)

The Common Adaptation Format [46] is a portable, system-independent XML format. It contains common elements related to the way adaptive content is represented in many AH systems. CAF was built to represent a part of the LAOS Adaptive Hypermedia model for authoring, and as such it is a very useful format for sharing and deploying authored AH courses. A CAF file describes a single AH course. The single course is represented by one *Goal Map* from the LAOS framework and all *Domain Maps* (at least one) that are used in the Goal Map. Each Domain Map consists of a hierarchy (tree) of domain concepts. Concepts have sub-concepts as well as domain attributes. The actual content is included in


```

        label CDATA ""
        type CDATA "">

<!--ELEMENT goalmodel (lesson)-->
<!--ELEMENT link (#PCDATA)-->

<!--ATTLIST link
        weight CDATA ""
        label CDATA "">

<!--ELEMENT lesson (link*, lesson*)-->

<!--ATTLIST lesson
        weight CDATA ""
        label CDATA "">

```

As the DTD shows, a CAF file consists of a Domain Model part and a goal model part. The Domain Model part consists of all concept maps (using the concept tag) needed for the goal model. The concept maps in the Domain Model consist of a name, a number of attributes, relations and sub concepts. Each attribute has a name and textual contents, be it a link, HTML or plain text. A relation has a name, and at least one *relationlink* to the related item, containing a weight, label and type.

The Goal Model itself, consists of exactly one lesson. Each lesson consists of a number of links to attributes from certain concepts in the domain maps, and a number of sub lessons. To exemplify how the CAF DTD is used, Figure 4 shows a CAF XML file with one domain map only, called ‘Questions’, with sub-concepts titled ‘Question1’ and so forth. All concepts have a title attribute and the questions have each a question and an answer attribute. The goal model part always has one lesson (since CAF only describes a lesson and the Domain Maps on which it is based). In this example, the ‘lesson’ (goal map) is comprised of the above mentioned questions, in the exact same order and structure as in the original domain map. Note that in the goal map, labels (‘question’ and ‘answer’, respectively) have been added for the questions and answers.

```

<CAF>

  <domainmodel>

    <! ...>

      <concept>

        <name>Questions</name>

      <! ...>

        <concept>

          <name>Question1</name>

          <attribute>

            <name>question</name>

            <contents>Where is Hypertext 2008? </contents>

          </attribute>

          <attribute>

            <name>answer</name>

            <contents>Pittsburgh</contents>

          </attribute>

        </concept>

      < ! ... following concepts omitted >

    </concept>

  </domainmodel>

  <goalmodel>

    <! ...>

    <lesson weight="0" label="question">

      <link weight="0" label="question">

        Questions\Question1\question </link>

      <link weight="0" label="answer">

        Questions\Question1\answer </link>

      </lesson>

    <! ... following lesson parts omitted>

  </goalmodel>

</CAF>

```

Figure 4 Adaptation content example: questions and answers

2.6.2 Ontology Based Content Description

Authoring content for AH, free from adaptation, is closely related to modelling the specific subject domain conceptually. In this section we introduce ontology based methods of describing the domain content.

The Adaptive Course Construction Toolkit (ACCT, see also section 2.7.4) [50], [51] is an authoring system for AH which is based around services and activities for learning. ACCT is an attempt to combine the best of the activities and learning-design driven authoring, such as in the Learning Activity Management System (LAMS) [107] system, and content driven AH authoring methods, such as represented by the AHA! Graph author. The system describes the domain it is teaching conceptually in a so called Subject Matter Concept Space (SMCS). This is a light-weight ontology describing the relationships within a subject matter domain. The ACCT supports the authoring of this Concept Space. It allows the definition of relationships between the concepts. The relationships are provided as patterns that can be customised.

As we will see in chapter 6, this is highly related to the description of the Domain Model in our novel graphical authoring model, the CAM.

2.7 Adaptive Hypermedia Authoring Systems

This section will give a brief description of a few AH authoring systems, in order to provide a view of the state of the art in AH authoring. We will first introduce the *My Online Teacher (MOT)* [42] system, then the authoring capabilities that are combined with the *Adaptive Hypermedia for All (AHA!)* [52], [54] adaptive system. Then we will introduce the *Learning Activity Management System* [107], which is not an AH system, but an educational activity management system. It does however contain an easy to use and popular authoring interface. Finally we will introduce the *Activity Course Construction Toolkit (ACCT)* [50], [51].

2.7.1 My Online Teacher (MOT)

My Online Teacher [42] is an authoring system based on the LAOS model. MOT was one of the first systems specifically designed for authoring of AH. It mainly implements the Domain Model and Goal Model layers in the LAOS authoring framework in a very simple and user-friendly web-forms interface. The adaptation strategy can be described separately, via a text editor or an adaptation strategy authoring tool, using the LAG adaptation language. The main idea behind MOT is that this would be the teachers' interface to AH authoring. A teacher would reuse an adaptation strategy in a LAG file and thus only have to deal with the simple forms in MOT, see for an example Figure 5.

The screenshot displays the 'My On-line Teacher' web interface. On the left, a sidebar titled 'Adaptive Hypermedia (original)' lists a hierarchy of topics: 'Adaptive Hypermedia', 'Authoring for Adaptive Hypermedia', 'AHAM', 'LAG', 'MOT', 'MOT Glossary', 'Learning Styles & AHA!', 'Adaptive educational hypermedia', 'Adaptive On-line Information Systems', 'LAOS', 'Past, Present and Future', 'Why AH?', 'Hypermedia', and 'Adaptive Hypermedia'. Each item is followed by a '[cut]' link. The main content area is titled 'Edit attribute' and contains the following fields: 'Concept name:' with the value 'Adaptive Hypermedia', 'Attribute name:' with the value 'title', and 'Attribute contents:' with a text box containing 'Adaptive Hypermedia'. Below these fields are 'save' and 'Upload' buttons. An 'Upload attribute resource:' section includes a text box, a 'Browse...' button, and an 'Upload' button.

Figure 5 Attribute editing and reading

MOT also has functionalities for copying and linking to courses of other teachers, as well as the possibility to locate and link related concepts across Domain Models. Moreover, MOT allows for metadata addition to lessons for adaptation. These metadata can be added to the Goal Map, or lesson, as the metadata needed for adaptation are usually lesson specific. In the LAOS framework these could theoretically be any metadata, however in

MOT these are currently restricted to labels and weights. This restriction was chosen in order to keep the authoring interface as simple as possible for teachers without a technical background. An example is shown in Figure 6. In the figure we see the same lesson as in Figure 5, but from a Goal Model perspective, and we see we can change labels and weights as a form of metadata.

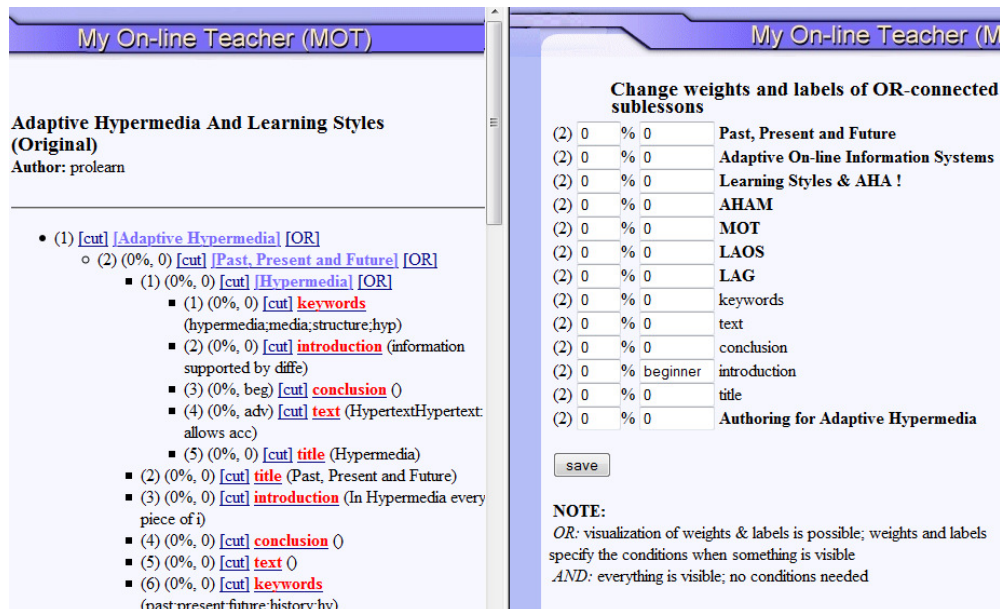


Figure 6 Use of weights and labels in MOT as metadata

The backend of MOT consists of a MySQL database, the design of which is in line with the CAF XML format, allowing easy transformation between lessons in the database and CAF files (see also Figure 7 for the UML diagram of the database). In chapter 3 we will see this in more detail.

The most recent version of MOT, MOT3.0 [67], introduces more usability features, as well as drag and drop functionality, a What You See Is What You Get (WYSIWYG) editor for text and (X)HTML. It also features import using various formats and from various sources (IMS-CP, SCORM, PowerPoint, and Wikipedia). The screenshot of MOT3.0 in Figure 8 shows an important usability improvement, the editable AJAX-based tree of concepts.

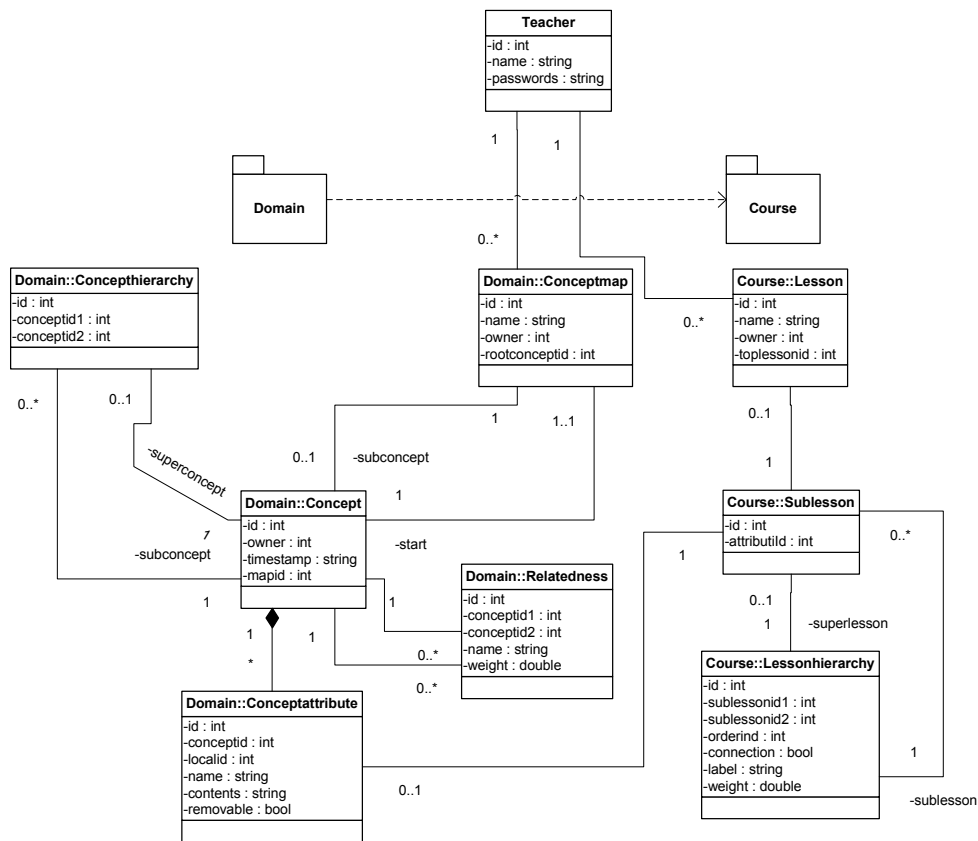
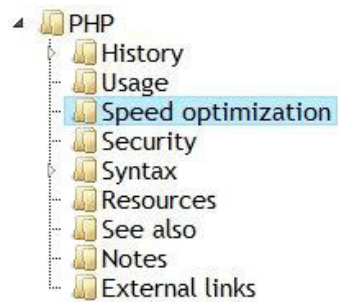


Figure 7 UML diagram of MOT database

PHP by *jonathan*



Speed optimization

Domain Map name: PHP

Creator: jonathan

Attributes:

title

text

Add Attribute

Relatedness relations:

None

Calculate relations with other concepts

Figure 8 The AJAX-based layout of MOT3.0

2.7.1.1 Programming Environment for Adaptation Language (PEAL) Environment for Authoring

Authoring strategies in the LAG language initially lacked tool support. At the start, there were no tools specifically aimed at authoring adaptation strategies, and the only way to tell whether the strategy was correct was to deploy it to an adaptation engine – for example, in AHA! [54], using existing converters [46]. Then an online editor was created, called MOTadapt¹⁹. This editor had some promising features, such as:

- the separation of the strategy description (aimed at the lay person author or domain specialist, to choose an adaptation strategy from),
- allowing only prescribed programming constructs and operations to be inserted (from a list), and also
- allowing for saving both of full strategies as well as procedures, effectively extending the LAG language this way, and allowing authors to reuse their own predefined procedures or procedures developed by other authors.

However, the MOTadapt editor did not allow for the separation between *initialization* (the initial setup of the system, defining what the users first see) and *implementation* (the loop of interaction between the users and the system, triggered as long as conditions hold true), which is a key feature of LAG. Still, this editor was an improvement when compared to using only a text editor, which gives no help and feedback whatsoever.

Recently, a new LAG adaptation strategy authoring tool has been created, called PEAL [45]. The PEAL tool delivers a new AJAX based programming online environment for the LAG language. It allows saving and exporting strategies written in LAG. The editor recognises the LAG language and gives users hints when they violate the grammar. It also features limited auto-completion, as well as a wizard for the creation of the basic code in the implementation section. In Figure 9 a screenshot is shown. As we can see, PEAL also highlights the various constructs of the LAG syntax for easier editing. Additionally, PEAL allows authors to save full strategies and smaller parts of the strategy, as a ‘task’, and

¹⁹ <http://e-learning.dsp.pub.ro/motadapt/>

authors can preview and select task code snippets and insert it into their code, similarly to the ideas in MOTadapt. PEAL also allows authors to save strategies which are not yet completed in their *private* space, and strategies which they wish to share in a *public* space. Thus, reuse of strategies amongst authors is supported, and *collaborative authoring* can take place.

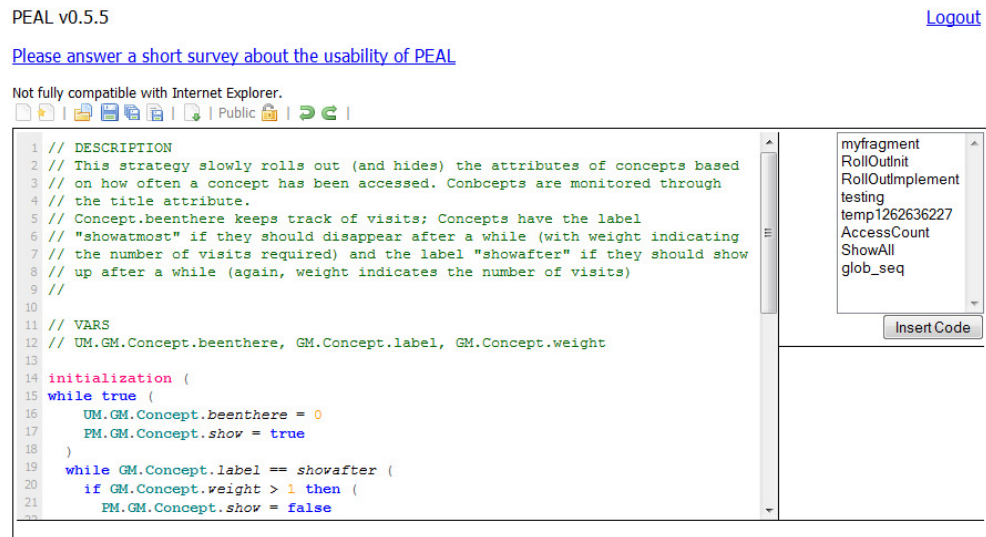


Figure 9 Programming Environment for Adaptation Language (PEAL).

PEAL has been tested in a small scale evaluation [45]. The evaluation showed that the editor was, in general, preferred over text editors. Moreover, PEAL was then deployed during a course on ‘Dynamic Web Based Systems’ for MSc level students in the Computer Science Department at The University of Warwick, in the period between October and December 2009. Results of this use and evaluation are currently still being processed. Overall, the results of the various evaluations showed that tool support for authoring adaptation languages, such as LAG, is indeed needed, and that PEAL presents an important step in that direction.

2.7.2 Adaptive Hypermedia for All (AHA!)

Adaptive Hypermedia for All (AHA!) [52], [54] is an Open Source AH platform, capable of performing content and link adaptation in (X)HTML and XML documents. The AHA! system is a general-purpose system based on the AHAM model [56]. Its uses include, but are not limited to, education.

From an authoring point of view, in early versions of AHA!, the authoring was done using XHTML files. The actions of the user resulted in user model updates, which in turn resulted in adaptation through a set of event-condition-action rules (ECA rules) that the author created by hand in these XHTML files. This was time-consuming and very difficult for non-technical authors. Therefore, the notions of concept Relationships and Concept Relationship Types (CRT) were introduced, and newer versions include the AHA! Concept Editor and Graph Author. The Concept Editor can be used to create and edit concepts, attributes and event condition action rules. The Graph Author allows the definition of concept relationships, and also of generic CRTs, which use placeholders to specify a type of relationship between concepts. In Figure 10 we see a screenshot of the graph editor, creating a presentation on beverages. The AHA! graph author does not assume that the domain is organised as a hierarchical tree. It does however split authoring into two tools, the concept editor and the graph editor, and because it mixes strict domain knowledge and adaptation information, its DM and AM are not very easy to reuse. The graph author works on existing domain concepts, not on generic ones, thus reusability of a graph created by the graph author is very low.

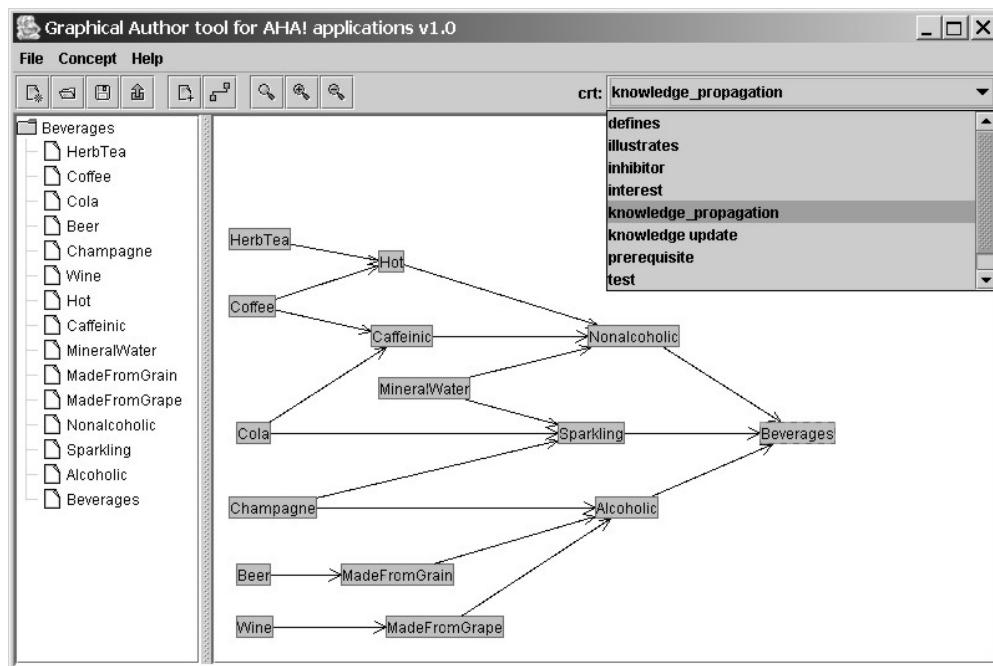


Figure 10 AHA! Graph Author Tool

2.7.3 Learning Activity Management System (LAMS)

The Learning Activity Management System [107] is an authoring system for online collaborative learning activities. It is not an authoring system for generic Adaptive Hypermedia. Its only target is that of education, and it performs authoring of learning activities, and has related goals to authoring of Adaptive Educational Hypermedia. LAMS is mostly known for its graphical interface, which is considered simple and user-friendly. In Figure 11, we see an example of the interface; the example contains a course consisting of several activities chat sessions, documents and media files. LAMS provides a visual environment for creating, storing and reusing sequences of learning activities. The provided interface allows teachers to link together these sequences in a drag and drop fashion. However, LAMS is fundamentally different from Adaptive Hypermedia, in that it is not driven by content, but by activities, and offers no real adaptation. The AH community can however learn from the success of its simple authoring interface.

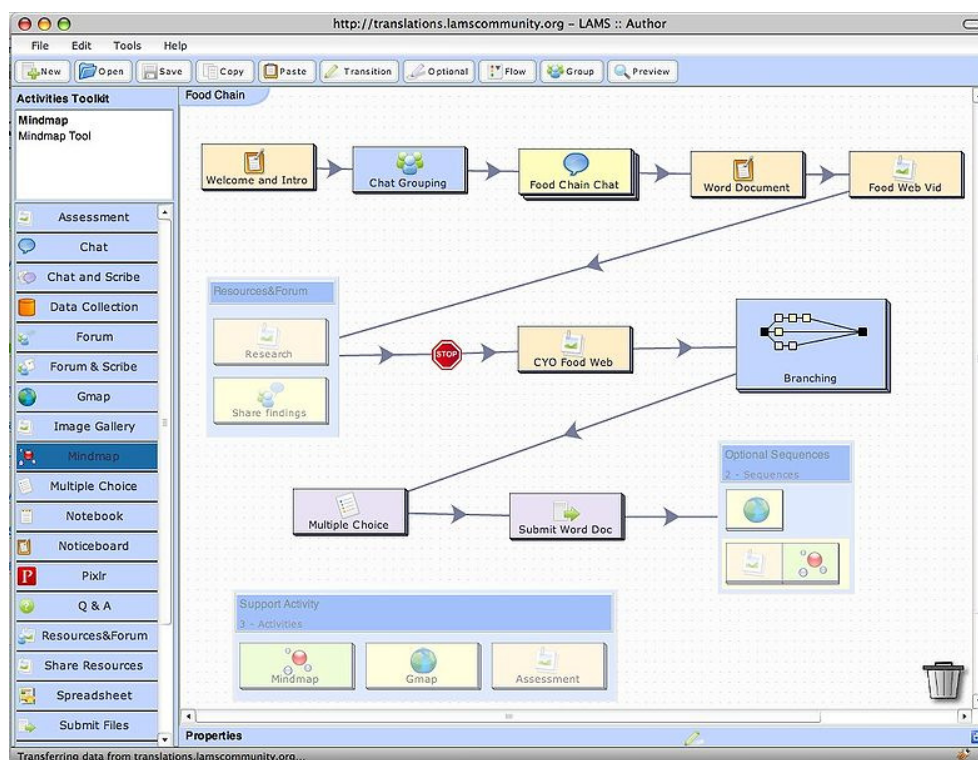


Figure 11 An example of a course in LAMS, image source: <http://wikipedia.org>

2.7.4 Activity Course Construction Toolkit (ACCT)

As mentioned before in section 2.6.2, the Adaptive Course Construction Toolkit (ACCT) [50], [51] is an authoring system for AH that promotes the use of services and activities for learning. The system is based on the ‘Multi-Model, Metadata Driven Approach to Adaptivity’ [35]. The ACCT was the first authoring tool for adaptive e-learning to allow for the combination of a Domain Model (or subject matter concept graph as it is called in ACCT) with adaptive concepts, a User Model and Narrative (pedagogic) template (narratives strategies) via a graphical user interface to design (and extend) adaptive course models. The ACCT allowed the narrative (pedagogic) templates to be selected and extended to develop new adaptive courses. However there was no editor for the creation of new templates and therefore pedagogic templates were predefined in the tool.

2.8 Related Adaptive Hypermedia and E-Learning

Projects

Adaptive Hypermedia is a lively research area and the issues around authoring of Adaptive Hypermedia has been getting a lot of attention recently, In this section we discuss some recent major collaborative research projects that seek to address some of these issues, either implicitly or explicitly.

2.8.1 ProLearn

The EU²⁰ FP6 PROLEARN network of Excellence²¹ (2005-2009) aimed to bring together the most important research groups in the area of professional learning and training, as well as other key organisations and industrial partners. A specific aim was thus bridging the gap between research and education at universities and training and continuous education within companies.

²⁰ EU: European Union

²¹ www.prolearn-project.org/

Our research was partly integrated in PROLEARN and its activities. While our approaches are not explicitly targeted at education within companies, personalised learning environments can also improve learning outcomes in professional education. Indeed, in a corporate setting it is very likely that various employees will have a wide range of prior knowledge and that some people will pick up different parts of the material at different rates. Therefore our approaches to improving the state of the art of authoring of AH and enable educators to create content and adaptation strategy more easily are useful in a corporate educational setting.

2.8.2 Adaptive Learning Space (ALS)

The EU Socrates - Minerva project Adaptive Learning Spaces (ALS)²² (2006-2009) aimed to provide technological means, which can partially compensate lack of face-to-face contact between instructors and learners and amongst learners themselves. To achieve this, ALS worked towards the following sub-goals.

- Increasing the range and amount of guidance and support that AH systems provide to learners and educators.
- Providing novel means to support the social cohesion of groups of learners and to engage the group members in collaborative tasks.

The project developed an openly available software infrastructure, built upon the state-of-the-art in the fields of e-learning and AH systems. The infrastructure supports the creation of personalised learning spaces, with a focus on learning activities, where learners are active members of their learning environments, instead of passive consumers of learning content.

Our research also contributed to the ALS project, to some extent. Our approach to automatic content addition (see mainly chapter 3) works towards the first sub-goal, increasing the amount of support and guidance available to educators at authoring time. Furthermore, our comparison of AEH and IMS-LD (see chapter 4) has been performed in

²² <http://www.als-project.org/>

the light of the second goal. The comparison investigated the general similarities and differences between AEH and IMS-LD, and also whether AEH can be used in systems where students are treated as active participants engaging in collaborative tasks, rather than passive consumers of content.

2.8.3 Generic Responsive Adaptive Personalised Learning Environment (GRAPPLE)

The GRAPPLE²³ EU FP7 STREP project aims to create a technology-enhanced learning (TEL) environment that guides learners through a life-long learning experience by adapting to preferences, knowledge, skills and learning goals, as well as the personal or social context.

The use of such AH systems has not been as widespread as one might expect, considering the appeal of personalisation and customisation. To overcome this issue, GRAPPLE incorporates its AH environment seamlessly into learning management systems which are in widespread use. In order to further promote acceptance, several training events are organised by GRAPPLE and once prototypes have been built, GRAPPLE will evaluate the usability and benefits of using adaptive TEL for the learning outcomes.

In order for educators to be able to create courses in the envisioned integrated TEL environment, authoring tools are required. Moreover, if the GRAPPLE environment is to find widespread acceptance, these tools have to be accessible and usable by educators without a technical background and knowledge of the underlying techniques and formats. The authoring tools enable educators to provide adaptive learning material to the learners. These authoring tools provide simple and usable interfaces for creating or importing content, designing learning activities and defining pedagogical properties of, and adaptation strategies for, the content and activities. This is done by introducing a novel graphical model for authoring AH and building an integrated authoring tool based upon that model (see also chapter 6). The last and most recent part of our research is a part of the research

²³ <http://www.grapple-project.org/>

on authoring within the GRAPPLE project. Chapter 3 gives more details on the tool developed within this context, the CAM authoring tool.

3 Automatic Content Addition

Authoring of adaptive hypermedia is a notoriously difficult endeavour [28], although its results can be extremely valuable, generating, for example, in the educational context, personalised (learning) experiences [21]. A solution to this problem is to use as much automatically generated authoring as possible, instead of authoring by hand. There is some research into how to automate authoring in different ways [33], [17], [110], [147]. A good basis is to use already annotated resources, which can be automatically retrieved when necessary, as dictated by the authoring process. A rich source of information that we found can be exploited in this sense is the Semantic Desktop [33], [135]. In the Semantic Desktop, resources can be categorised by rich ontologies, and semantic links express various kinds of semantic relationships between these resources. For example, the Semantic Desktop stores not only the name of a document, but also information about where this document was created, when and by whom, which of the colleagues sent it, and how often and in what context it was accessed. All these metadata are generated automatically, by the appropriate applications and stored in an application independent way as RDF metadata [129] in the user's personal data store. This rich set of metadata clearly makes it easier for the user or applications to (semi-)automatically retrieve appropriate material for different contexts: for example, when an author wants to select appropriate materials for a lecture. Of course, in the latter context, the author still has to create some basic lesson material, serving as a retrieval framework.

In [95], we described the interaction and exchange of data between the Beagle++ environment [11], [31], which is an advanced search and indexing engine for the Semantic Desktop, generating and utilizing metadata information and the adaptive hypermedia authoring environment MOT (My Online Teacher) [42], [119], a sophisticated system for authoring personalised e-courses.

The objective of this chapter is to show that tools can be developed which harvest multiple alternative content which is both highly relevant for the course and is aligned with the

teaching strategy for the course under construction. This can for example be done by supplementing the available content for the course under construction, with content already stored/accessible on the author's desktop. The chapter will also show evaluations of the relevance of the harvested content and alignment to its teaching strategy. The chapter is my own work. The ideas were formed during an MSc dissertation, however the finalisation of the ideas as well as both prototypes and evaluation rounds were conducted during the PhD degree. The work has been previously published in [89], [90], [91], [92], [93], [94], [95], in which the co-authors have acted as mentors.

The rest of this chapter is organised as follows. In section 3.1 a scenario motivating the need and use of automatic content addition is given. Section 3.2 introduces our approach to automatic content addition in line with the scenario. Section 3.3 describes the prototype, which. It is evaluated in section 3.4 in a number of evaluation steps. Based upon the results of the evaluation, a second prototype, described in section 3.5, was built. In section 3.6 we show how the second prototype was evaluated. Section 3.7 introduces both the state of the art in authoring of Adaptive Hypermedia as well as in the Semantic Desktop and finally section 3.8 draws conclusions about our approach.

3.1 Authoring and Learning Scenario

Consider Dr. Brown who is preparing a new on-line course on early 19th century European history for first year undergraduate history students. The university allocates a limited amount of time for this. Adaptive Hypermedia is considered to result in a better learning experience [21] and he wants to benefit from automatic help during authoring. Therefore he decides to use MOT [42], an adaptive educational hypermedia authoring environment with adaptive authoring support. This decision means that he would only have to author a standard basic course, as he usually does. Still, this costs him slightly more time, as he has to divide his course into conceptual entities and add semantic labels. After this, however, the authoring system can automatically enrich the course based on (pedagogical) strategies chosen by Dr. Brown. For example, the system can consider the version of the course

created by Dr. Brown as the version for beginner students²⁴, who do not aspire to higher grades or deep knowledge on the subject. For advanced students, who want to achieve high grades, or deeper knowledge, the adaptive authoring system can use a semantic personal desktop search to automatically find any existing scientific papers that are relevant to the current course on Dr. Brown's desktop. These scientific papers can be used as alternative or additional material to the main storyline of the static course. This mechanism builds upon a number of assumptions.

- Dr. Brown is a specialist in the subject taught or at least has advanced knowledge of and a reasonable amount of interest in the subject. He will therefore at least read papers of interest on the subject, which are likely to be stored on his computer.
- Research papers on the subject taught are considered as useful extra resources for advanced learners.
- Dr. Brown has been using Beagle++ Semantic Desktop System [11], [31] to store both papers and all relevant metadata automatically in RDF format.

The storing of papers will have taken place over many years and Dr. Brown may not know exactly where on his computer each article relevant to the current course is. However, the Beagle++ Semantic Desktop System can be used by the authoring tool to retrieve relevant papers found on Dr. Brown's desktop. He then selects the papers he wants²⁵ and fine-tunes the semantic labels for the chosen adaptation strategy.

Now, with relatively very little effort, Dr Brown has created an adaptive course, which will provide a richer learning experience for student.

²⁴ Note that the approach is not limited to a beginner- intermediate- advanced strategy, a teacher can decide to use any strategy he likes and set labels and weights accordingly. It is always a difficult trade-off between offering as rich a functionality as possible while keeping the interface as simple as possible. For this reason we have provided default settings for many options. The defaults will not always be ideal, but should provide a workable result.

An other example that could be employed is, e.g., a resource type-based strategy, see:

<http://prolearn.dcs.warwick.ac.uk/strategies.html>

²⁵ This extra selection step is necessary for privacy: Dr. Brown needs to approve which papers from his desktop are to be shown to the students.

3.2 The Approach

In this section we introduce our method and system setup. As can be seen in Figure 12, Beagle++, the Semantic Desktop Environment used in our prototype, stores all metadata in the Sesame RDF database [133]. All Beagle++ components that generate metadata (for example, the email, publication, web cache and file metadata generators) add metadata to this database. All Beagle++ components which use metadata (for example, the search and indexing module, the ranking module or the browsing modules) retrieve their data from this repository and, in some cases, write back new data (such as the PageRank value for documents or other resources). It is easy to accommodate additional modules in this environment by writing appropriate interface components, which read and write from this repository. This is what we have done for MOT [42], [119]. We have focused on the semi-automatic addition of articles stored on the user's desktop to a MOT lesson [42]. This represents an instance of the concept of adaptive authoring: authoring that adapts to the author's needs. In MOT, the addition is done to an existing lesson. Based on pedagogic goals, the author can then process the data by adding more information on the article after the conversion.

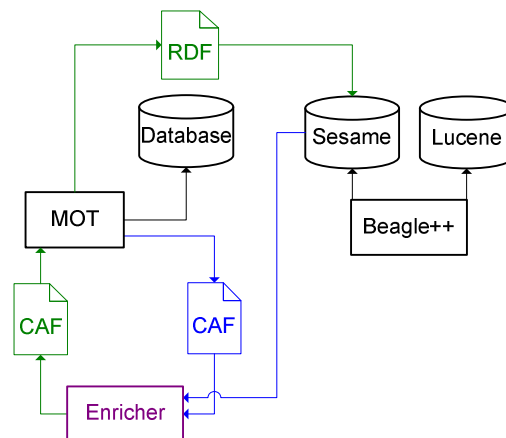


Figure 12 System overview of Enricher

These additions can then be fed back into the RDF store, if necessary. We use CAF (Common Adaptation Format) [47], a system-independent XML exchange format, to simplify the transformation process from RDF to the MOT MySQL storage format.

3.2.1 Metadata Schemas

As highlighted by the scenario, both automatically generated metadata as well as manually added metadata are necessary. *Automatically generated* metadata enable description and retrieval of appropriate articles. *Manual annotation* allows a higher level of control, addition of extra content, and attributes such as pedagogical weights and labels which are necessary to build the final adaptive course product. The next sub-sections detail these two kinds of metadata, from the point of view of the two worlds that come together: *Semantic Desktop* and *authoring of adaptive hypermedia*.

3.2.1.1 Metadata in the Semantic Desktop

Beagle++ [11], [32] is an extension of *Beagle* [10] that generates and stores additional metadata describing the documents indexed by *Beagle*, as well as other resources and their relationships on a Semantic Desktop. Such additional metadata automatically annotate material the user has read, used, written or commented upon. Applications that generate such desktop behaviour metadata include: applications for file access or editing on the desktop, Internet browsing and files stored from the Internet, and mail exchanges with their respective files stored from mail attachments [32], [74]. Figure 13 depicts an instance of this ontology illustrating files annotated with their publication metadata, file system attributes, web history and mailing context (for example, the fact that files are attached to specific e-mails). In Figure 13, the rectangle in the upper left part describes e-mails (with subject, body, sender and status attributes) and their attachments; the rectangle in the upper right part shows publications written by different authors at various conferences (with attributes such as title and publication year); the rectangle in the lower left part displays the web cache history (web pages visited and dates of visits). The rectangle in the lower right part of the figure shows files which were saved on a person's desktop, with their name and specific directory. Files on the desktop may have been saved via any of the other three processes (from emails, from websites, or from conferences), so an attachment entity and a file entity may refer to the same object. There is currently however no version control in *Beagle++*, so no clear relationship can be found between versions of the same file.

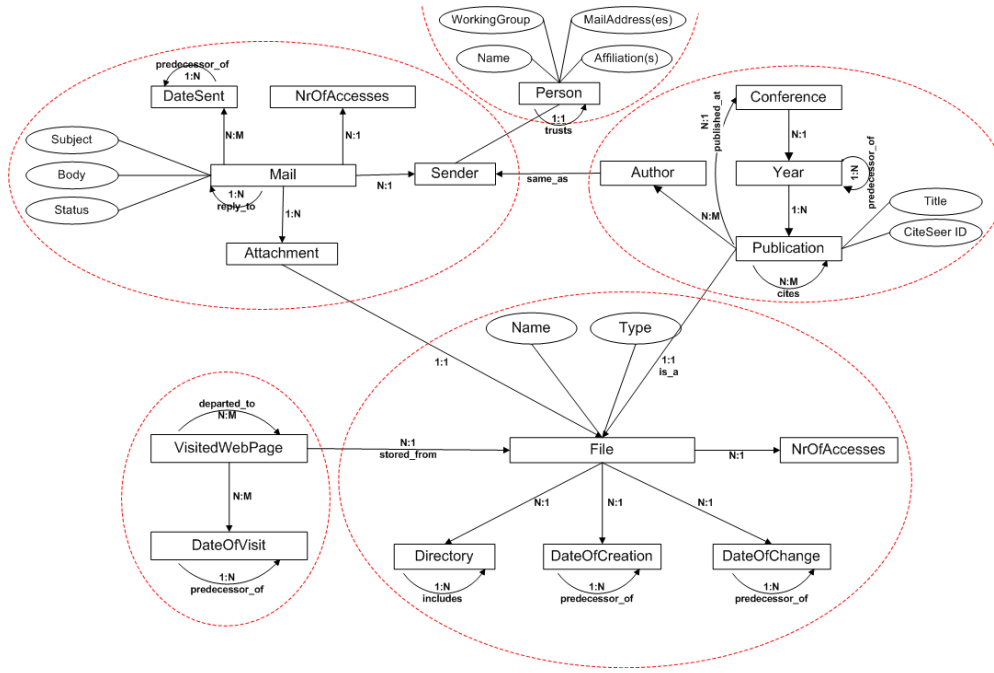


Figure 13 RDF schema of Semantic Desktop metadata in Beagle++

3.2.2 Enriching the Domain Model and Goal Model with Metadata

As MOT is mainly a tool for authoring educational (adaptive) material, the internal information structures are based on strict hierarchies (as advocated in [42]). When enriching the domain-models and lessons, we therefore aim at getting the right information in the right place in this hierarchy. To achieve this, the program first queries the Sesame database, using as search terms title and keywords of each domain concept found in the current existing lesson. The RDF query in the SeRQL [18] language appears as follows:

```
select x from x {p} y where y like "**keyword" ignore case
```

Some alternative retrieval methods have been studied and implemented, as follows. For computing the mutual relevance between a given article and a given concept, in order to decide the appropriate place of articles in the concept hierarchy, we have developed two slightly different theoretical alternatives, called Concept-oriented and Article-oriented methods. In the next sections we will discuss the ranking alternatives, as well as some options that influence the ranking results.

3.2.3 Concept-oriented Relevance Ranking Method

The first relevance ranking method is computing relevance according to

Equation 1:

Equation 1 Concept oriented ranking

$$rank(a,c) = \frac{|k(c) \cap k(a)|}{|k(a)|}$$

where:

$rank(a,c)$ is the rank of article a with respect to the current domain concept c ;

$k(c)$ is the set of keywords belonging to the current domain concept c ;

$k(a)$ is the set of keywords belonging to the current article a ;

$|S|$ = the cardinality of the set S , for a given set S .

This formula is concept-oriented, in the sense that articles ‘battle’ for the same concept: a given article is placed in the appropriate place in the hierarchy by it.

The actual implementation of the concept-oriented formula is not trivial for the RDF database that we have, as there is no keyword property available for retrieved objects (articles). Therefore, these keywords have to be extracted from the title property of the object. This means that the computation of the cardinality of the keywords, $|k(a)|$, can be interpreted in many ways. The easiest way is to set it to the number of words in the title property. Another option is to compute the number of relevant keywords (for example, eliminating stop-words such as ‘the’ and ‘a’) by connecting them to a given ontology or dictionary, such as WordNet.

3.2.4 Article-oriented Relevance Ranking Method

A second implementation of the rank is given by Equation 2.

Equation 2 Article oriented ranking

$$rank(a,c) = \frac{|k(c) \cap k(a)|}{|k(c)|}$$

The equation shows how many of the keywords (shared by the article and the concept) are present in the concept. For example, if a concept has fewer keywords than another one, but the keywords shared between article and concept are the same, the former concept will have a higher rank and ‘win’ the article. This formula thus shows to which of the concepts the

article should be attributed. Therefore, this formula is article-oriented, in the sense that concepts ‘battle’ for the same article.

3.2.5 Sets versus Multisets in Computation of Relevance Ranking

Next, once the formula is chosen, there is another possible distinction to be made: we have chosen to implement two different versions for computing the cardinality of the intersection, one set-based (with intersection operation on sets, as defined above) and one with multisets or bags (and the respective intersection operation on bags).

In set theory, a bag can be formally defined as a pair (A, m) where A is some set and $m: A \rightarrow \mathbb{N}$ is a function from A to the set $\mathbb{N} = \{1, 2, 3, \dots\}$ of (positive) natural numbers. The set A is called the underlying set of elements. For each $a \in A$ the multiplicity (number of occurrences) of a is the number $m(a)$.

The intersection can then be defined as:

$$(A \cap B, f(A \cap B)), \text{ where: } f(x) = \min\{m(x \in A), m(x \in B)\}.$$

$$\text{For example: } \{(a, 2), (b, 1)\} \cap \{(a, 3), (x, 1)\} = \{(a, 2)\}$$

The reason to use bags instead of sets is that the number of times keywords appear in certain texts can be relevant in itself (not just which keywords). A text containing a greater number of occurrences of a specific keyword could be a better match for that keyword than a text with only one occurrence of the respective keyword.

3.2.6 Allowing Duplicates Among Sibling Concepts in the Concept Hierarchy

The same resource may be relevant in more than one place within the hierarchy. In that case, the resource will be added to the place where it has the highest relevance, by default. If there are more places in the hierarchy with a value equal to the highest relevance, the one with the higher position in the tree wins. If, however, there are siblings with the same position in the tree and with the (same) highest relevance, a decision has to be made, either to allow duplicates, or to select randomly one of the candidate sibling concepts and allocate the resource to it. The ‘*allow duplicates*’ option in the Sesame2MOT Enricher v1 application controls this particular option

3.2.7 Adding Metadata as Separate Concepts or as Attributes

The retrieved metadata also have a structure. For example, a retrieved paper might have a location it was presented at and a year it was presented in. These metadata can be added either as attributes of the new article concept in MOT, or as a set of new sub-concepts, with their own attributes. As we can see in Figure 14, with the ‘Add metadata as separate concepts’ option, the author can select between these two possibilities in the Sesame2MOT Enricher v1 application.

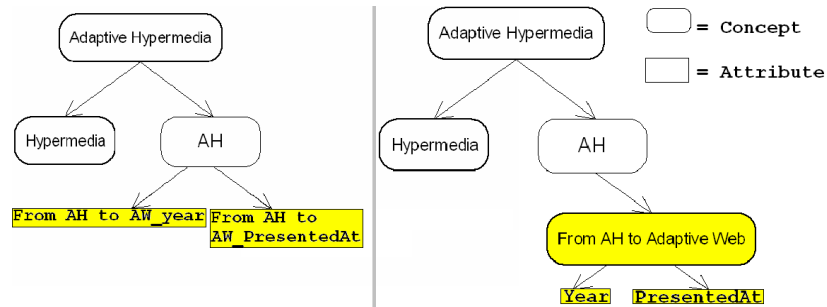


Figure 14 Right side, result with 'add metadata as separate concepts', left: without

3.2.8 Handling Flexible Schemas

The previous sections implicitly assume fixed schemas, that is, all schema elements, relationships and attributes are defined and fixed. In reality, however, our Sesame data store contains data based on different schemas and different versions of these schemas, as metadata and metadata schemas continuously evolve on the Semantic Desktop. Although this does not create great problems in our transformation process (we store all schemas and schema versions together with our metadata in RDF/RDFS format), it can lead to problems for metadata generation rules. For example, rules that specifically refer to certain elements or attributes in a given schema are not viable (cannot be reused) if the schema evolves. The solution we propose in this paper is based on malleable schemas, which allow us to flexibly describe our metadata as well as employ imprecise matching over these schemas, to flexibly refer to the appropriate attributes.

3.2.8.1 Extended Malleable Schema

Malleable Schemas (introduced in [60]) are a new concept in use in the database and web communities. Although problems with integration of information on the web have been

recognised early (for example various schemas and non-structured data), solutions proposed often involve either individual *mediating or merging of two or more schemas* [24], or *mapping to a higher level schema* [71], [133] or *ontology* [70]. More advanced solutions deal with *gradually evolving schemas* [142].

Malleable schemas provide a mechanism by which the modeller can capture the imprecise aspects of a domain during the modelling phase in a well principled fashion [60]. This is done based on keyword matching; however, these keywords are elements of the schema, instead of arbitrary keyword fields. Unlike earlier solutions, malleable schemas provide an answer to more than one problem when integrating information on the web and especially on the desktop, including:

- multiple original schemas with varying semantics;
- evolution of schemas and semantics;
- the need for only partial integration (as opposed to full integration), since often only a part of the existing schema is relevant for a particular application or user, and integrating the whole schema can be both an demanding as well as a superfluous exercise.

Malleable schemas help in capturing the important (relevant) aspects of a domain at modelling time, without having to commit to a strict schema. More importantly, the vague parts of the schema can, in principle, evolve later towards more structure, or can just as well remain as they are.

The data model of malleable schemas as described in [60] is an object oriented one, which matches the RDF/RDFS data model [129]. RDF/RDFS represents information as triples *<subject, predicate, object>* where subject is represented by *class* elements of the schema (or instances thereof), the predicate represents a *relationship* or an *attribute* and object can be another class element, an instance thereof, or a value. Relationships have a *domain*, where the domain is a *set of classes*. Each attribute has a *range*, specifying *allowed values* for the attribute. Malleable schemas are composed of similar elements to regular schemas. The major differences to conventional schemas are as follows.

- Classes and relationships (properties) do not have to be named precisely the names can be keywords or phrases, as well as regular expressions, and a distinction is made between precisely and imprecisely named elements.
- Structure can vary a distinction is made between strict structure and variable structure elements. The latter allow us to flexibly refer to classes in paths composed of more than one property.

Malleable schemas thus provide a simplified, unified view on a collection of related and overlapping base schemas. The relationships and overlaps of the base schemas are described by mappings. An example schema is presented in Figure 15.

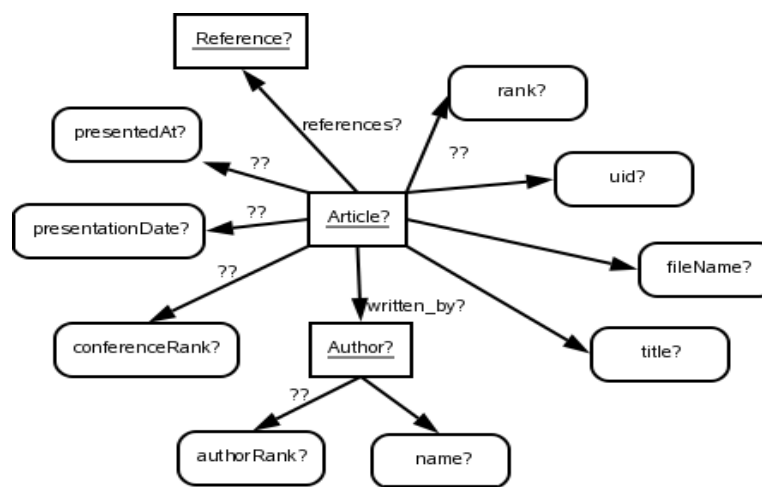


Figure 15 RDF malleable schema of directory data

Figure 15 shows the schema on which queries between MOT and the Semantic Desktop data in Beagle++ are made, taking into account the possibility that the schema may evolve, as discussed in sub-section 3.2.8.2.

Term imprecision: Flexible class and attribute names

In Figure 15, imprecision is denoted via *single question marks* '?'. For instance, the entity 'Article' is imprecise (or malleable) and is written as 'Article?'. In other words, a malleable schema allows imprecision in the naming of classes. For example, 'File' can sometimes appear as 'Article', sometimes as 'Artikel'. In the latter, the language has to be identified, whilst in the former, synonyms have to be searched for. Such an uncertainty about class names can appear, as listed above, when the same class has different names in two different schemas, but also, when the target schema that is queried is not known. This

malleable feature of schemas is based on the ones proposed in [60]. In the following, the malleability definition in [60] is extended, based on the needs we have found in our transformation and conversion exercise between schemas.

Term imprecision: Flexible relations or property names

Just as classes can be imprecise, properties of classes can also be only partially known. For instance, we can expect that the malleable class ‘*Author?*’ can have a property ‘*name*’, but it could also be ‘*firstName*’ and ‘*lastName*’, or be expressed in another form. Therefore, we depict this expectation by adding to our schema the malleable property ‘*name?*’. Again, the question mark ‘?’ denotes the imprecision.

Furthermore, composed words can appear, such as in ‘*presentedAt*’ relating to ‘*conference*’. In such a case, synonyms are not enough. First, decomposition into simple words has to be performed, and then the expression identified.

Flexible paths

Beside the naming differences, schemas can also have different structures. A class can be connected to another class or property via a direct connection, or via an imprecise, indirect connection. In our example (Figure 15), the imprecise attribute ‘*presentedAt?*’ can be directly or indirectly connected to the imprecise object ‘*Article?*’. This would correspond in Figure 13 to the relationship between the (not displayed) property ‘*Name*’ of the class ‘*Conference*’, to the entity ‘*Publication*’. ‘*Name*’ in Figure 13 is not a direct attribute of the class ‘*Publication*’. Therefore, in Figure 15, the property ‘*presentedAt?*’ (equivalent to the above mentioned property ‘*Name*’ in Figure 13), appears as an imprecisely linked property of the class ‘*Article?*’ (equivalent to the above mentioned class ‘*Publication*’ in Figure 13). In Figure 15, such indirect, imprecise connections (of undetermined length) are marked by double question marks ‘??’.

The above three types of imprecision can be resolved in various ways. *Term similarity* [60], for example, can be resolved via WordNet [152] or by other similarity measures. We can rely on extensive research regarding similarity measures between individual schemas

(*instance similarity* [60] or *structural similarity*: for example, editing distance [153], approximate nearest neighbour search [138], unordered tree matching [133] and *schema-corpus similarity* [79]).

It is important to note that, for malleable schemas, complete resolution of the imprecision and schema mapping is not desirable. The resolution is only interesting and important when specific queries are posed against the malleable schema. The elements of the malleable schema which do not appear in the queries do not need to be bound to concrete schema instances or schema alternatives. For malleable schemas, the resolution process of the imprecision is an *iterative discovery process*.

3.2.8.2 Queries and Rules with Malleable Schema Elements

The next issue is how the query process takes place in the presence of malleable schemas and how end-user (for example, learner) adaptation (personalisation) based on malleable schemas can be implemented. As hinted in [60], there are two main query targets: *instances* and *schemas*. Whilst querying based on regular schemas usually targets the instances, for malleable schemas, asking questions about the structure and composition of the schema itself can also be interesting as it is also, to some degree, unknown.

Instance querying

Instance querying is the type of querying usually performed: a schema instance is queried about the instances of elements that it contains. For example, we want to search for articles written by certain authors. We base our instance query system on [60] by expanding it, though, as we want to express the imprecision of *terms* (classes or properties), as well as the imprecision of *structure* in our queries.

In the following, we exemplify these three types of imprecision in a query. Let us suppose we want to find an article written by author “Cristea”, which was sent to the current user by “Nejdl” via email. First, in Example 1, we look at how we would write this query if we knew the exact structure of the RDFS base schema (as depicted in Figure 13). As we are

querying RDF data, the language uses a simplified form of SeRQL from Sesame; moreover, this was the implementation language used for our transformation steps described in the first part of this chapter.

Example 1 - Standard query:

```
SELECT Title FROM { bplus:Publication} art:title {bplus:Title},
    { bplus:Publication} art:authored_by { bplus:Author},
    { bplus:Publication} rdfs:subClassOf { bplus:File},
    { bplus:Attachment} rdfs:subClassOf { bplus:File},
    {Mail} bplus: Attachment{ bplus:Attachment};
    bplus: email_reply_to { bplus:Sender},
    {Sender} bplus:name {"Nejdl"}
WHERE bplus:Publication = bplus:Attachment
    AND { bplus:Author} like "##Cristea"

using namespace
    rdfs = <http://www.w3.org/1999/02/22-rdf-syntax-ns#>,
    art = < http://www.win.tue.nl/~acristea/
        schemas/articleSchema.rdf#>
    bplus = <http://www.l3s.de/domain_l3s#>
```

The example returns the desired article(s), if the terminology, as well as the relations used in these query, are correct. We will now gradually relax this query towards a malleable query.

In Example 2, we rephrase this query, based *partially* on the malleable schema in Figure 15. In this case, we are not aware of the precise naming and terminology of the classes of the basis schemas; moreover, we cannot bind the terminology to a given namespace.

Example 2 - Term imprecision - class imprecision:

```
SELECT Title? FROM {Article?} title {Title?},
    {Article?} authored_by {Author?},
    {Mail?} Attachment {Article?};
    Sender {Sender?},
    {Sender?} name {"Nejdl"}
WHERE {Author?} like "##Cristea"
```

As can be seen, Example 2 still assumes some knowledge of the correct terminology of properties, as well as knowledge about the structure, in order to return the desired article(s).

Imprecision is introduced via the imprecise term ‘*Article?*’, which can be mapped to several classes of the original schema(s): ‘*Publication*’, ‘*File*’ and ‘*Attachment*’. As we can see, the imprecision of terms has the side-effect that queries can be written in a more compact way. Explicit reference to equivalent classes is not necessary in the query.

The following relaxation allows for properties to be also imprecise (Example 3).

Example 3 - Term imprecision - property imprecision:

```
SELECT Title? FROM {Article?} title? {Title?},
      {Article?} written_by? {Author?},
      {Mail?} has_attached? {Article?};
      has_sender? {Sender?},
      {Sender?} name { "Nejdl" }
WHERE {Author?} like "*#Cristea"
```

In Example 3, properties such as ‘*written_by?*’ are imprecise. This property will have to be connected during the resolution process with the property ‘*authored_by*’ in Figure 13. This step allows all terms of the query to be vague. This corresponds to the situation where either the actual names of the classes, attributes and properties are forgotten, or they are unknown. The structure of the RDF schema, however, still has to be known.

Finally, we make another relaxation step, as depicted in Examples 4a and 4b. Example 4a is based on the malleable schema in Figure 15, extended with mail information.

Example 4a- Relation imprecision:

```
SELECT Article? FROM {Article?} written_by? {Author?};
      ??sent_by? { "Nejdl" },
      {Author?} name? { "Cristea" }
```

The relation ‘*??sent_by?*’ is both imprecise in *terminology* (it should be equivalent to ‘*email_reply_to*’ in Figure 13.); this is why it has a question mark on its right side: ‘*sent_by?*’ ; as well as imprecise in *structure* (it actually represents a path between ‘*Attachment*’ and ‘*Sender*’ from Figure 13., for example, ‘*Attachment<has_attached-Mail-email_reply_to>Sender*’; this is the reason why it also has two question marks on its left side: ‘*??sent_by?*’).

The query can be relaxed even further, as in Example 4b. This corresponds to a simplified malleable schema, a reduced version of the schema from Figure 15. In Example 4b, the connection between ‘*Article*’ and author name is imprecise as well.

Example 4b - Relation imprecision:

```
SELECT Article? FROM {Article?} ??written_by? {"Cristea"};
                                ??sent_by? {"Nejdl"}
```

Notice how in queries 4a and 4b the query has become more compact and almost like natural language. This resemblance is due to the fact that the malleable query is similar to the way people ask questions: they only remember partially terminology and relations and some information about the structure. Moreover, the individual representations remembered by different people differ. Still, individuals are capable of communicating with each other despite these different models (schemas), which provides additional motivation to express uncertainties in the world around us through mechanisms like malleable schemas and malleable queries.

Schema querying

Schema querying is only necessary when dealing with imprecise schemas [60]. As our malleable schemas are imprecise both in the *naming* of the elements (classes and properties) and in their *structure*, schema queries can serve to identify the actual naming and structure of the base schemas. This is useful in resolving the imprecision of our malleable schema.

In the following, we exemplify the three types of imprecision defined above, for a proposed schema query.

In our first example, we want to find the correct term for a *class* or an *attribute*. Here, we want to know how the class ‘*Article?*’ is called in the base schema(s), which is connected via a ‘*written_by?*’-like property with a class similar to ‘*Author?*’.

Example 5a - Term imprecision - class imprecision:

```
SELECT Name(Article?)
FROM {Article?} written_by? {Author?}
```

For the base schemas in Figure 13., the answer would be ‘*Publication*’. If we also wanted to obtain ‘Attachment’ and ‘File’, we would need to relax the triple conditions, as in Example 5b.

Example 5b - Term imprecision - class imprecision:

```
SELECT Name(Article?)
FROM {Article?} * {*}
```

In our second example, we want to find the correct term for a *property*. The property name we’re looking for is ‘*written_by?*’, from the same triple as above.

Example 6 - Term imprecision - property imprecision:

```
SELECT Name(written_by?)
FROM {Article?} written_by? {Author?}
```

The query result is the property ‘*authored_by*’ from the schema in Figure 13. In our third and last example, we want to find the *correct path* between two objects (two classes, or a class and an attribute). We exemplify this by looking for the conference rank of a given article (Example 6).

Example 6 - Relation imprecision:

```
SELECT Path(??conferenceRank?)
FROM {Article?} ??conferenceRank? {ConferenceRank?}
```

The result will retrieve the path between ‘Publication’ and ‘rank’, for example, ‘published_at - Conference - has_a’ (from Figure 13).

The two types of queries (instance and schema query) defined and exemplified above can be performed automatically by search systems, in order to automatically enrich, in our current application, the authored learning material for the adaptive, personalised learning environment of the end-user, the student. The author is expected to use malleable schemas

transparently and leave the processing to a query system and concentrate only on the extra manual annotation and extension of the material.

3.3 First Prototype

The initial prototype was built as a stand-alone Java application, due to the fact that it was supposed to be placed on the same machine as the Semantic Desktop application and it needed the access rights to the articles. The processing steps to be done by authors were as follows:

step 1) download a CAF XML file from the MOT authoring system,

step 2) load this CAF XML file into the prototype,

step 3) add resources as described in the previous section,

step 4) save the CAF XML file, and

step 5) upload this new, updated CAF XML file to the MOT authoring system again.

The interface interaction with users is depicted in Figure 16. The first two steps, to be performed in the MOT authoring system, are not shown, as the focus is on the Sesame2MOT Enricher v1 application. Figure 16 displays the resource addition in step 3. Authors can choose between concept-oriented versus article-oriented processing, between allowing (or not allowing) duplicates among siblings, between computing the resource keywords as sets or as multi-sets, and between adding metadata about articles as separate concepts or as attributes of the matching domain concept. All these options conform to the theoretical description in the previous section. Moreover, the author has the option to change the default way that the new resources will be labelled: they would by default be labelled as advanced, with a weight of 50. If the author would like to later use a different educational adaptation strategy with this course, he could change the labelling and weighting manually in the MOT authoring environment.

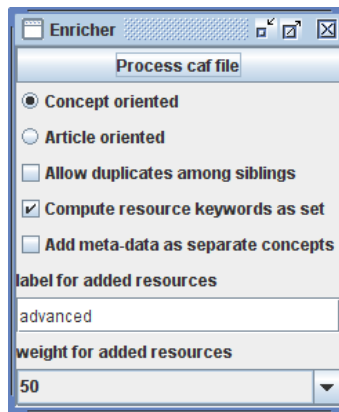


Figure 16 CAF loading; options and ranking method selection screen

After specifying the desired options for the matching mechanism between articles and domain concepts, the list of possible resources to add is displayed to the author, as depicted in Figure 17. This represents the second part of step 3 in the processing sequence. This step allows the author to overview the automatic matching process and hands him the ultimate decision about which papers to actually include in the final course. This allows for a checking stage, where unsuitable documents are filtered out (for instance, the exam paper with solutions on the topic of databases might appear as very suitable for the database course, but it is not appropriate to show to the students). For such reasons, fully automatic processing at this stage is inappropriate. By using semi-automatic processing, the teacher keeps overall control.

After the author has decided which articles to include, he saves them by pressing the 'Save caf & zip file' button Figure 17 and then deciding where on his computer to save the exported CAF XML file. These actions represent step 4 in the processing.

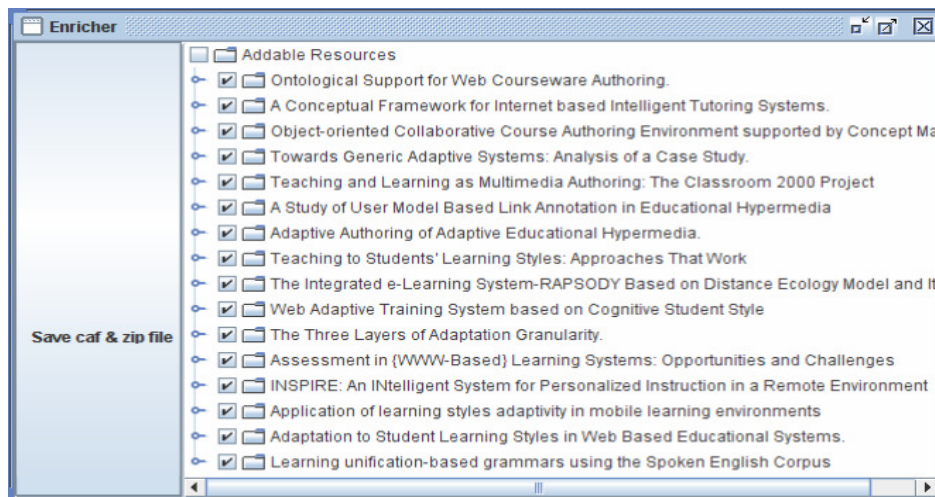


Figure 17 Resource selection screen

3.4 Evaluations

The (perceived) usability and usefulness of the first prototype was evaluated in three different steps, described in this section. First a small scale qualitative analysis was performed using a ‘*think aloud*’ method. Then two quantitative evaluations were performed. After that, a quantitative questionnaire was designed to answer questions about the perceived usefulness as well as usability of the tool. Finally the usability of the system was scored using a standard SUS (System Usability Scale, [19]), especially compared to authoring in MOT exclusively by hand.

3.4.1 Small Scale Qualitative Analysis

The initial evaluation of the conversion process and the Sesame2MOT Enricher v1 application took place in two steps: the first step was a small-scale qualitative experiment with 4 PhD candidates in Educational Technology attending the IMPDET course organised by the Joensuu University, Finland. This evaluation was based on the think aloud method. Below we list the conclusions we have drawn from the experiment and improvements we have made in our second prototype.

- The first prototype has no hints on what every step does and how users should make a choice. In the prototype, information buttons are added, where the authors can get more information about the choices they make, upon request.

- The first prototype appears to hang when a CAF file is selected. It is however processing. The second prototype features a progress bar to indicate that the application is still processing and has not crashed.
- The selection of articles is confusing. The second prototype features a new selection mechanism, which shows the location in the tree and offers the opportunity to change weights and labels for the articles that are to be added at an individual basis or at group level, using multiple select.
- The prototype should be integrated into MOT so that authors do not have to handle CAF files manually.

The second step was of a larger scale and contains therefore a larger amount of quantitative evaluation results, although qualitative information was also sought.

3.4.2 Qualitative Analyses

3.4.2.1 Evaluation Setup

This second and third evaluation steps were conducted at the Politehnica University of Bucharest in January of 2007. It took place within an intensive two-week course on ‘Adaptive Hypermedia and The Semantic Web’²⁶, which was delivered as an alternative track to the regular Intelligent Systems course. The students were fourth year undergraduates in Engineering studies and second year Masters students in Computer Science from the English-language stream. Firstly, basic knowledge on Adaptive Hypermedia and Semantic Web was addressed. The first course week was dedicated to theory and finished with a theoretical exam. Out of the initial 61 students, only the students with a satisfactory knowledge of the theory were selected to continue with the practical part of the alternative track. The rest returned to the mainstream course. The 33 students who passed the theory exam worked with the two versions of MOT (old versus new) and the Sesame2MOT (Enricher) conversion, the prototype constructed for the automatic authoring approach (see section 3.2 for more details). This selection process ensured that the

²⁶ Note that the lecture about the prototype and the theory behind it took about 1.5 hours. This training part was similar for all courses and interaction sessions used for the evaluation.

evaluations were performed by people who understood at least what they are aiming at. After these experiments, they were requested to submit a number of questionnaires, to answer both generic and specific issues regarding the automatic generation of adaptivity and personalisation via the methods outlined in section 3.2.

3.4.2.2 Hypotheses

We based our evaluation firstly on a number of generic, high level hypotheses:

- **v1 H 1.** The respondents enjoyed working as authors in the system.
- **v1 H 2.** The respondents understood the system.
- **v1 H 3.** The respondents considered that theory and practice match.
- **v1 H 4.** The respondents considered the general idea of Adaptive Authoring useful.

We refined these into more specific, lower granularity hypotheses (see Table 1), which ultimately generated our questions for the questionnaires. To explain the construction of the sub-hypotheses, take, for instance, hypothesis v1 H3. There, we check the matching between theory and the implementation. For the Sesame2MOT Enricher v1 application, from a theoretical point of view, we have defined different ranking methods and other options, such as allowing duplicate imported articles. These have been implemented as options for the user to select and therefore, in this particular case, matching theory and practice means firstly that these methods render different results, and secondly, that these different results should be just as the theory has predicted. Therefore, sub-hypothesis v1 H 3.4 and its own sub-hypotheses, v1 H 3.4.1, v1 H 3.4.2 and v1 H 3.4.3 emerged. As said, the hypotheses and sub-hypotheses feed into and determine the questions. The questions, unlike the hypotheses, were neutralised to the extent to which it was possible. Also, in additio to our specific questions, the SUS (System Usability Scale, [19]) questionnaire was used, which is a standard questionnaire especially aimed at removing bias, by alternating negative and positive questions [92].

3.4.2.3 Results and Discussion

In this section we present and discuss the results from the second evaluation step, which focussed on the Sesame2MOT Enricher v1 application. We will first briefly discuss the

quantitative feedback in general terms and then outline the numerical feedback obtained. We will show whether the hypotheses mentioned in section 5.2 have been confirmed.

The qualitative feedback showed that the stand-alone nature of the Sesame2MOT Enricher v1 application was quite confusing and an integrated version would be needed. Users considered the separate access to the different systems cumbersome and many installation steps and manual data interchange steps were considered time consuming and frustrating. It was also pointed out that, whilst working fine in isolation, when a whole group used the prototype at the same time, the connection between the Java application and the Sesame server was unreliable. We noted that the initial idea was that people would use the Sesame server containing their own Semantic Desktop on their own desktop. The issue was improved considerably with the integrated version, as the two servers (running the authoring system and the Enricher application respectively) were no longer far apart. A final problematic issue was the selection of appropriate weights and labels. Selecting these was found to be not very straightforward and the application did not allow for individual changes of weights and labels of articles. These problems were addressed in the next, improved implementation, where users can choose whether to use the recommended labels and weights with each strategy and then later on still have the possibility to change individual (or groups of) labels and weights. Moreover, the new prototype also features a strategy selection, which influences the weights and labels selected.

For testing our hypotheses we have used, where possible, numerical averages and tested their significance with the help of a Student's T-test. We assumed a confidence of 95%. Thus a hypothesis can be confirmed if the mean is $M > 0$ and the probability is $P < 0.05$. In order to obtain numerical averages, we mapped the multiple-choice answers follows: 'Yes' was mapped to 1, 'no' to -1 and 'mostly' to 0. Hence the average was always 0 and the T-test was applied by comparing against the neutral result of 0. Table 2 presents each hypothesis, the T-test results (T value, degrees of freedom (Df), Mean M, probability P) and whether the results show that the hypothesis was confirmed or not. The main hypotheses are shown in bold. Their result is obtained by combining the results of the sub-hypotheses.

Table 2 Sesame2MOT Conversion hypotheses results

Nr.	Hypotheses	T	Df	M	P	Confirmed ? (M>0; P<0.05)
v1 H1	The respondents enjoyed working as authors in the system	2.709	31	0.438	0.011	Not Confirmed
v1 H 2	The respondents understood the system.					Confirmed
v1 H 2.1	Concept oriented formula	4.458	31	0.625	0.000	Confirmed
v1 H 2.2	Article oriented formula	3.788	31	0.563	0.001	Confirmed
v1 H 2.3	Allowing duplicates / not	10.063	31	0.875	0.000	Confirmed
v1 H 2.4	Computing resources as set / not set	5.271	31	0.688	0.000	Confirmed
v1 H 2.5	Adding metadata as separate concepts	6.313	31	0.750	0.000	Confirmed
v1 H 3	The respondents considered that theory and practice match.					Confirmed
v1 H 3.1	The two ranking methods (concept-, article-oriented) do deliver different results.					Confirmed
v1 H 3.1.1	Concept Oriented delivers a different result from Article Oriented ²⁷	6.313	31	0.750	0.508	Not Confirmed

²⁷ Note that logic dictates that v1 H3.1.1 and v1 H3.1.2 should have the same result. They were put in different questions formulated slightly differently to check for bias.

Table 2 Continued: Sesame2MOT Conversion hypotheses results

Nr.	Hypotheses	T	Df	M	P	Confirmed ? (M>0; P<0.05)
v1 H 3.1.2	Article Oriented delivers a different result from Concept Oriented	6.313	31	0.750	0.508	Not Confirmed
v1 H 3.2	Ranking methods (concept-, article-oriented) <i>are in line with the theory</i> . In particular:		31			Confirmed
v1 H 3.2.1	Concept Oriented	2.252	31	0.375	0.032	Confirmed
v1 H 3.2.2	Article Oriented	2.709	31	0.438	0.011	Confirmed
v1 H 3.3	The different <i>options influence the result</i> ; in particular:		31			Confirmed
v1 H 3.3.1	Allowing duplicates	7.760	31	0.813	0.000	Confirmed
v1 H 3.3.2	Computing resources as set / not set	3.215	31	0.500	0.032	Confirmed
v1 H 3.3.3	Adding meta-data as separate concepts / not separate	6.313	31	0.750	0.000	Confirmed
v1 H 3.4	The results of the conversion <i>are in line with the theory</i> ; in particular:					Confirmed
v1 H 3.4.1	The two ranking methods	see 3.3.1.2	see 3.3 .1. 2	See 3.3.1.2	see 3.3.1.2	Confirmed (see 5.3.1.2)
v1 H 3.4.2	Allowing duplicates / not	7.760	31	0.813	0.000	Confirmed

Table 2 Continued: Sesame2MOT Conversion hypotheses results

Nr.	Hypotheses	T	Df	M	P	Confirmed ? (M>0; P<0.05)
v1 H 3.4.3	Computing resources as set/ not	2.252	31	0.375	0.032	Not Confirmed
v1 H 3.4.4	Adding metadata as separate concepts / not separate	4.458	31	0.625	0.000	Confirmed
v1 H 4	General idea useful	15.000	31	0.938	0.000	Confirmed

As we have seen, most hypotheses have been confirmed based on the data. The data about Article vs. Concept oriented ranking, the computation of keywords as a set and the students' enjoyment was not statistically significant although it showed a supportive mean. The Sesame2MOT conversion is indeed considered useful and in line with the theory. Its options are mostly understood. Respondents agreed strongly with most of our hypotheses, with all means above zero. Looking at the ones with lower scores, such as concept-oriented and article-oriented method, as well as computation of resource as set, they were less sure in their statements. This is probably due to the fact that they did not use these options enough. This shows that more targeted evaluations may be necessary to establish without a doubt the acceptance rate of these features.

3.4.2.4 System Usability Scale (SUS)

With an SUS score the usability of systems can be compared. The average score can be contrasted and visual graphs can be constructed to identify specific problem points. The questions (which are alternately positive and negative) are plotted on a circle using a scale from 1 (strongly disagree) to 5 (strongly agree), with 1 in the centre and 5 at the border. When the results for the questions are placed on the scale, the ideal system should show a perfect star shape, as positive and negative questions alternate. In Figure 18, the SUS scores for the different systems are shown in such a SUS graph. The figure shows that the systems have relatively similar scores. Visible differences are that Sesame2MOT seems to have a

higher perceived learning threshold, whereas the old MOT is considered more inconsistent and more cumbersome.

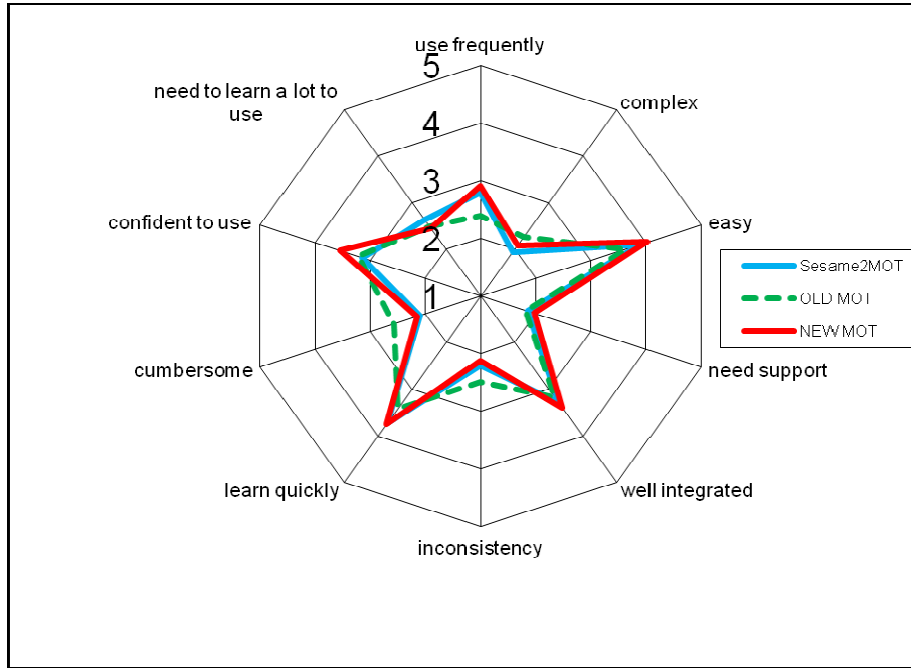


Figure 18 SUS score for the three systems

Normalised responses range from 0 to 4, see [19]. Thus we applied a T-test comparing the normalised results against the average neutral value of 2. A paired T-test was used, since we compared answers of the same sample (group of students). Moreover, the main hypotheses were further divided into sub-hypotheses.

3.4.2.5 Results

Below we list the main hypotheses and comment on how much they are supported.

- *The respondents' overall preference, from a usability perspective, is as follows, in increasing order: old MOT, new MOT, Sesame2MOT.*

The results on learning preferences and the preference for Sesame2MOT over the new MOT (difference -0.07 (>0 exp.); $p=0.18<0.05$; $t=-1.44$) were not significant.

The hypothesis cannot be supported. Preference for the new over the old MOT (diff 0.26 (>0 expected); $p=0.00<0.05$; $t=4.16$) was confirmed.

- *The new MOT is more usable, hence we should base further developments on this version of MOT.*

For all different parts, as well as overall SUS score (see hypothesis 3), the new version of MOT is preferred over the old version. Thus we should indeed focus further development on the new version. The hypothesis is supported.

None of the hypotheses related to learning threshold showed any significant difference between the three systems. This is possibly due to the fact that systems respondents had to learn all the theory before working with the three systems, or that both MOTs are very similar from a theoretical point of view.

We computed the correlation between the SUS scores for the 3 different systems. This showed that the respondents' answers to all three systems' SUS questionnaires are significantly correlated. This seems to be due to one of the following two reasons:

1. respondents were not quite aware for which systems they were filling in the SUS questionnaire (this suspicion is based on some questions from students), or
2. the students perceived the three systems as variants or parts of the same system.

Moreover, we also found that the correlation between the scores for the new MOT and for the Sesame2MOT conversion is highest. This could indicate that a substantial number of respondents viewed the Sesame2MOT conversion and the new MOT as one system, since Sesame2MOT is currently integrated into the new MOT.

From a usability point of view, we have seen that respondents liked working with the new MOT and the Sesame2MOT conversion, but were less positive about the old MOT.

A highlighted drawback is that Sesame2MOT has a steeper learning curve. We also see that integration with the MOT systems could be improved. From the quantitative feedback gained we have seen that the latter is one of the main causes of the steeper learning curve. Therefore we have further integrated the conversion into the new MOT system for the second prototype, described in the next section.

3.5 Second Prototype

The new, improved prototype is web-based. It is integrated with the 'My Online Teacher' system. Therefore, the author does not need to deal with CAF files anymore. CAF files are

still used behind the scenes. The author accesses the new prototype by clicking the enrich link in the Goal Model view of MOT as shown in Figure 19.



Figure 19 New prototype is accessed via enrich link in Goal Model view

This link will take the author to a screen, shown in Figure 21, where he can select the ranking method and the other options as mentioned in section 3, similar to the old prototype, however the screen does feature a new option of strategy selection. Users can choose to select a strategy and let the weights and labels be recommended based on their choice of strategy. The CAF file does not need to be loaded. As the web-based system knows which Goal Map the author accessed previously, and has access to the CAF convertor, it can generate the CAF file without intervention from the user.

The next step features an improved version of the resource selection screen (Figure 20). The checkboxes allow for deciding which of the recommended resources are to be added. On the right hand side there is a small outline of the Goal Map hierarchy. If the author clicks on one of the titles of resources it will highlight (in yellow) the place in the hierarchy where the resource will be added when the checkbox is checked. In this way, more visual information and a sense and feel of the structure and placement of new resources is given to the authors, as compared to the initial prototype. Moreover, the form at the bottom allows the author to change weights and labels for individual resources (for all resources currently highlighted). The resource selection screen is shown in Figure 20.

Finally, after the author has selected the resources, the process as described in Figure 12 is completed. As the web-based Sesame2MOT Enricher v2 application has direct access to the import functionality for CAF files in MOT, there is no need for the author to manually upload this CAF file. The application does however retain the option for the author to download the modified CAF file. This can be useful, for example, for external use or backup purposes.

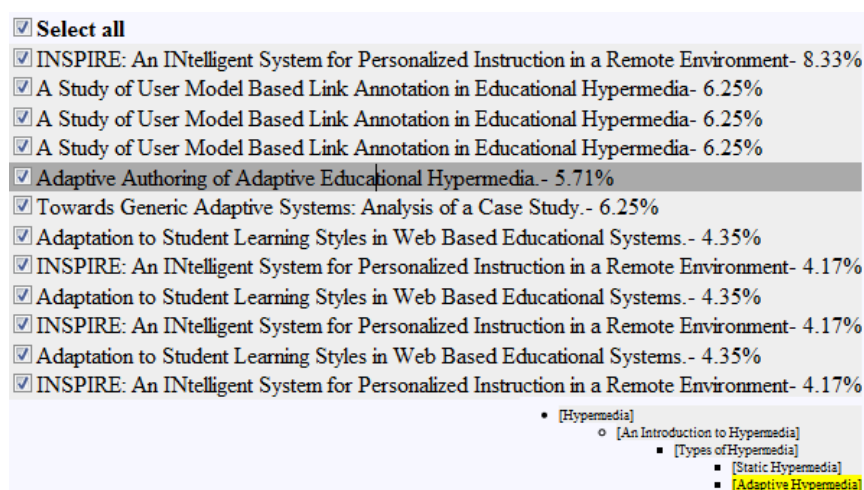


Figure 20 Resource selection

Enrichment of existing Goal Map:

User: prolearn Lesson: Adaptive Hypermedia & Learning Styles

Ranking method: ☒ Concept Oriented ☐ Article Oriented

Allow duplicates among siblings: ☒

Compute resource keywords as set: ☐

Add meta-data as separate concepts: ☒

Strategies:

show all
beginner-intermediate-advanced

Get keywords from MOT attributes:

title
pattern
text
explanation
exercise

Use default label and weight: ☐

Label:

Weight:

Use default Sesame server (for ☒ experiments):

Sesame URI:

Sesame username:

Sesame password:

Sesame Database:

Remember Sesame settings (using ☒ cookies):

Figure 21 Options and ranking method selecting screen

The deployment screen allows the author to deploy the new enriched Goal Map. The option of deploying to MOT is similar to importing the modified CAF file in the previous prototype [46], but requires less effort from the author. There is also an additional option to deploy to AHA!. This offers a way to directly convert the created course into a lesson in an AHA! System. For this, the application requires some additional information on where to find AHA!. The deployment screen implementing the described step is shown in Figure 22.

User: prolearn

[Download CAF file](#)

☒ Deploy to MOT

Enter the lesson name

Naming of the new GoalMap and its Concept Maps

☒ Leave them as they are

☐ Rename lesson/concept if the name exists

☐ Merge lesson/concept if the name exists

☐ Deploy to AHA

AHA! URI:

AHA! author name:

AHA! author password:

AHA! course name:

Figure 22 Deployment screen

3.6 Evaluation of second prototype

3.6.1 Evaluation Setup

The third and final evaluation was again conducted with another batch of students from the ‘Politehnica’ University of Bucharest, and focused on the new web-based Sesame2MOT Enricher v2 application. It was conducted in May 2008 with a group of thirty students studying in a “Web Programming” course²⁸, partially delivered via distance learning. Students collaborated in the creation of new content in MOT and answered our questionnaire. The students were enrolled in the third year of Computer Science. Note that the web-based prototype was particularly suitable because the course was about web programming and was partially delivered online. The course covered, among other things, the basics of Adaptive Hypermedia and the theory behind our system setup. Directly after the lecture on our system setup students were requested to complete a questionnaire.

3.6.2 Hypotheses

We again based our final evaluation on a number of generic, high level hypotheses, follows:

- **v2 H 1.** The students enjoyed working as authors in the system.
- **v2 H 2.** The students (think they) understand the new web-based Sesame2MOT Enricher v2 function.

²⁸ Note the lecture on the prototype and the theory behind it took about 1.5 hours. This part was similar for both courses used for the evaluation.

- **v2 H 3.** Students think the interface of the web-based Sesame2MOT Enricher v2 application is clear.
- **v2 H 4.** Students perceive the web-based Sesame2MOT Enricher v2 application as being fast enough²⁹.
- **v2 H 5.** Students notice that the two ranking methods deliver different results.
- **v2 H 6.** Students notice that the results of the respective ranking methods are in line with the theory (so did what the students expected).
- **v2 H 7.** Students understand the use of and theory behind the ranking methods.
- **v2 H 8.** Students understand the option 'Compute resource keywords as set'.
- **v2 H 9.** Students notice that the option 'Compute resource keywords as set' influences the results.

3.6.3 Results and Discussion

In this section we present and discuss the numerical results from one of our questionnaires, which focussed on the web-based Sesame2MOT Enricher v2 application. For testing our hypotheses, we have again used, where possible, numerical averages, and tested their significances with the help of Student's T-test. This time we had 3 possible answers for each question - 'yes', 'no' and 'no answer'. We assumed 'no answer' to be neutral and mapped the answers to 1, -1 and 0 respectively. Note that this means the average is 0 and tests were therefore done against the average (neutral) value of 0. Assuming a confidence of 95%, a hypothesis can be confirmed, if the mean is $M > 0$ and the probability is $P < 0.05$. A table with each hypothesis, T-test results (T value, degrees of freedom Df, Mean M, probability P) and whether the results show that it was confirmed, is displayed in Table 3.

²⁹ Note that 'fast enough' reflects here the perceived performance for the user, to be distinguished from the actual, measurable performance.

Table 3 Final evaluation results, web-based Sesame2MOT Enricher v2 application

Nr.	Hypotheses	T	Df	M	P	Confirmed (M>0; P<0.05)
V2 H 1	The students enjoyed working as authors in the system.	4.219	22	0.52174	0.000	Confirmed
V2 H 2	The students (think they) understand the new web-based Sesame2MOT Enricher v2 application.	3.219	22	0.39130	0.004	Confirmed
V2 H 3	Students feel the interface of the web-based Sesame2MOT Enricher v2 application is clear.	2.152	22	0.34783	0.043	<i>Not Confirmed</i>
V2 H 4	Students think the web-based Sesame2MOT Enricher v2 application is fast enough.	0.569	22	0.08696	0.575	<i>Not Confirmed</i>
V2 H 5	The students notice that the two ranking methods do deliver different results.	5.348	22	0.56522	0.000	Confirmed
V2 H 6	The students notice that the results of the respective ranking methods are in line with the theory (so did what the students expected).	4.899	22	0.52174	0.000	Confirmed
V2 H 7	The students understand the use and theory behind the ranking methods.	2.598	22	0.39130	0.016	<i>Not Confirmed</i>
V2 H 8	The students understand the option 'Compute resource keywords as set'.	3.425	22	0.52174	0.002	Confirmed

Table 3 Continued: Final evaluation results, web-based Sesame2MOT Enricher v2 application

Nr.	Hypotheses	T	Df	M	P	Confirmed (M>0; P<0.05)
V2 H 9	The students notice that the option 'Compute resource keywords as set' influences the results.	2.237	22	0.39130	0.036	<i>Not Confirmed</i>

We have seen that students enjoyed working with the new web-based Sesame2MOT Enricher v2.0 application and after our explanations and working with it, they understand it. Whether the interface is clear and whether the application is fast enough could not be confirmed. The performance bottleneck was found to be the connection to the Semantic Desktop setup.

The students noticed the difference between the ranking methods and found them to be in line with the theory, but we could not establish whether they understood the actual use of the different ranking methods (although P is low so it could be acceptable for different confidence requirements). On the other hand, students understood the option 'Compute resource keywords as set' but we could not establish if they noted the difference. This may also be due to the fact that the examples they were told to use did not have many instances of multiple keywords occurrence in MOT.

We have furthermore experienced, whilst demonstrating the application to PhD students, that without the basic theoretical knowledge, the interface seems a little complicated. Balancing a high level of functionality and a high level of usability are often contradictory goals, and compromises have to be accepted. In the second prototype this was partially addressed by offering standard settings for most options. Also, a list of strategies was introduced, where the weights and labels would depend on the chosen strategy (unless the author chooses to change them). However, for the future, we recommend putting these options in a settings screen so that they do not confuse beginning authors.

Also note that the populations used in the experiments were very different, both in terms of age and experience (PhD versus Master students and third year students) as well as in terms of application areas (Educational Technology versus Engineering and Computer Science). In this way, we could obtain information about the tool usage from various potential users.

3.7 Related Research

Authoring of adaptive hypermedia is notoriously difficult work [21]. Research on improving the process of adaptive hypermedia authoring ranges from ontology-based authoring [110], to integrating standards and their representations [77], [17], using data mining techniques [147], web services [113], interfacing techniques between authoring systems [48] and adaptation languages [47]. Moreover, for all the directions above, there are developments towards graphical authoring tools [84], [68], to aid the complex task of authoring (pedagogical) adaptation strategies.

Our work also relates to efforts regarding searching and ranking within RDF metadata [31]. The difference is that we are comparing resources, from the Semantic Desktop setup, in RDF metadata, with resources in the authoring environment which are not currently stored in a semantic way. Dolog et al. [59] propose a complete semantic-based approach, in which the user model would be stored in RDF and standards like LOM [98] (Learning Object Metadata) would be used for describing the learning resources.

Our work has provided a new approach to exploit a Semantic Desktop [11], [135] for aiding Adaptive Hypermedia authors. Though the approach provides content, it also helps in the authoring of adaptation, as labels and weights can be provided for the retrieved resources to offer them as content-alternatives, according to a strategy chosen by the author.

3.8 Conclusions

In this chapter we have briefly reviewed an authoring environment for personalised courses, as well as a Sesame2MOT Enricher mechanism and prototype based on Semantic Desktop

technology. The chapter describes the theoretical considerations for the implementation that uses data and relations stored by a Semantic Desktop environment in order to (semi)-automatically enrich adaptive courses, and the evaluation of these considerations as well as of the prototype. From the three evaluation steps performed so far the promising result is that the theoretical concept of Adaptive Authoring of Adaptive Hypermedia was perceived as useful. However, the actual implementation has room for improvements. We have gained some important feedback into possible improvements to the Sesame2MOT Enricher application itself, as described in section 6. Respondents in our experiments pointed out that the integration is currently not optimal and the user interface needs enhancements. Based upon some initial evaluations we developed a web-based prototype and ran a third series of evaluations. These evaluations showed that students enjoyed working with the new web-based Sesame2MOT Enricher v2.0 application, they mostly understood it and, importantly, they noticed the difference between ranking methods and confirmed that the ranking is in line with the theory. We did not get a conclusive answer on the interface and performance. We feel that the interface may need careful explanations and the performance could be improved, especially in terms of the connection to Sesame.

Concluding, we claim that this research has shown new ways in which the complex problem of authoring for personalisation and adaptivity can be approached, by exploiting existing resources and introducing (semi)-automatic steps in the authoring process. In such a way, adaptation is performed at the early stage of authoring and is not just an end product of authoring. This research has shown the promises of semi-automatic authoring. Other means can be exploited to help authors in their tasks and adapt to their needs. In previous research [42], it has been shown that authoring adaptation can be performed by automatically adding relatedness relations between concepts in a domain map. This current research is bolder, in that it allows for adding both content and relations. We hope we have thus paved the way for further research in this exciting area of Adaptive Authoring of Adaptive Hypermedia.

As a further line of research we recommend concentrating on automatic methods of providing personalisation, such as data mining techniques. Also, combining Semantic Web techniques with Social Web and Web 2.0 techniques should be investigated as a means to allow students to be both learners and authors (contributors) to their learning environment. For this purpose, it is useful to estimate how all these types of users can deal with the system.

4 Adaptation Languages and Standards, a Comparative Analysis between IMS-Learning Design and Adaptive Educational Hypermedia

Adaptation and personalisation are gradually 'seeping' into various aspects of our daily life via the devices that we use such as the Web, the phone and the Palmtop. Personalisation was added to the Web as an afterthought to solve various legacy problems (such as the 'one-size-fits-all' web and issues of personal information security).

Adaptive Hypermedia [21] research offers solutions to these problems, although other ad-hoc ones appear everywhere - in the commercial domain [151] as well as in the educational one. It is therefore vital to foresee upcoming trends, to prepare for them in advance, and to apply the lessons of the past, especially in new developments.

IMS Learning Design [100] is an e-learning specification, with strong pedagogical roots and a de facto emerging standard, which models pedagogical scenarios, aiming at covering various learning situations and roles. Its forefathers are design methodology and pedagogy. IMS-LD is quickly gaining popularity and is touted as being the future of all educational web-material and interaction.

Adaptive Educational Hypermedia (AEH) [22] adds personalisation and adaptation to e-learning. It caters for each learner individually, for example, adapting to his knowledge, needs, preferences or learning styles, conforming to learner centred education. It is based on hypermedia (any collection of nodes and links), adaptivity (its main strength, based primarily on user models) and finally, distance learning.

As the goals of both AEH and IMS-LD are the same [39], to create the best possible learning experience for the learner, the objective of this chapter is to investigate howfar the

two currently align; to what extent AEH can be expressed in IMS-LD and vice versa. Where this is not currently possible we identify missing features in both AEH and IMS-LD. The chapter is mainly my own work. However the observation that IMS-LD and AEH have common goals and there may be lessons that the communities can learn from each other was made previously in [39]. The authors of this paper have acted as mentors and the work has been published in [85].

In the rest of this chapter, we first introduce the representation and semantics of both AEH and IMS-LD in section 4.1. Then, in section 4.2 we introduce the criteria according to which we will compare AEH and IMS-LD. In section 4.3 we compare adaptation features of AEH to those of IMS-LD, first according to Brusilovsky's taxonomy [21] for AEH, and then according to adaptation paradigms associated with IMS-LD. Section 4.4 summarises the comparison results according to our criteria. Section 4.5 puts our comparison in perspective with related research. Finally in section 4.6 we see what conclusions we can draw with regard to if and how AEH and IMS-LD can benefit from each other.

4.1 Representation and Semantics

In this section we focus on the representation of learning material and semantics in AEH and IMS-LD. In AEH, the LAOS framework [41], described in section 2.4.8 is one of the most comprehensive and commonly used frameworks for representing adaptive content for authoring. LAOS extends previous frameworks such as AHAM [56] in the sense that it introduces an even higher separation of concerns for adaptive hypermedia authoring. Thus, we can claim that comparisons made between IMS-LD and LAOS highlight differences between the IMS-LD and the AEH view in general. Therefore, we can use LAOS for the purpose of this comparison. In the remainder of this section we shall sketch LAOS and IMS-LD and comment on the main similarities and differences.

4.2 Comparison Purpose and Criteria

4.2.1 IMS-Learning Design

IMS-Learning Design [96] is an e-learning specification for pedagogical scenarios. There are various players [1], [148], [117], authoring tools [146] and engines [126] for IMS-LD. It has three implementation levels of incremental expressiveness (A, B and C). They allow the modelling of Units of Learning (UoL). They describe collaboration, adaptation, adaptability or any other pedagogical method. The division into levels is as follows. Level A provides methods, activities and roles. Level B adds properties, conditions, calculations, global elements and a monitoring service. Level C rounds it off with notifications [105], [142]. Units of Learning (UoLs) are made up of a manifest, describing the adaptation, a set of resources and optional external XHTML files that can improve a few features, for example, a specific usage of properties or services. Each UoL is written taking a structured metaphor that defines runs, plays, acts and activities. IMS-LD provides a full XML representation model from a lesson plan to a final UoL running online. Its lifecycle is defined in three isolated steps: modelling, publishing and playing. This implies that design-time and run-time are separated. When any modification is required, the author has to re-start designing, publish the new package and play it afterward. This is similar to the authoring and delivery process in AEH. The learning roles are sharply defined and can be assigned to different people. End-users of a IMS-LD based system can be students, tutors or authors. Hence, IMS-LD is a flexible way to represent, edit and execute a variety of pedagogical models. Furthermore, some features in levels B and C allow several types of adaptation. The appropriate use of conditions, global elements, calculations, monitoring services, properties and notifications allows personalised UoL based on flow, content, interface and evaluation [27].

The *manifest.xml* file below defines a similar example to the one previously described for the CAF and LAG representation in sections 2.5.1 and 2.6.1: it first hides the answers and, only after the questions they belong to have been accessed, it shows them.

```
...<properties>
```

```

    <locpers-property identifier="Question1">
        <datatype datatype="boolean"/>
        <initial-value>0</initial-value>
    </locpers-property>
</properties>
<!-- If question1 is accessed show answer -->
<if>
    <is>
        <property-ref ref="Question1"/>
        <property-value>1</property-value>
    </is>
</if>
<then>
    <show>
        <class class="Answer1" />
    </show>
</then>...

```

Figure 23 A manifest example of a UOL

4.2.2 Comparison of high-level features of LAOS and IMS-LD

4.2.2.1 General Content Representation:

- **Comparison:** From the point of view of representation and semantics, especially, for general representation, IMS-LD prescribes an XML format to describe properties and global XHTML files that use these properties for its content. Although AEH also allows XML representations (like CAF), the clear difference is that there is currently not one standardised way of describing the content, and different systems may use different ways of representing the same content. The use of XML and the introduction of a standard make IMS-LD more portable and allow a high level of reuse. This is desirable for any authoring system for e-learning and especially for authoring of adaptive material, which is notoriously complex and time consuming. For such authoring, the ‘write-once, use many’ paradigm is vital.
- **Conclusion:** The field of AEH would, we believe, benefit from a clearly defined and well thought-through standard and a unification of approaches. Such an approach is currently sought after in the new EU FP7 GRAPPLE project.

4.2.2.2 Content Extent:

- Comparison: An IMS-LD manifest is significantly more verbose than the combination of CAF and LAG. A reason for this is that IMS-LD manifests need to specify much more information, which may or may not be relevant to the current application. This allows various enriched functionalities, but costs in readability and space. Such enhanced functionality is illustrated by the following scenario. After authoring in AEH, the author does not interact anymore with either the students or the content. This is mainly because the focus has been on creating automatic adaptation and not on providing a software tool for teachers to communicate in real time with their students. In IMS-LD, provisions for such communication are, however, present. One way in which IMS-LD allows this communication is using specific definitions of various roles, specified via the manifest.
- Conclusion: This shows that IMS-LD and AEH have some complementarities. In terms of educational value of the experience, both IMS-LD, with focus on people and their roles in the learning process, as well as personalisation to the learner's needs, as supported by AEH, are necessary. It is conceivable that the two approaches could co-exist together in learning systems. Such issues are explored within the GRAPPLE project in work packages on authoring and on IMS-LD applicability.

4.2.2.3 Generic Conceptual point of view:

- Comparison: Both AEH and IMS-LD use a multi-layered method for describing the content and adaptation. In AEH, this is done via the authoring model layers, and in IMS-LD via the different levels. However, the levels in AEH represent a clear separation between content, grouping of content and adaptation, whereas in IMS-LD, the division is based on certain functionality features. Past experiments [40] show that a clear separation of the adaptation from the content (such as in LAG) is very beneficial, as it allows reuse of advanced adaptation strategies

created by programmers, for people with little or no programming knowledge (for example teachers).

- Conclusion: IMS-LD should allow for a clear separation of adaptation from content, as is supported by (some of) the adaptive hypermedia frameworks and methodologies. This would allow a much more flexible approach to reuse of the authored products.

4.2.3 Comparison of low-level Features of LAOS and IMS-LD

Static content representation:

- Comparison: Looking deeper into the data representation and semantics, we note the following. ‘Static’ content (called ‘domain content’ in AEH) is represented by IMS-LD as XHTML documents, tagged as resources and stored as separate files (see Figure 23). In AEH, such content can be represented in various ways (there is no standard). CAF uses the domain concept hierarchy representation, storing all data for a lesson in one file (see Figure 23). However, the AHA! adaptation engine [55], for instance, interprets CAF data and divides it between several XHTML documents, in a fashion not that different from the IMS-LD representation.
- Conclusion: XHTML representation of static content seems to be the best way to deal with the atomic, indivisible pieces of static information, that build the building blocks of an e-learning system and that can be reused in various sequences and configurations to allow for personalisation to the learner. In our examples we describe a CAF file with a set of questions and their answers, showing the latter only after learners have seen the questions. The way questions and answers are described is not very different in CAF or IMS-LD, yet the way the adaptation is described differs greatly. For IMS-LD, rules are described via a hierarchical XML structure. In LAG, rules are defined via a dedicated programming language. Most AEH represent adaptation at the level of adaptation assembly language (as in the LAG framework) and formats vary: rules can be en-

capsulated into concepts (as in AHA! [55]), or kept separately, in XSLT sheets (as in WHURLE [117]).

Another great difference between LAOS AEH and IMS-LD lies in the issue of reusability. AEH LAG strategies can be reused to adapt different content, as long as they are written in general terms rather than for specific concepts. In IMS-LD, however, the adaptation rules are embedded within the content; this makes reuse virtually impossible. Note that the *manifest.xml* files could be reused, but this only defines a set of properties to which to adapt, it does not specify how to adapt to these properties.

4.3 Adaptation

Both AEH and IMS-LD aim to provide a better learning experience tailored towards the end-users needs (teachers, learners and administration staff). Thus, both provide possibilities for personalisation and, to some extent, adaptation. In this section we investigate the differences and similarities between AEH and IMS-LD with respect to how adaptation is achieved.

4.3.1 Adaptation Engineering Taxonomy

Brusilovsky's Taxonomy [21] describes almost exhaustively, from a technical, adaptation engineering point of view, the types of adaptation encountered in Adaptive Hypermedia. The two main adaptation types are Adaptive Presentation and Adaptive Navigation Support. These are divided into sub-classes which we investigate in turn. Moreover, AH defines the concepts of adaptivity as system-driven adaptation (for example, adaptation that is induced by a rule-base reasoning system connected to the user interface) and adaptability, as user-driven changes and adaptation (for example, adaptation that is performed via interfaces allowing users different levels of options and changes in parameters, such as the setting of preferences). All types of adaptation in Brusilovsky's taxonomy can be performed via adaptivity and adaptability, in general. However, AH systems usually aim for a low cognitive overhead for the user, thus striving towards 'pure' adaptivity. Next, we

examine to what extent the classes of Brusilovsky's taxonomy can be represented in CAF and LAG as well as IMS-LD.

4.3.1.1 Adaptive Presentation

Adaptive multimedia presentation

Multimedia adaptivity can only be achieved in current AEH systems at the level of access: different media, such as text, video or others, can be accessed, based on rules and guided by the user model. Such an example is the strategy that caters for visual versus verbal students, written in LAG, 'vis/verb' [1]. Adaptive multimedia presentations are also possible in IMS-LD. In, for example, the 'e-adventure' project, an adaptive game's story board is specified with the use of IMS-LD [116]. It is then possible to tune the game according to the learner's profile.

Adaptive text presentation

IMS-LD does not yet provide specific methods for natural language adaptation, nor do most current AEH systems, in spite of some early research in this area.

Canned text adaptation by means of inserting, removing, altering, sorting, dimming fragments or stretching text, is currently the most used type of adaptation in AH systems. A simple example is the Q&A strategy shown in Figure 24. Canned text adaptation (showing and hiding fragments of text) can be done in IMS-LD as follows. In the XHTML resources of IMS-LD the authors can define their own pieces of text inside DIV layers (division layers). DIV elements in IMS-LD are just used as placeholders to define custom tags. Authors can thus add conditions, for example, if a certain element should be shown or not. However, this method is not inherent to IMS-LD, but to (X)HTML in general and it is extremely low-level (adaptation assembly language in LAG) and time consuming.

Adaptation of modality

Adaptation of modality, such as providing multiple language alternatives, or providing different media (videos, text, audio) could be implemented in IMS-LD, by adding all

modalities to the resource files and then selecting which one to show upon certain conditions. Such adaptation is typical, for instance, for browsers, so for AEH it is less of a mainstream research direction. In CAF and LAG, such adaptation of modality can be simulated in a similar way to canned text adaptation (for different languages) or adaptive multimedia presentation (for different version of media).

4.3.1.2 Adaptive Navigation Support

Direct guidance

In AEH, direct guidance means often providing adaptive navigation items such as menus, and a ‘Next’ Button, only offering the user the possibility to navigate to a suitable link. This can be added via the LAG language with the statement:

```
PM.GM.Next = true
```

In IMS-LD, direct guidance can be achieved by using DIV layers in the XHTML resources. In effect, it can then emulate menu-adaptation.

Adaptive link sorting

Adaptive link sorting can, in theory, be done in the same way as hiding and showing of text. In IMS-LD, different DIV layers could be defined with the correct sorting and the right one showing at the right moment. However, links cannot actually appear in different orders, just at different learning stages. In LAG, a simple version of sorting is achieved by using the ‘To Do’ list, that specifies which links to access next:

```
PM.GM.ToDo = true
```

Adaptive link hiding

Adaptive link hiding can be performed in the same fashion as the hiding and showing of text. In IMS-LD this can be implemented, since hiding and showing can be performed over resources. In LAG, the following expression will hide not only the respective concept, but also the link (for example, in the ‘To Do’ list) to that concept:

```
PM.GM.Concept.show = false
```

Adaptive link annotation

Again, in IMS-LD, using DIV layers in the XHTML resources makes it possible to obtain link annotations. In the conversion to AHA! [55] of LAG, links to concepts in the menu illustrate adaptive link annotations, via bullets coloured in the traditional (pseudo-)traffic-light colours (green, for ready to visit; red, for not appropriate yet; white, for already seen).

Adaptive link generation

Adaptive link generation means providing links on the fly where no links were designed before. This can be, for instance, from a database of links, or from the open web. In AEH (thus in CAF and LAG), links to concepts are provided on the fly when the concepts are appropriate for the learner, so from this point of view, they are adaptively generated links. For IMS-LD, adaptive link generation would require the ability to change the XHTML resources at run-time. IMS-LD allows some limited changes on the fly, via a UoL run-time. As long as the possibility has previously been provided for at design-time, (permanent) links can be added at run-time, by either a teacher or a student. This represents adaptable³⁰ (but not adaptive) link generation.

Map adaptation

Adaptive maps can be achieved in a number of ways. For IMS-LD, the most labour-intensive approach for the author would be to create a picture for every possible version of the map. A smarter method would be to have the map drawn by DIV elements defined in some style sheet. However, this is too demanding for current authoring tools. For AEH there is some new research in this direction [22], but most current systems do not provide it.

³⁰ thus, from an epistemology point of view, an analytic, a priori solution, instead of a (more interesting) synthetic, a posteriori one;

4.3.2 Pedagogy-based Adaptation Classification

In a literature study that analyses adaptation from the point of view of its pedagogical goal, instead of a technical viewpoint, eight different kinds of adaptation in e-learning systems were defined [28]: Interface based, Learning flow based [128], Content based, Interactive problem solving support [114], Adaptive information filtering, Adaptive user grouping, Adaptive evaluation and Changes on the fly [138]. This classification was previously used [27], [28] to describe the expression power in terms of adaptation for IMS-LD. Therefore, after analysing how IMS-LD compares in terms of AH and AEH adaptation, we now investigate how AEH systems in general and CAF and LAG in particular, can cope with the demands of this classification. The results are shown side-by-side with the IMS-LD results. For the sake of knowledge exchange between these domains, we also map this new classification over Brusilovsky's taxonomy.

4.3.2.1 Interface Based Adaptation

Interface adaptation is based on menu options, navigation facilities and visualisation facilities. In such general terms, this type of adaptation would reflect on basically all aspects of Brusilovsky's taxonomy. More specifically, it can reflect on adaptation performed on such interface aspects as menus (items), display options, size of windows and fonts. Thus, this is the most commonly encountered type of adaptation in AEH. Examples can be found at [1], for example, the 'Beginner/Intermediate/Advanced' strategy uses adaptive navigation support to colour recommended links green and non-recommended ones red, guiding learners through these three learning stages (from beginner to advanced).

For IMS-LD, this issue relates to the user interface provided with IMS-LD players such as the CopperCore player [148], the Reload Player [13] and Sled [142]. The current generation of these tools does not provide facilities to allow interface adaptation at run-time, although Sled can be customised during the set-up using style sheets. Current IMS-LD players cannot change the size and position of their panels or working areas, the definition of their windows (adaptation of modality) or any other navigation facility (adaptive navigation support). These players cannot change basic features, like font-size, font-colour, font-type

or alignment, either. However, limited adaptation is possible inside the Unit of Learning, if we use two resources: DIV and environments. DIV layers can be shown and hidden at run-time by any of the main participants in the learning process (student, teacher and a set of rules). Inside a DIV layer we can define the different options and/or the look and feel of the same content, meaning a de facto interface-based adaptation. Furthermore, environments can provide different set-ups (contents, approaches and views) related to the same Unit of Learning, leading to a final personalised interface. Although neither of these two solutions (DIV layers and environments) is based on the external wrapper/player, they can provide a simulation of interface adaptation.

4.3.2.2 Learning Flow Based Adaptation

Generally speaking, adapting the learning flow means creating different sequences of learning events for the different learners. This is actually fundamental in AEH, where different paths are generated based on user models. A learning flow example for AEH, where a learner can choose between a visual, verbal or neutral presentation, is the ‘vis/verb’ strategy [1].

For IMS-LD, describing an adaptive learning flow uses four (out of the five) Level elements [26]: properties, calculations, global elements and conditions. In addition, monitoring services can be added to track the students’ behaviour and allow the teacher to adapt the flow dynamically³¹, for example, ‘Learning to Listen to Jazz’ (all examples are at [109]). A student can learn about four different Jazz styles in a sequential way and he can choose between a thematic itinerary and a historical itinerary, following different milestones in the course. This would correspond to adaptable navigation support in AEH (not system-driven adaptation, but user-driven adaptability and control). An additional example is ‘GeoQuiz 3’ where the activities are defined by the performance of a student after answering an evaluation form. Depending on the final score and the related level acquired, one or another activity is shown. Here, actual adaptivity (system-driven

³¹ Note that this is different from AEH adaptation, which is predefined and cannot be interfered with during runtime.

adaptation), in the form of canned-text AEH adaptation is portrayed. Thus, the description of an adaptive learning flow is mainly based on four IMS-LD elements at Level B: properties, calculations, global elements and conditions. All these elements exist in Adaptive Hypermedia as well.

4.3.2.3 Content and Activity Based Adaptation

Content Based Adaptation, as in personalisation based on domain content, is also fundamental to AEH systems and can be performed via adaptability, but mostly, it is performed via adaptivity (system-control). For example, the ‘Roll-out’ LAG strategy in [1] allows for different content to be displayed to the same learner when revisiting a page, thus taking into account his new experience and learning state. If content is regarded as activity content, such content adaptation can be achieved in AEH by switching between the showing and hiding of several linked environments, for example, showing or hiding an HTML page containing a frame with the linked environment (for example, YouTube videos, or websites). AEH can also support adaptive activities [123], [124], [23] by adaptive selection of both activities and services.

In IMS-LD, the content of an activity needs a resource linked to the element ‘Activity Description’. Although this link cannot be changed at run-time, three other elements can be modified dynamically: 1) the content inside an XHTML resource, defining classes and DIV layers that can be hidden and shown based on certain parameters; 2) the content of pre-defined properties/variables, that can be replaced with other content typed-in on the fly by the learner; 3) the content of an activity can be adapted by switching, showing or hiding one of several linked environments. Examples are *‘Learning Activities with Conditions’*, where a student decides the granularity level that he wants and *‘From Lesson Plan to Learning Design Level B’*, where again a student takes control and switches on or off the audio support of the UoL [109]. These examples map to AEH adaptability. An additional approach to content-based adaptation is through the modification of contents linked to fixed resources and based on external tools. For instance, a resource linked to a wiki service hosted outside an IMS-LD UoL could adapt its content dynamically, based on students’

tutors' or authors' contributions. This also maps to AEH adaptability, based on interaction of several users and user types (thus is not adaptation).

4.3.2.4 Interactive Problem Solving Support

This type of adaptation involves helping students to solve problems in a gradual, adaptive way. Adaptive Hypermedia can produce interactive problem solving support. For example, one could create a set of questions and gradual hints for the students, in order for them to find the right answers. One of the first AEH systems, the LISP tutor [7], was based on interactive problem solving. Adaptive Hypermedia however does not offer any support for author or teacher interference with the run-time flow. For IMS-LD, the latter type of interaction is more appropriate. For example, problem support can be carried out 1) by a tutor editing previous hints, 2) by executing specific design-time rules, or 3) by a combination of both mechanisms. An example is 'What is Greatness' [109], where the tutor moderates the contributions of a group of students on a particular problem, providing access to the next step when the tutor thinks that the current one is finished.

4.3.2.5 Adaptive Information Filtering

Adaptive Hypermedia relies heavily on information filtering. If we look at LAOS, then the Goal Map collects and filters information from the Domain Maps. The adaptation then does the lower granularity filtering, to show the user exactly what is appropriate, according to his profile. Thus, AEH and some of its methodology (especially the research on open corpus adaptive hypermedia) is related to adaptive information filtering, but, as to date, they are still different areas. IMS-LD is not designed to provide adaptive information retrieval or filtering. Some rudimentary facilities are available through the index-search service.

4.3.2.6 Adaptive User Grouping

In IMS-LD, by using management systems provided by several IMS-LD tools and engines – Coppercore, Reload, CopperAuthor [146] - once the UoL is published, the administrator (for example, the teacher) can add or delete users and assign them to a specific run of that UoL. This means a de facto grouping [25]. However, the dynamic creation of roles after the publishing process is not currently possible. Once a definition of roles or stakeholders is

available and a UoL run is defined, specific users can be added to, or removed from, any of these groups and these users can play the run. While not yet incorporated in all AH systems, Group adaptation is a topic that is attracting more interest recently [127], [149].

4.3.2.7 Adaptive Evaluation

Adaptive Evaluation can be performed in Adaptive Hypermedia. In LAG, for example, scores could be kept in user model variables and then used to adapt the content the user sees to the scores of previously performed evaluations. In IMS-LD, taking the performance of a student in a Unit of Learning as input, a full set of parameters can be stored in local properties to be used in the adaptation of formative or summative evaluations. In Geo Quiz 3, certain actions and answers of a student can be allocated into variables pre-defined at design-time and they can also be interpreted at run-time following a set of rules. In this way, both the evaluation system, the content itself and even the interpretation of the results, can change for each student.

4.3.2.8 Changes On the Fly

In IMS-LD, with the current tools, once a UoL is published it is not possible to change structure, method or definition of basic parameters (such as conditions or properties). However if a UoL is so designed, a tutor is able to change the way a student perceives the course and the flow: 1) the tutor can update the content, based on pre-defined content or on new contributions; and 2) the tutor can also influence the learning itinerary, uploading files, showing and hiding content elements and structure elements. This means that a tutor is able to change things on the fly, as long as he had previously defined that possibility in design-time.

After authoring in AEH, the author normally does not interact anymore with either the students or the content. This is mainly because the focus has been on creating automatic adaptation and not on providing a software tool for teachers to communicate in real time with their students. Therefore, adaptation on the fly as in IMS-LD is currently not possible in Adaptive Hypermedia.

4.4 Results Summary according to Comparison Criteria

In Table 4 we present a brief summary of our comparison. The table shows the types of adaptation we analysed and whether these types can be achieved using IMS-LD or AEH. The table describes the type of adaptation, whether it is supported in IMS-LD, AEH and comments on how the support can be achieved. The table gives an overall idea of how IMS-LD and CAF and LAG can benefit from each other. CAF and LAG already convert to many formats (such as RDF [129], IMS-QTI, IMS-CP [99]) and systems (such as Blackboard [15] and WHURLE [117]). Therefore, a straightforward step would be to export CAF content and LAG strategies into IMS-LD at least to the extent to which this is possible, as suggested by Table 4.

Table 4 Comparison of types of adaptation

Type	IMS-LD	AEH: CAF and LAG	Comments
<i>Adaptive multimedia presentation</i>	<i>Partially Supported</i>	<i>Partially Supported</i>	IMS-LD: Can select multimedia presentation in an adaptive way but not directly adapt the multimedia presentation. CAF and LAG: Cannot support adaptive multimedia presentations directly but can generate adaptive SMIL [140].
<i>Natural language adaptation</i>	<i>Not Supported</i>	<i>Not Supported</i>	IMS-LD: Does not support natural language adaptation. CAF and LAG: Does not support natural language adaptation.
<i>Canned text adaptation</i>	<i>Supported</i>	<i>Supported</i>	IMS-LD: Supports adaptation of canned text fragments. CAF and LAG: Supports adaptation of canned text fragments.
<i>Adaptation of modality</i>	<i>Partially Supported</i>	<i>Supported</i>	IMS-LD: Modalities can be added to resource file, and opened upon certain conditions. CAF and LAG: Can support adaptation of modality in a similar way to canned text adaptation or adaptive multimedia presentation.
<i>Direct guidance</i>	<i>Supported</i>	<i>Supported</i>	IMS-LD: Can emulate menu-adaptation using DIV layers in resources. CAF and LAG: Supports adapting navigational elements (menus, links) for direct guidance.
<i>Adaptive link sorting</i>	<i>Partially Supported</i>	<i>Supported</i>	IMS-LD: DIV layers can be defined with the correct sorting, but links cannot appear in different orders. CAF and LAG: Supports link sorting, a simple example is the 'To Do' list.
<i>Adaptive link hiding</i>	<i>Supported</i>	<i>Supported</i>	IMS-LD: Supported in the same way as showing and hiding text. CAF and LAG: When hiding concepts, also the links to it are hidden (or annotated)
<i>Adaptive link annotation</i>	<i>Supported</i>	<i>Supported</i>	IMS-LD: Using DIV layers in resources. CAF and LAG: Links in the menu are annotated based on suitability using coloured bullets.
<i>Adaptive link generation</i>	<i>Partially Supported</i>	<i>Supported</i>	IMS-LD: Requires changes on the fly to the XHTML, this is possible via UoL runtime, but only if the change (adding a link) has been provided for at design-time. CAF and LAG: Links are provided in the menu based on suitability for example.
<i>Map adaptation</i>	<i>Partially Supported</i>	<i>Not Supported</i>	IMS-LD: Can be achieved via a series of pictures, or using DIV elements, but it cannot be authored by current authoring tools. CAF and LAG: There is some research in this direction but most systems do not support map adaptation.

Table 4 Continued: Comparison of types of adaptation

Type	IMS-LD	AEH: CAF and LAG	Comments
<i>Interface based</i>	<i>Partially Supported</i>	<i>Supported</i>	IMS-LD: The adaptation of the interface in current IMS-LD player is limited. However some interface adaptation can be done in the UoL using the DIV elements. CAF and LAG: This is the most commonly encountered type of adaptation in AEH after canned text adaptation.
<i>Learning flow based</i>	<i>Supported</i>	<i>Supported</i>	IMS-LD: IMS-LD is built around adapting the learning flow and sequencing dynamically. CAF and LAG: Creating different sequencing is fundamental in AEH.
<i>Content and activity based</i>	<i>Supported</i>	<i>Supported</i>	IMS-LD: The content of an activity needs a resource linked to the element 'Activity Description'. The content in a resource, of predefined properties/variables and of an activity can be adapted by switching, showing or hiding one of several linked environments. CAF and LAG: Content based adaptation can be performed via adaptability (user initiated) or adaptivity (system initiated). If the content is regarded as activity content, such content adaptation can be achieved in AEH by switching between the showing and hiding of several linked environments.
<i>Interactive problem solving support</i>	<i>Supported</i>	<i>Supported</i>	IMS-LD: Support can be carried out by interaction with a tutor or via rules defined at design-time. CAF and LAG: Some earlier AEH systems were built around adaptive problem solving support. However there is no support for author or teacher interaction
<i>Adaptive information filtering</i>	<i>Partially Supported</i>	<i>Supported</i>	IMS-LD: Some rudimentary facilities are available through the index search service. CAF and LAG: The Goal Map filters information from the Domain Maps, the adaptation does lower granularity filtering, to show exactly what is appropriate, according to the User Model.
<i>Adaptive user grouping</i>	<i>Supported</i>	<i>Not Supported</i>	IMS-LD: Support for roles and groups are explicitly supported. CAF and LAG: In general not supported but it is an active area of research.
<i>Adaptive evaluation</i>	<i>Supported</i>	<i>Supported</i>	IMS-LD: Explicit support of tests. CAF and LAG: Explicit support of tests in some systems simulated using canned text adaptation in others.
<i>Changes on the fly</i>	<i>Supported</i>	<i>Not Supported</i>	IMS-LD: Supports interaction between the teacher and the running UoL. CAF and LAG: Not supported

4.5 Related Research

What IMS-LD and LAOS (with the current instantiation of CAF and LAG) have in common is the fact that they intend to provide generic and flexible languages for expressing, in the first case, various pedagogies and in the second, various adaptation forms. In the following, we will look into other research into such generic and flexible languages, as well as into pattern languages [69] for related purposes.

EML (Educational Modelling Language) is the predecessor of IMS-LD. It has also been developed by the Open University of The Netherlands and it was an early attempt to codify

units of study including the roles and interactions involved, in a standard way. EML has now however been supplanted by IMS-LD.

In the domain of AEH, another language, LAG-XLS [136] has been developed, as a special-purpose language to expressing learning styles for AEH. While this language is flexible and XML-based (thus easily transportable), it is not general purpose.

An example of related research in the educational domain is [9]. This chapter presents a first attempt to extract design patterns for Adaptive Web-based Educational systems based on Adaptive Hypermedia, detailing especially the user modelling patterns and their relations. These patterns could inform strategies as described by LAG or IMS-LD, for example, although the chapter keeps the granularity of the patterns relatively high.

Another example of an adaptation language in a different domain of adaptation is the generic language for adaptation of system services in middleware as described in [78]. This language allows system services to be dynamically adapted to suit the needs of the specific middleware. Commonalities exist at the level of adaptation specification, but, as the domains are quite different, the similarities end there.

In yet another application domain, the research into Pattern Languages [69] at the University of Oregon for planning its campus layout could be considered related, in the sense that this research uses pattern languages to be able to adapt to changes in campus development. This type of research shows the flexibility of using patterns in defining adaptation and, as a next step, special purpose languages for adaptation, as in the current chapter.

Authoring and pattern languages go hand in hand, as shown for the domain of e-learning [31]. This research is still a work in progress, as is the one in [13], in which a simpler authoring system for reusable rules for IMS-LD is being designed. The latter is attempting to marry IMS-LD with AEH concepts.

However, there is as yet very little research on generic languages that allow authoring and engineering of adaptive learning material in such a way that reuse of material is allowed.

4.6 Conclusions

Adaptation is a term used in many ways: some consider adaptation to be personalisation, that is, changes in a (learning) system that reflects changes in the user (learner). Others look further and include in the term changes related to the (perceived) quality of service of the respective (learning) system, or to the device used (for example, handheld versus desktop). As we saw in section 4.5, adaptation can also be seen in a much broader sense, for example, adapting campus planning towards its users needs. One of the problems therefore when comparing systems that claim to provide adaptation and personalisation, generally speaking, but in particular for the e-learning field, is the fact that the definitions used for the term ‘adaptation’ vary greatly.

In this chapter, we felt compelled, therefore, to use two definitions of adaptation, one based on the Adaptive Hypermedia taxonomy by Brusilovsky [21] and the other, a pedagogy-based adaptation classification, most used to describe adaptation in IMS-LD [28], in order to merge the concept of adaptation in personalised distance learning for these two fields. We have seen that, whilst adaptation is possible to some extent in IMS-LD, it is clear that this field would benefit from the previous findings in AH and AEH in particular. Conversely, AEH does not provide for specific features that are required for flexible pedagogical settings, such as on the fly changes and multiple user roles. From AEH systems point of view, the current view is that such functionality can be provided externally, by other systems. However, adaptivity, could also be adaptivity ‘for collaborative groups. In such cases, for instance, adaptation and roles are intrinsically related, so a merged solution needs to be found. The precise extent to which both AEH and IMS-LD support adaptation for collaboration still needs to be explored further. Also further research is necessary to find out how exactly the knowledge of the two fields could be

combined; as IMS-LD is the de facto emerging standard, the best option might be to extend IMS-LD with the whole range of AEH functionality.

This chapter thus merges the ontologies of IMS-LD and CAF and LAG at a high level of semantics. This is an important step in connecting two seemingly unrelated fields, that of adaptive educational hypermedia and that of IMS Learning Design.

5 Improved Reuse Capabilities of Adaptation Strategies

The use of adaptive systems [21] is increasingly popular. Commercial systems on the web (for example, Amazon) or beyond (PDA device software) present at least a rudimentary type of adaptation. However, adaptation specification cannot be fully expressed by standards³² yet and most commercial and non-commercial systems rely on proprietary, custom designed, system specific, non-portable and non-interoperable adaptation. An intermediary solution, until standards emerge, is the creation of Adaptation Languages, which, with their power of semantics-based reuse, appear as a reliable future vehicle for all [43], [136]. Once written, the same adaptation strategy can be used for various domains. For example, the strategy for beginner-intermediate-advanced written in the LAG language [43], could be used to teach students of varying knowledge level studying databases, mathematics or poetry. Similarly, the same Domain Model can be used with various adaptation strategies. For example, a Java course can be taught via a strategy differentiating between beginner and advanced users, or between visual and verbal users. However, there are a number of limitations regarding adaptation engines, which ultimately influence the efficient authoring of adaptation strategies, as based on an analysis of Interbook³³ [64] WHURLE³⁴ [118], AHA! [52], [54] and Personal Reader [3].

Thus, in this chapter we define and analyse these limitations, illustrating them via a case study of a simple, yet powerful Adaptation Language, the LAG language [43]. Moreover, we propose a meta-language, as a supplement to LAG, showing how introducing it can overcome such limitations. Importantly, this solution is compatible with existing adaptation engines, instead of requiring the creation of new engines. The rest of this chapter is organised as follows. Section 5.1 introduces the issues and limitations that adaptation

³² SCORM Simple sequencing allows basic adaptation. IMS-LD promises more for the future.

³³ <http://www.contrib.andrew.cmu.edu/~plb/InterBook.html>

³⁴ <http://whurle.sourceforge.net/>

engines pose regarding reuse. Section 5.2 presents a case study, illustrating the need for improved reuse capabilities. Section 5.3 outlines possible solutions to the issues and limitations of Adaptation Engines. In Section 5.5 we introduce our meta-level addition to the LAG adaptation language to realise some of the possible solutions. Section 5.5 outlines related research and in section 5.6 we outline conclusions and further work

5.1 Adaptation Engine Issues and Limitations

The following are issues and limitations identified as influencing the authoring flexibility of adaptive hypermedia systems:

- L1. Most adaptive hypermedia delivery systems determine the adaptation on a per-concept base [3]. A broad knowledge of the whole content at every adaptation step is (usually) unavailable, mainly due to run-time complexity limitations. Thus, adaptation strategies cannot specify complex inter-concept rules; for example, a strategy with an arbitrary set of labels denoting topics of interest, displaying to the user concepts related to his topic, without limiting the possible topics at design-time.
- L2. Adaptation engines do not (usually) allow for non-instantiated program variables [3]. Thus, authoring strategies which involve an unknown number of types and categories are currently not permitted. All domain-related variables need to be instantiated in the authoring stage.
- L3. There are extreme difficulties arising when combining multiple strategies [3]. Adaptation engines usually update sets of variables based on some triggering rules, without knowing which high-level adaptation strategies these variables represent. An example of a combined strategy currently difficult to implement is one where the system checks whether the user prefers text or images and then displays the preferred type of content, filtered via a beginner-intermediate-advanced strategy, where concepts are shown based on the user's knowledge.

In AHA! [52], [54] reasoning is mainly done on a per-concept base (for persistent attributes). Volatile attributes can contain expressions, which reference other attributes, allowing for backward reasoning. However, this does not fix limitation L1 entirely. This

method only allows for access to variables concerning concepts that have already been visited before or are in the same line of hierarchy. AHA! also does not allow for any free program variables (L2). AHA! can combine strategies (via the LAG language [43]) but does not offer any solution to conflicting naming (L3).

InterBook [64] uses a knowledge-based approach to create adaptive, interactive electronic textbooks. Adaptation is more limited than in AHA!: it uses a classification of domain concepts into a spectrum and allows for adaptation towards the user's current knowledge state. The prerequisites are computed on a per-concept base and neither free variables nor combined strategies are possible (L1-L3).

In WHURLE [118], the lesson plan specifies a path through the content chunks. Rules are defined on a per-concept base (L1) and no free program variables are allowed (L2) [106]. Multiple strategies are possible by using XML pipelines [156]. The issue of different strategies using conflicting naming (L3), however, remains.

Personal Reader [3] can deal with more sophisticated issues. It uses an RDF ontology with complex reasoning, so limitation L1 does not apply. However, it still does not offer free program variables (L2). Combining rules in an RDF ontology is less problematic, as multiple relationships can be defined at the same time. There are however limitations as to what can be implemented efficiently. For example, if we look at the OWL³⁵ ontology language (based on RDF), we see that although OWL Full is complete and has no limitations as to what can be expressed; only the very limited set of OWL Lite can be implemented efficiently.

5.2 A Case Study

Section 2.5.1 introduces the LAG language. It contains an example, showing a strategy, contains three levels, *beginner*, *intermediate* and *advanced*, and adapts the content show to

³⁵ <http://www.w3.org/TR/owl-ref/>

the user based on their level. The strategy works well because it ‘knows’ what labels to expect in the CAF file representing the Goal and Constraints model: ‘beg’, ‘int’, ‘adv’. Currently, if other labels are also present, the conversion ignores them. However, what happens if we want to represent strategies with more complex labels, such as the ones in the previous section? There, we had, for example, various labels starting with ‘*beginner_*’ or ‘*intermediate_*’, but we did not know in advance how many types of such labels exist. Still, we should expect to be able to perform some adaptive strategy and express it in the form of an adaptation program. As variables need to be instantiated, this introduces an intermediate step in the processing, as the next section shows.

5.3 Solutions to Adaptation Engines Issues and Limitations

Previously, we have seen an illustration of two of the current limitations listed in section 2: (L1) concept-based adaptation, where the same rule has to be copied in all concepts and one concept does not (normally) affect other concepts directly and (L2) the fact that adaptation engines do not allow for non-instantiated variables. A straightforward way of defeating these problems would be to build new adaptation engines. The first scenario could be achieved by establishing which labels exist, in the initialisation step. The second issue could be overcome by either allowing arrays of labels, or otherwise allowing multiple data to be stored in the label. However, in order to function with current systems, these issues should be solved in the authoring stage. For the third limitation (L3), the difficulty in application of multiple strategies, the MOT to AHA! converter, for example, has already implemented an elegant solution (unique to our knowledge so far), in that it can apply multiple LAG files, with different adaptation strategies, with the order of execution set by priorities of the respective strategies (1: highest priority; any following number: lower priority):

```
priority x    /* where x is a number */
```

Nevertheless, this method could override previous variables (for example, if two strategies use `UM.Concept.knowledge`, only the update of the highest priority strategy counts). Thus, a unitary strategy merge, keeping track of all variables in use, based on multiple labels for

domain-related concepts and attributes, is preferable. Moreover, many types of variables (for example, arrays) are not allowed by Adaptation Languages, due to lack of adaptation engine support, limiting the adaptation that can be expressed.

However, we have noticed that a) strategies have common patterns, as has already been shown previously in [20], which could be reused; b) templates based on these patterns could reduce the designers' work; c) there is a strong preference towards XML-based processing and interfacing. Thus, XML-based templates should be used to move the processing to the authoring side and facilitate the extraction of reuse patterns.

For the creation of LAG files based upon a LAG template, explicit knowledge about the content is needed. CAF represents a flexible format for Adaptive Hypermedia content and is also used by AHA!. Therefore, a CAF file will be our choice for the content. For the LAG template files, the LAG files which follow the extended LAG description introduced in section 2.5.1 will be required. A pre-processor can replace the constructs added in section 5 by traditional LAG constructs. The resulting LAG file will then describe the same adaptation behaviour as the template LAG file, but for the specific labels encountered in the CAF lesson.

Implementing the pre-processor as a web-based application enables it to transfer both the unchanged CAF file as well as the resulting LAG file to the AHA! system, provided, of course, the appropriate rights are set and the pre-processor is on the same system as AHA! (currently AHA! only allows uploading files through a Java tool). To facilitate the use of multiple strategies, it should be possible to select multiple LAG templates. The user should be given a choice between creating the AHA! lesson and downloading the resulting LAG file. This process could, if the direct lesson creation is used, smoothly replace the current process, without requiring any extra effort from the user. This process is shown in Figure 24.

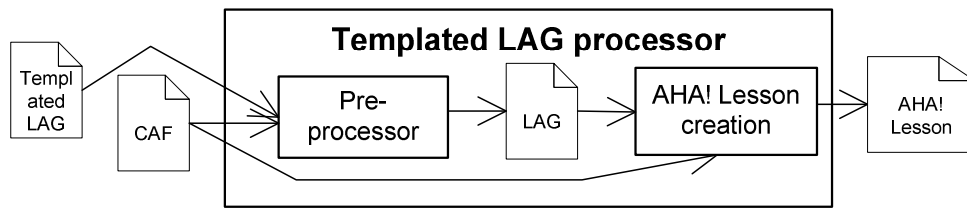


Figure 24 System setup of template LAG Pre-processor.

The approach however has some drawbacks. While the meta language makes strategies more re-usable, it also makes them difficult to read and create. The approach leads to the creation of self-modifying code, which makes it harder to maintain the strategies. Therefore this approach is an explicit departure from the idea that teachers should create their own strategies. Experts will have to create strategies which teachers can simply reuse. It is clear that for the approach to work, experts need to be available to write the reusable templates. While the chapter shows that the approach can technically work, it has not been tested if teachers are able to easily reuse the strategies.

5.4 Meta-level Addition to LAG

To solve the limitations mentioned in section 2, we add, as said, a pre-processing step to the whole authoring process. This step takes a LAG template and the content, in the form of a CAF file, and pre-processes it. The result is a new LAG file which extends the strategy sketched by the LAG template for the specific content described in the CAF file. We want to accommodate future changes to LAG, as well as have our approach be reusable and easily implemented and maintained. Therefore we propose an XML-based notation for the template LAG files, while keeping the original LAG language unchanged for compatibility with current systems. Note that alternatively the changes could be incorporated into the LAG language directly but then it would lose its compatibility with existing systems. Since CAF is already written in an XML-based notation, both documents can be used as input for an XSLT transformation which generates the resulting LAG file. Below we give the DTD (document type definition) for the template LAG file.

```

<!ELEMENT TLAG ((LAGfragment*, LIKE*))>
<!ELEMENT LIKE attribute CDATA value CDATA (LAGfragment, MATCH,
LAGfragment, (LAGfragment*, LABEL, LAGfragment*))>

```

```

<!ELEMENT LAGfragment (#PCDATA)>
<!ELEMENT MATCH EMPTY>
<!ELEMENT LABEL EMPTY>

```

A template LAG file consists of a number of blocks of the following kind: a number of LAG fragments followed by a *LIKE* block. The LAG fragments contain LAG adaptation program snippets, similar to the examples showed in section 3.3. The *LIKE* blocks consist of an attribute and a regular expression against which it is matched, followed by a fragment of LAG program. The word *MATCH* represents the place where the *LABEL* needs to match the regular expression.

Below we show a fragment of the beginner-intermediate-advanced strategy. It shows how template LAG can be used to create an adaptation strategy that works with a CAF file, where the content is labelled with *beginner*, *intermediate* or *advanced*, according to its difficulty:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Server SYSTEM "ttag.dtd">
<TLAG> ...
  <LAGfragment>UM.GM.beginner_number= 0 </LAGfragment>
  <LIKE attribute='GM.Concept.label' value='*beginner*'>
    <LAGfragment> while(UM.GM.label= </LAGfragment>
    <MATCH/>
    <LAGfragment>)(UM.GM.beginner_number+=1)</LAGfragment>
  </LIKE> ...
</TLAG>

```

The following is an extract of the result of the pre-processing of a LAG template and the CAF file of the earlier example.

```

while (UM.GM.label.beginner_title||UM.GM.label.beginner_text.)
  (UM.GM.beginner_number+=1) ...

```

The complete result is a LAG file, tailored towards the content of the CAF file. In the snippet above we see that the variable *UM.GM.beginner_number* is increased by one for each variable using the label *UM.GM.label.beginner_title* or *UM.GM.label.beginner_text*. These were exactly the labels matching the regular expression *'*beginner*'*. Applying (the DTD of) the LAG template solves some of the problems mentioned before.

Below we list again the limitations, discussed before, and describe our solutions for these limitations.

L1. *Limitation:* adaptation on a per-concept base; a broad knowledge of the whole content at every step of the adaptation is (usually) unavailable.

Solution: such knowledge is not necessary in the adaptation engine. It is acceptable that this type of knowledge can be acquired as a one-off, at authoring time, as it is not to be expected that content labels will change at execution time. Therefore, the authoring strategy should contain this knowledge. As for an author it is difficult to manually extract all the pedagogical label types existent in a course, templates such as the DTD of the template LAG above can help in dealing with groups of labels (such as all labels containing ‘beginner’, that is, ‘*beginner*’). An author can then generate the appropriate adaptation strategy (of which a snippet is shown above) in an easy and quick manner, making use of existing patterns in the authoring strategy itself.

L2. *Limitation:* adaptation engines do not usually allow for non-instantiated program variables.

Solution: unknown domain-related variables can be instantiated in the authoring stage, with the help of patterns specified via the LAG template language based on the above DTD. It is not necessary for an author to perform these searches manually; the two-step authoring system can extract unknown variables for him.

L3. *Limitation:* the extreme difficulties arising when combining multiple strategies.

Solution: similar pattern extraction mechanisms have to be used in order to merge adaptation strategies. In (nearly) every system there is a limited number of weights and labels; this causes problems in combining a number of strategies greater than the number of weights and labels available. A solution to this can be to apply pattern matching on labels in order to be able to encode multiple strategies, by using the same label field. This thus enhances simple prioritisation of strategies, as it allows the combination of multiple strategies which each requires specific labels.

5.5 Related Research

In the creation process of AH, reusability is an important topic. Reusability considerations regarding the content offered in AH systems have been given a lot of attention. It was one of the key features of the LAOS model [41], see also section 2.4.8 and other systems such as ACCT [50], [51], see section 2.7.4, with its Subject Matter Concept Space and to some extent WebML [30] and XAHM) [29] have also taken it into account.

However reusability considerations also extend to the definition of adaptive behaviour. For this very reason the LAG framework [43] (see section 2.4.9) was introduced. The LAG-XLS [136] language defines an XML-based extension to the LAG language, which aims at supporting frequently used instructional methods to support Learning Styles. Similarly to LAG-XLS our work has defined limitations of the LAG language. However the aim of the LAG-XLS language is to enable direct support for representing Learning Styles. Our aim has been of a more generic nature: to promote reuse of adaptation strategies in general.

A key part of the approach is the steps taken to allow strategies to be combined more easily. Additional work by others has only highlighted the need for an easier way to combine strategies. In [134] the authors indicate see how an adaptation strategy could be based on media mix theory and how defining such a strategy would be come much easier if it could be defined in separate strategies and combined effectively. This would then enable non-technical authors to select and use the desired combination of strategies.

5.6 Conclusions

In this chapter we have analysed adaptation problems inherent in current adaptation engines, which reduce the power and generality of Adaptation Languages. We described and exemplified these issues with the help of the LAG language, currently one of the only exchange formats of Adaptation Language specification between systems. Moreover, we have moved one step further, by proposing improvements that can overcome run-time issues of adaptation engines, by solving them at the authoring stage. More specifically,

templates can be used to create adaptation strategies, customised for the given Domain Models and pedagogical labels. For this purpose, we have proposed the template LAG language. The process is technically implemented by adding a pre-processor to the system setup, which has access to content at compile-time only. In such a way, more powerful adaptation strategies can be created for existing adaptation engines.

6 Graphical Authoring with CAM

On-line courses are becoming increasingly popular and the use of so called Learning Management Systems (see section 2.1.9) is becoming more widespread. As mentioned, Adaptive Hypermedia has the potential to offer a richer learning experience, personalised for each learner. To realise this potential, however, education authors need to have the ability to easily create adaptive material. The GRAPPLE³⁶ project aims to integrate AH with major Learning Management Systems and to provide an environment that delivers personalised courses in a LMS interface.

However, designing AH is a much more complex and time-consuming task than creating a course in a LMS. As previously discussed, there exist several Adaptive Hypermedia reference models, like AHAM [153] and LAOS [41], that are specifically developed for authoring (instead of general purpose models of AH, such as XAHM [29], Munich [104] and GAHM [122]). There also exist several tools built based on these models. The AHA! graph editor, see section 2.7.2, for example offers an easy to use visual interface to authoring of AH. The MOT tool on the other hand, which we saw in section 2.7.1, offers a simple web-forms based interface, which features extensive reusability, not only of the content, but also of the adaptation as whole strategies. Other tools, as we saw in section 2.7, all have their particular strong points, however, even when using tools developed based upon these models, authoring remains a difficult and time consuming task. Authoring of adaptation strategies is particularly difficult for the layperson author, or for the author who may be a domain specialist, but have no programming experience. A novel reference model for the use of graphical tools is introduced in [84]. Existing graphical authoring tools (for example the Graph Author developed for AHA! [3]) use concrete connections between concepts and the adaptivity is specified in a single layer. This approach means that the reusability of the adaptivity is limited.

³⁶ <http://www.grapple-project.org/>

To simplify adaptive behaviour authoring for an educational author, a visual environment is considered most intuitive. We have previously introduced the CAM model [84], a generic model for authoring that allows adaptation to be defined in a visual, graphical, flexible and user friendly way. In this chapter we refine this model and introduce the *design of the new authoring tool*, which allows the specification/creation of a CAM instance in a visual manner. In the overall authoring approach of the GRAPPLE project³⁷, a tool is being developed to translate these CAM instances into usable adaptive courses for multiple adaptation delivery systems, like, for example, AHA!. Finally, we introduce a *novel set of adaptation languages* to deal with the new way of authoring adaptation in a visual fashion.

The remainder of this chapter is organised as follows. In section 6.1, we will give a motivating scenario, showing how the tool based on our model will help a teacher without technical background author an adaptive course. In previous research various pedagogical strategies have been implemented in Adaptive Educational Hypermedia. In section 6.2 we discuss how a selection of these strategies can be represented using the new tool. Section 6.3 will introduce the CAM model. In section 6.4 we outline the design for adaptation languages. The shell and web services are briefly described in section 6.4. The design of the authoring tool is outline in section 6.6. Appendix I features the user guide to the GRAPPLE Authoring Tool, which describes the tool from a users perspective using a simple example course. Section 6.7 discusses an evaluation of the authoring tool. Section 6.8 gives an overview of related research and the associated CAM model and finally we conclude in section 6.9.

6.1 Motivating Scenario

We illustrate the authoring process of a *CAM instance* by means of a scenario in which a teacher needs to express some *generic* and *specific prerequisite relationships*.

³⁷ <http://www.grapple-project.org/>

Dr. Williams prepares a new on-line course on 'The Solar System' for first year undergraduate students. He has two options: he can either try to define a link structure between the course pages in such a way that students never see a link to information they cannot yet understand (because of missing previous knowledge) or he can define a CAM instance with prerequisite relationships and then rely on the Adaptive Learning Environment (ALE) to ensure that students are only guided towards pages for which they have all prerequisite knowledge. Although it is often argued that defining adaptation (a CAM instance in this case) means creating an adaptive course and is more work than creating a static course, the converse is actually true: the first option, to create a static course that is such that students can only follow links to information they are ready to understand, is a nearly (or perhaps completely) impossible task and would require a lot of very careful work in selecting links to show to (all) students. Moreover, if differences in students' needs are to be modelled in such a structure, it would mean that Dr. Williams would have to define a different course for each student. Instead, Dr. Williams opts for using the GRAPPLE Authoring Tool to create his adaptive course. He realises that there are many ways to present the material of his course. He decides that he first wants to introduce the general notions of the Solar System, star and planet, and then teach about the specific stars and planets in our Solar System. He, however, decides he does not mind which planet students start with (Note that Dr. Williams could have chosen to present first the stars and planets and then the general notion of a Solar System, star and planet, or even an entirely different strategy altogether).

Dr. Williams wants to create a prerequisite relationship from "Planet" to "Jupiter", as the students should first learn something about the notion of a Star before learning about the Sun. Then he repeats this process of linking for every Star and Planet in his course, to end up with the desired result. The authoring tool allows authors to draw a prerequisite relationship between a set of (prerequisite) concepts on the left and a set of concepts on the right. First, he drags the prerequisite relationship onto the canvas, which would look as shown in Figure 25 (upper part of image).

Then he drags the concepts *Planet* and *Jupiter* to the sockets source and target, respectively. He repeats this for every Planet and Star. After that the display would look as shown in the lower part of Figure 25. These relations represent *specific prerequisite relations*.

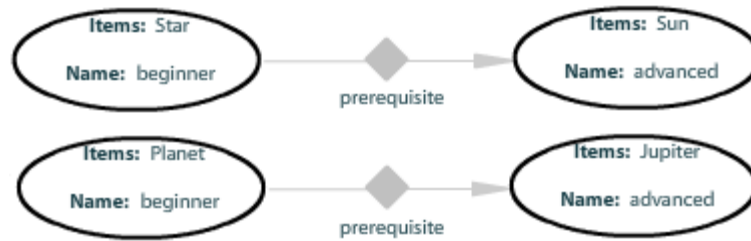


Figure 25 Instantiated CRTs in the CAM for Planet and Star and two of the concrete planets and stars

This would create the result Dr. Williams ultimately wanted to achieve. However, Dr. Williams realises that, if he persists with his chosen teaching strategy, “Planet” is not just a prerequisite for “Jupiter” but for every planet in the Solar System. He realises he can reduce the amount of work he has to do and decides thus to use a different CRT instance, as displayed in Figure 26.



Figure 26 Generic relation between the general concepts Star and Planet and the concrete star and planet Sun and Jupiter

Thus, the specific concept relationships above are replaced with one *generic relationship*, called *is_a-prereq*: the sockets can contain any number (≥ 1) of concepts. The behaviour will be that each concept in the left socket is a prerequisite for the concept in the right socket, to which it is related via the “*is_a*” relation in the Domain Model. This illustrates a situation where a domain relationship can be reused to influence adaptation, thus associating certain behaviour with this relationship. Thus, the CRT is applied to the whole domain map. Alternatively, Dr. Williams could have just dragged and dropped the concepts

he wishes to participate in this generic prerequisite relation, and not reuse the Domain Model relation.

Something perhaps not immediately obvious from this example is that there are two implied possible uses of this authoring tool (plus a combined third one):

- In the example, the “*is_a*” attribute is a Domain Model (DM) property, probably prior to this derived, from a subject-ontology. Which concepts have “Planet” as a prerequisite depends purely on the DM and this is thus independent of the learner taking the course.
- It is equally possible to use an attribute from the user model in a relationship, thus creating relationships that are not only user-dependent, but also dependent on the “current” instance of the user model.
- There is finally a third case, by combining the previous two. The learning application can for instance *recommend* topics from a list that, first of all, depends on the DM, but that also depends on the user’s knowledge. For instance, only those recommended topics may be shown of which the user still has little or no knowledge.

Note that when the relationship only depends on DM information (like in the example), Dr. Williams may not have to define the CRT himself. He would just use an appropriate one from the CRT library. This is why the descriptions of the CRTs are very important for Dr. Williams to make sense on how to use CRTs written by other authors or programmers.

6.2 Pedagogical Strategies in the GRAPPLE Authoring Tool (GAT)

In the previous section, we have seen a scenario illustrating how a teacher can create or customise an adaptive lesson. Previous research has defined interesting pedagogically sound adaptation strategies, representing different learning scenarios based on learners’ needs and preferences, and some also based on complex (and controversial) pedagogical

foundations, such as learning styles, for Adaptive Hypermedia³⁸ [6], [34]. In this section we will explore some of these strategies in relation to the GAT. More specifically, we will check how, in principle, such strategies can be expressed in the new CAM component of the GAT and CAM languages. As the CAM model is aimed to be richer than previous attempts, it should at least be able to express the basic strategies we have defined before via languages such as LAG [43] and LAG-XLS [136]. The CAM model and the GAT are more flexible, however, and can express strategies beyond what is analysed here.

The aim is to show that the new tool will allow for enough expressivity to articulate most of the strategies that were designed with previous tools. The aim is not to provide a complete set of strategies or even to present the best strategies for teaching. We have therefore selected some of the more technologically challenging strategies. The use of learning styles, upon which some of these strategies are based, is controversial [34], but we argue that an authoring tool should allow for expressing them, as long as there are teaching situations where such adaptivity is required.

Note that the solutions we present below are not unique solutions for these strategies in the GAT. As in any language, the CAM visual language allows for some redundancy, and often the author's preference can determine which way is used to express a certain adaptive behaviour. These solutions should be seen as example solutions.

Also note that we only illustrate the visual adaptation language here, describing the adaptation behaviour in natural language. This is the level that would be enough for Dr. Williams to create his adaptation with. However, the adaptation behaviour associated with each CRT needs to be described in an adaptation language (such as LAG [45], [48] or GAL [145]), but this complexity remains hidden to the average author.

³⁸ See also our strategies page: <http://prolearn.dcs.warwick.ac.uk/strategies.html>

6.2.1 Rollout

The rollout strategy is a very simple strategy that allows authors to decide when a certain concept or concept part should be shown, depending on the number of times a user is visiting given concepts. Specifically, concepts to be shown after a certain number of steps could be classified as *'showafter'* and metadata containing that number of steps, can be attached. Similarly, concepts classified as *'showatmost'* (should only be displayed at most the given number of steps) can be stored in metadata. Pedagogically, such a strategy is useful when, for example, beginner users need more information at the beginning, which is irrelevant later on and needs to be removed (thus using *'showatmost'* for a certain number of steps). Similarly, not all information should be made available from the start, to avoid cognitive overload. Some of it can appear later, when the learner is ready for it (thus using *'showafter'* a given number of steps).

The strategy demands that a concept is shown after its parent has been viewed a given number of times. For this purpose we create a CRT for the relationship called *'rollout'*. The CRT has two sockets called *'showafter'* and *'showatmost'*. The relationship's associated adaptation determines that the resources in the socket called *'showafter'* will be displayed after the parent concept in the Domain Model, have been visited a number of times by the user. For this strategy, we assume the *'is_part_of'* relationship between Domain Model concepts fulfils the parent-child relationship. Similarly, the concepts, which should be shown at most a number of times, are placed in the *'showatmost'* socket.

It is not acceptable that the number of times after which a concept should be shown or hidden would be added in the DM, due to its pedagogic and adaptation-specific nature. Thus, two possible processing methods for this exist:

- The label (metadata) is not added; this will require distinct relationships for each number. So *'showafter2'*, where the target is shown after the source has been seen 2 times, and *'showafter3'*, where the target is shown after the source has been seen 3 times, would be distinct CRTs. This is very inefficient and a burden on the author.

- The CRT can take a number of parameters, which can be filled in when it is instantiated in the CAM. This way, there can be a parameter for the number of times after which a concept is shown and for the number of times after which it is hidden. This means that the same CRT can be used with different settings for the number of times. In fact there could be a number of sockets, with each a parameter belonging to it. This way, for example, some concepts are shown after visiting the parent concept 2 times, while others after 3 times and so forth.

Figure 27 shows how this relationship would look like in the CAM. We see the two sockets instantiated with concepts *sun*, *planet* and *star* respectively.

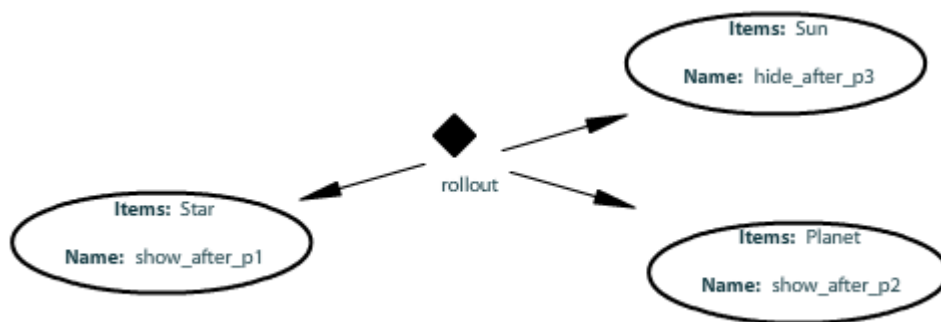


Figure 27 Showafter relationship

Similarly, we will have a relation called '*showatmost*' which hides resources (in the target socket) if the parent (in the source socket) has been shown more than the threshold, indicated by the '*showatmost*' label.

Note that we also need to make sure that for each concept a count of user accesses is kept in the user model. This can be done with a relationship '*countaccess*' relating a concept to itself. Figure 28 shows the '*countaccess*' relationship as visualised in the CAM component after dragging/selecting it from the CRT instance list; the CRT is not yet instantiated (no DM concepts appear in the Source/Target sockets)



Figure 28 'Countaccess' relationship

6.2.2 Depth First

The depth first strategy is used for sequential learners [66]. One topic at a time is presented, and the student is allowed to go in-depth (hence, the strategy name) in this topic first, before he proceeds with the next topic. Preferably, no menus are shown to such students and all they need to access is a ‘next’ button, taking them to their next study material, whether statically linked, or adaptively generated.

For the depth-first strategy, again, we assume the ‘*is_part_of*’ relationship between Domain Model concepts fulfils the parent-child relationship. The socket, which can contain any number of concepts, is suggestively called *concepts in hierarchy*. This CRT’s adaptive behaviour counterpart uses the relation from each concept to each of its children, called *Show next child XOR next sibling*. As shown in Figure 29, the logic in the constraint takes care of showing the appropriate next concept, either the next child or the next sibling, as visualised in the CAM component after dragging/selecting it from the CRT instance.

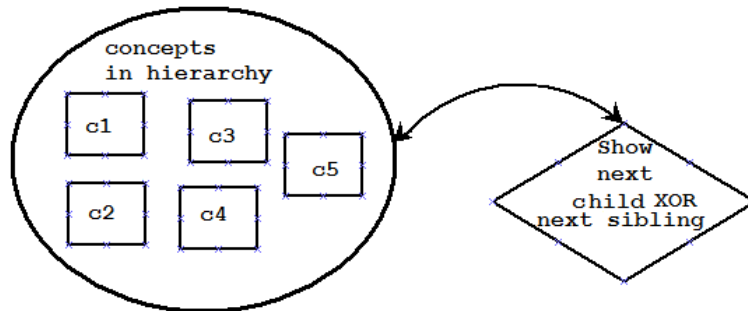


Figure 29 The main relation implementing the ‘Depth First’

The CRT condition must ensure that $_X$ is the child (or next sibling) of $_Y$ and thus needs to be displayed to the user. Moreover, the CRT needs to specify how to update the User Model variable that keeps track of the current position of the learner within the hierarchical course. The condition only shows the next sibling if the concept does not have any children left to be shown.

Finally, we add a starting concept to the start CRT, which shows the first concept unconditionally. This is a good example where, in principle, a domain relation can be translated directly into a CRT. Thus, the CAM component should be able to specify that,

for a given DM tree, each *parent* \rightarrow *first-child* relation, as well as each *finished-child*, *sibling* \rightarrow *next-sibling*, as well as each *finished-child*, *last-sibling* \rightarrow *next-parent* relations should all be transformed into prerequisite relations.

6.2.3 Breadth First

The breadth first strategy is used for global or holistic learners [66]. These learners like to see the global ‘picture’ first, before they dive into any topic. For such students, menus and other orientation devices are quite helpful.

Thus, implementation of this strategy has to start with the ordering of the concepts in a hierarchy. Normally, this would just reuse a parent-child DM relationship, such as the ‘*is_part_of*’ relationship, just as the depth-first strategy, only with a different processing mechanism. The socket, which can contain any number of concepts, is suggestively called *concepts in hierarchy*. This CRT’s adaptive behaviour counterpart uses the relation from each concept to each of its children, called *Show next sibling XOR next child*. Figure 30 shows how the logic in the constraint takes care of showing the appropriate next concept, either the next sibling, or the next child, as visualised in the CAM component after dragging/selecting it from the CRT component.

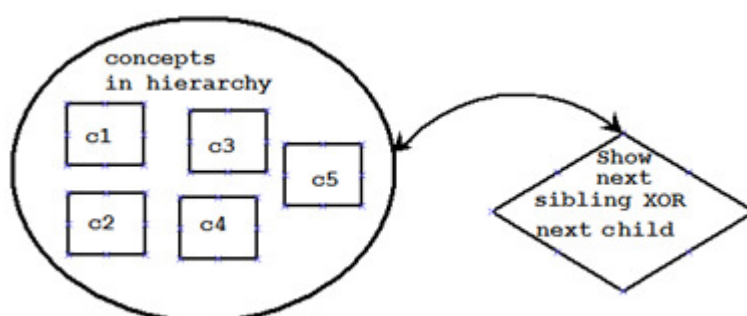


Figure 30 The main relation implementing the ‘Breadth First’

6.2.4 Visual – Verbal

Visual-verbal preference [66] corresponds to a strategy which does not need concept ordering. Students are shown visual material (graphs, pictures, video, flash, simulations) if they have a visual preference and verbal material (text, audio) if they have a verbal nature.

For visual-verbal we need only one relationship, a basic prerequisite relationship with a constraint like `_X.label == UM.preference`. One of the challenges is to define via one or more CRTs how this preference of the learner is detected. For this, we have several alternatives. We could design a CRT which expresses the fact that the learner will set his preferences via a menu. Alternatively, we could create a different CRT, which defines the collection of concepts which are always visible, and separate CRTs defining which of these always visible concepts can, by being selected by the user, change the current preference setup in the user model. Furthermore, the UM variables can be set via an initial questionnaire or test.

Here, beside the issue of the step in the authoring process where the labels are set (similarly to the Rollout discussion), the issue of creating a CRT, which specifies how the user input is to occur, appears. Most Adaptive Hypermedia systems only focus on showing existing concepts, based on interaction such as clicks, or perhaps scrolling and time on page. However, free user input is usually not catered for. To alleviate this issue when the user has to make a choice (such as in choosing between his verbal or visual preference), in some existing Adaptive Hypermedia systems a work-around with special concepts is used. These concepts represent the choices (thus, in our case, a concept for visual and a concept for verbal preference), and accessing these concepts (clicking on them) triggers the desired setting. Figure 31 illustrates the CAM component after dragging/selecting it from the CRT instance list; the CRT is instantiated with concepts Sun and Star.



Figure 31 Visual vs. verbal

6.2.5 Beginner – Intermediate – Advanced

The beginner-intermediate-advanced strategy is a frequently used strategy, dividing the available course material into three types: material aimed at beginners, material for intermediate students and material for advanced students. Typically, students that have been

identified as beginners are not allowed to see material from the higher levels till their status is changed (for example by reading all material in their level, or taking a test).

For the strategy implementation in the CAM component, we drag a relationship called *beginner intermediate advanced* onto the canvas. This relationship has three sockets, which can each contain an arbitrary number of concepts, and are suggestively named *beginner*, *intermediate* and *advanced*. The adaptation associated with the relationship shows all concepts in the beginner socket, but hides all concepts in the intermediate and advanced sockets first. It then shows the concepts in the intermediate sockets, if all concepts in the beginner socket have been seen, and displays all concepts in the advanced socket, only if all concepts in the intermediate socket have been seen. Note that this requires the sets of concepts to be strictly disjunctive. That is, a concept cannot be in the beginner and intermediate socket at the same time, as this would hide part one of the beginner concepts and create a deadlock. An alternative would be to specify additional conditions for the concepts, such as the fact that only resources with specific labels are allowed in a given socket. For instance, only resources in image format are allowed in the beginner socket and text format for intermediate socket. In this way, the same concepts could be dragged and dropped into different sockets, meaning that their different representations (corresponding to their different labels) can be shown when the different sockets are triggered. Figure 32, shows the visualisation in the CAM component after dragging/selecting it from the CRT instance list and the CRT is instantiated.

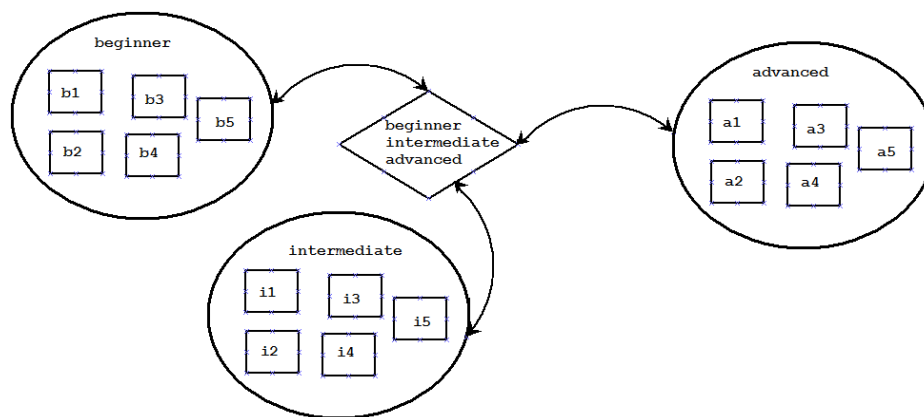


Figure 32 Beginner - intermediate - advanced

6.2.6 Relatedness

The Domain Model used by the CAM component can inherit multiple layers of relations. These relations can be further used in the adaptation process. For instance, an advanced student can be shown all related concepts, whereas a beginner student is only shown concepts within his own course.

In the new model, there is no explicit '*related*' relation. Instead, the Domain Model has *named* relations. Hence two concepts are related if they have any named relationship between them in the DM. These DM relations can be used in CRTs, as illustrated in the cases above. In the general case, any relation in the DM can be used in the CRT definitions, in order to associate an adaptive behaviour to that particular relation. Therefore we can achieve the relatedness relation with a unary CRT that accesses concepts from the DM related to the concepts in the socket. This is similar to the setup of the depth first CRT in Figure 29 and Figure 30, and thus the visual representation is not repeated.

6.2.7 Conclusions

In this section we have analysed various adaptation and pedagogical strategies and how the CAM component would be able to represent them and improve the access of teachers to such complex adaptation notions. In such a visual way, teachers can get to grips with the authoring tasks more easily, without knowledge of adaptation languages.

While trying to express the (selection of) learning style related strategies we noticed some common issues:

1. It is clear that we need to have some view of the Domain Model in order for the teacher to see what the available concepts are.
2. A wizard-like interface for ready-made strategies could be very helpful, while still allowing customisation.
3. The step-wise processing as previously implicitly assumed in LAOS/LAG based systems is still desirable. Otherwise some strategies like Breadth- and Depth-First will not be possible, as inference rules will make sure the whole content will

directly be visible. Thus, rules need to be triggered one-step-at-a-time, when certain events occur (for example, a mouse-click). It is envisioned that, if desired, it should be possible to specify rules that trigger other rules, like in AHA!.

However, in the CAM environment, this needs to be done in a visual way.

In the LAOS/ LAG conversions to AHA!, one could control to a certain extent what kind of menus and other guidance the student would get. This represents adaptation of the presentation layer in LAOS and reflects on interface changes and display for the student. It is desirable that in the new CAM-based systems this control will also be present.

With these conclusions, we proceeded to the design and implementation of the CAM component. First, we will visit the CAM model definition.

6.3 Design of the Concept Adaptation Model (CAM)

Model

The Concept Adaptation Model was introduced for the first time in [84]. In the meantime, after many discussions within the authoring community, some changes were deemed necessary and the model has been refined. In this section we will go into its design in more details. The CAM model is based upon lessons learnt from the AHAM [153] and LAOS [41] models, as well as being inspired by the multi-model, metadata-driven approach to content adaptation [6] and has incorporated ideas from other models, such as Dexter [79], XAHM [29], Munich [104], UWE [104], ADAPT [71]. The approach taken builds upon the ACCT. Similarly to the ACCT [50], [51], the GRAPPLE Authoring Tool (GAT), which is based on the CAM model, features a graphical user interface for specifying an adaptive course. The ACCT does not include an editor for pedagogic strategies and these strategies are therefore predefined. The GAT, however, features a dedicated component for specifying and modifying adaptive strategies. The section is organised as follows. First we investigate the shortcomings of AHAM or LAOS and give an overview of the CAM model and finally we indicate how the different layers in the CAM model interrelate.

6.3.1 CAM

In order to achieve the visual, graphical, flexible and user friendly way of authoring desired, the authoring framework needs to be even more general and flexible. In this section we consider its design in more details.

The CAM model contains an extensible number of layers, which may be different for each application. This is due to the fact that, for instance, for a specific application, page-presentation and quality-of-service parameters might need to be stored independently, if they are major components of that application, whilst for another one, the approach of storing them together (as in LAOS) is acceptable. Thus, application-dependent new layers should be allowed to be defined by authors. However the DM, UM and AM, layers are described as mandatory, as without these basic functionalities no adaptive system will function (DM for the underlying domain information, UM to describe the user and the AM to describe the adaptation rules).

There will always be a DM and UM layer and at least one layer with adaptation aspects, so the structure of the CAM model is a generalisation of the AHAM model and either equivalent to, or a generalisation of the more refined LAOS model. It is important to make the distinction between the three different types of relations between concepts: hierarchical, semantic and pedagogic relations. The first two types relate to the specific domain knowledge, while the third type refers to the pedagogical strategy used. In the CAM model relations exist both at the DM level and at the CRT level. Relations at the DM level should enhance the domain knowledge but not relate to pedagogy, while the converse is true for the CRT relations. This strict division can greatly improve the re-usability of both the domain content and the adaptation strategies in the CRTs. To summarise the CAM model consists of the following layers:

- *Domain Model*, similar to the LAOS DM. The DM in the CAM model only allows domain-specific information (concepts, links and domain-specific metadata), like in LAOS, but unlike in AHAM.
- *User Model* similar to the one in AHAM and LAOS.

- A number (at least 1) of *adaptation models*, extending the ideas of generality and specificity: generic adaptation rules by means of CRTs, similar to AHAM and LAOS, can be applied to specific concepts in the adaptation model. Each type of relationship provides a different layer, or view on the course.

Thus, the authoring tool will comprise a DM authoring component, a UM authoring component, a CRT authoring component and a CAM authoring component. The presence of the CRT ensures that the complexity of authoring specification is hidden in the definition of these concept relationships, whilst the presence of the CAM ensures a high-level, non-programming access to the authoring behaviour specification. In this chapter, due to its important role in integrating all the other authoring elements, the focus is on the *CAM model* and *GAT tool*.

The relationships defined in the different CAM layers do not yet express the actual adaptation that will take place. A prerequisite may be translated to a rule that will change the presentation of links to concepts, but it may also be translated to the conditional inclusion of a prerequisite explanation (fragment).

6.3.2 Link with LMSs

So far in this chapter, we have implicitly assumed authors are creating stand alone AH courses. However one of the aims of the GRAPPLE project is to connect AH with major LMSs. To achieve this goal, an author has to be able to create an AH course, which will run inside the LMS. The GRAPPLE project has defined a system architecture and various components to enable this [125]. This has a few implications for the authoring process. It is possible to author an AH course in isolation and then run it inside an LMS. An author may however want to take advantage of the LMS. This can be done, by inserting concepts in a CAM with an *external location*, rather than from a DM. This external location can for example be a test on an LMS. Another way to take advantage of the integrated set-up is to share part of the user model between different courses. The architecture enables authors to inspect what variables are currently in use in the user model, and this enables them to use existing variables in their own courses.

6.3.3 Relations between the Layers of the CAM model

The Domain Model describes the domain concepts, their attributes and their semantic relationships. These relationships do not attach any adaptive behaviour but merely indicate a semantic link. In the CAM model, concepts from the DM model can be selected and the adaptive behaviour can be attached to these concepts. For a concept to appear in the final adaptive lesson, it needs thus to be selected in the CAM.

The CAM model only puts together the final picture of the adaptive behaviour, much like the pieces in a jigsaw. The types of possible adaptive behaviour are defined in the CRT library created by the CRT model. The CAM can link a number of domain concepts using a certain CRT and therefore concretely establish the adaptive behaviour defined in the CRT in a more generic way. CRTs dragged into the CAM model can be then populated with domain concepts from the DM model, as long as the particular CRT allows for that particular type of concept.

6.3.4 Relations between the CAM and the Adaptation Engine

The CAM model groups domain descriptions and adaptation descriptions into adaptation strategies or adaptive story lines. These story lines need to be exported to an adaptation engine. This engine can be the GRAPPLE adaptation engine developed in the GRAPPLE project or any other engine that can with work or has converters for one or more of the exported adaptation languages. The CAM model thus will export several languages of various levels: the CAM XML language, a combination of CAM, CRT and DM language, or the adaptation engine's own GRAPPLE Adaptation Language [142] (see also section 2.5.3). It is theoretically possible to export to other adaptation languages, such as the LAG language [43], [48].

6.3.5 Model Verification

The authoring process (for the concept structures and the adaptation) which is focused on the creation of concept relationships, may appear thus to be simple. Using different layers for different CRTs can potentially make the understanding of the conceptual structure relatively easy, too. However, this simplicity is partly an illusion. Depending on how the

concept relationships are translated (using a translation model) to the low level adaptation rules for the adaptation engine, the (graph-like) structure of concept relationships of a single layer may already cause problems and the combination of concept relationships from different layers may cause more issues. We illustrate this with some examples.

Consider a simple structure where *A is a prerequisite for B*, *B is a prerequisite for C* and *C is a prerequisite for A*. This may cause a problem or not, depending on how prerequisites are used in the learning application.

- When “A is a prerequisite for B” results in links to B being recommended only after learning enough about A it is possible that the cycle of prerequisites causes deadlock: the links to A, B and C will never be recommended to the learner.
- When “A is a prerequisite for B” means that a short explanation of A will automatically be inserted into a page about B to compensate for the missing prerequisite knowledge then there is no problem. If A is accessed first it contains a prerequisite explanation of C, possibly preceded by an explanation of B.

Problems with undesirable structures like cycles are relatively easy to detect within a single layer. These problems become much more unpredictable when looking at the adaptation rules that result from translating the concept relationships from all layers. The most common types of problems are those of *termination* and *confluence*. *Termination* is the problem where contradicting rules may continue to send the learner through the same sequence without ever reaching a final state in the course.

Confluence is the problem of unpredictability, because it cannot accurately be predicted which rule will be executed first (which layer will gain priority), and the outcome (experience for the learner) may differ between two different learners even if they perform exactly the same actions.

Authoring adaptive behaviour using graphs makes the authoring process easier, but also increases the chance of authors introducing potential problems, for example termination and confluence as above. It is not feasible to do a full verification of the graphs in the

CAM. The reason behind this is that the graphs are conditional, so static structure analysis alone will not be sufficient. In general, verification would imply simulation of all possible paths. In the scope of the CAM component (the part of the authoring tool that allows specifying the CAM) it has been decided to do a partial verification. CRTs will define how many and what type of entities need to be connected to it. The CAM component will verify this when the author tries to make the connections. It will also warn the author about cycle structures in the graphs in the CAM, but will not prevent these as conditions in the CRTs may mean that in practice there is no problem (there are no execution paths containing a cycle).

6.3.6 CRT and DM Combination Problems

If any concepts are allowed for any CRT relation, it is possible that the adaptation may not work properly due to termination and confluence issues, as described in the previous section. We envision the following new set of issues that can arise and allow for the corresponding corrective actions, as follows.

- **Wrong arity in CRT:** For instance, if a certain prerequisite needs only one concept on the left side and one on the right, but multiple ones are dragged into the left (or right) side, this should trigger an error.
- **Wrong type in CRT:** Similarly, if a certain type of metadata is expected in the placeholder of a CRT (such as `_Y.type=artist`) and the concept dragged is not of that particular type, an error should be triggered and the drag and drop should not be allowed.
- **Wrong CRT combination:** for instance, when two perfectly acceptable CRTs are combined, but when the result is not acceptable (for example, when the right side of the first CRT is not compatible with the left side of the following CRT). In this case, again, an error should be triggered and the combination should not be allowed. If the combination is possible, but results in an enhanced set of constraints, the DM concepts dragged into the overlapping placeholder would have to respect both sets of constraints.

The design of the CAM component helps in tackling these problems. With the tool, the author will be able to show only CRTs of the same type at once, creating different ‘layers’ for the different types of CRTs. This however does not prevent unintended effects between different CRTs. A way to tackle this would be to implement priorities among the different ‘layers’.

6.4 Design of the Adaptation Languages

The different adaptation languages have been designed by the GRAPPLE Work Package 3 team. We at The University of Warwick have been heavily involved in collaborative discussions around all the languages, as well as implementing the CAM component, the web services and the Shell, which brings the different components together. The DM and CRT [5] components have been implemented by Giuntlabs and The University of Graz respectively. To describe adaptive behaviour in the scope of the CAM, a language format is needed to describe each of the three main components, the DM, CRT and the CAM. The languages will have to deal both with the visual as well as the semantic aspects of the model. In this section we will describe each of the adaptation languages. The languages share a common header. This particular setup is standard for such applications [56], [41], and is essential only because the different components, DM, CRT and CAM need to have a common reference point and be able to exchange information. At a later stage in this process, two other extensions would need to be integrated, one for supporting Simulations, and the other one for supporting Virtual Reality enhanced learning. An example of the header is shown below.

```
<header>
  <modeluuid>99863eb7alc2</modeluuid>
  <modeltype>CAM</modeltype>
  <authoruuid>27214759-3a23-41d5-84f8-bddd419483de</authoruuid>
  <authorisation>readwrite</authorisation>
  <creationtime>124225</creationtime>
  <updatetime>124291</updatetime>
  <title>example-lesson</title>
  <description>example</description>
</header>
```

The example below is the header for the CAM example in section 4.3. The header contains the following information:

- A unique identifier, the *modeluuid*,
- The type of model (DM, CRT or CAM), in the *modeltype* tag.
- The authorisation tag - *none*, *read* or *readwrite* - indicating what privileges authors other than the original author have. The original author always has all privileges and this is not listed.
- The creation time and update time, time-stamps indicating when the model was created and last updated.
- A title and a description.

6.4.1 DM Language

The DM language is a language that describes the conceptual domain or subject matter graph. The DM has to pass the Domain Models with related concepts, relationships and related attributes to the Conceptual Adaptation Model. The following attributes describe a Domain Model.

- **Name:** name or title of the Domain Model
- **CreationDate:** the date and time the Domain Model was created
- **Author:** identifier of the author that created the Domain Model
- **Description:** a description of the scope of the Domain Model
- **Keywords:** words or phrases that identify the Domain Model conceptually
- **Identifier:** a unique identifier (uuid) for the Domain Model. The identifier is important for storage, retrieval and sharing of the Domain Model.
- **Concepts:** Concepts are individual conceptual entities. The division of the whole conceptual domain into entities should be done along logical lines and would be relatively intuitive for a teacher to define. These concepts can then have content connected to them.
 - **Name:** the name or title of a Concept
 - **Identifier:** a unique identifier (uuid) for the concept

- **Description:** a description of the scope of the concept
- **Keywords:** words or phrases that identify the concept conceptually
- **Resources:** the content, connected to the concept.
 - **Name:** the name of single resource
 - **Location:** indicates the path (URI) of the resource, if available.
 - **UsageType:** any resource can be used in a different context such as introduction, image, body text, conclusion,
 - **Standard LOM metadata:** standard LOM [98] metadata is available for any resource
- **Relationships:** binary relationships between concepts, allow the authors to create a logical taxonomy of the subject domain. The relationships are binary and directed, because this most closely represents hyperlinks in hypermedia. Furthermore everything that can be expressed with undirected or n-ary relationships can also be expressed with a number of directed binary relationships. Directed binary relationships have as added benefit that they are easy to understand for an author. The relationships in the DM language do not have a specified behaviour, as this is decided later in the model. As the DM is not an ontology and does not use any reasoning related to the student adaptation, there is no inference made based upon these relationships. This was a design decision that aimed at introducing the separation of concerns in a unequivocal fashion in the authoring tool and authoring language representations. However, this relationship can be used later on in various ways, including extensions of the DM towards simple reasoning within it. Moreover, in the CRT component, it can be specified that a certain CRT is (often) used for concepts that are involved in a certain relationship in the DM. Thus, for instance, a part-of relation can be connected to a prerequisite behaviour via a CRT. Furthermore, a CRT can also access the concepts that are linked to a concept by a specific relationship, by explicitly mentioning this relationship in its code.
 - **Name:** name of the relationship.

- **Concept1:** The first concept involved in the relationship
- **Concept2:** The second concept involved in the relationship

Any Domain Model has to be composed of at least one concept and the concepts can be connected by directed binary relationships between the concepts. In the DM a concept cannot be connected to itself via a relationship.

The author can add new attributes (custom attributes)Domain Model, Concepts, Resources and Relationships to the Domain Model.

The conceptual representation required for the graphical authoring that we want to achieve is not easily overlaid on most standards, described in Section 2.1 (for example, LOM, SCORM), because of a rigid composition of elements and structures. As we have seen in section 2.1.6, IMS-VDEX was designed for expressing and exchanging a simple set of human language terms in a machine readable way. IMS-VDEX [102], [65] provides the right amount of freedom of conceptual content representation that is suitable for our graphical authoring of content for adaptation. Therefore the DM language [84] is based on the IMS-VDEX standard. Choosing an existing standard for the Domain Model increases the use of standards in AH and can contribute to solving the issue of the lack of standards described in section 1.1.

The DM language also specifies the location of the physical resources which represent the actual content. The DM format specifies which resources the concepts link to, directly, or via metadata and conditions, resolved by the engine at runtime.

The DM component uses a visual representation of the DM XML format, the IMS-VDEX based format, in order for the author to be able to create and edit Domain Models. The author is not meant to deal with the XML format directly. The DM format is included in the CAM format in the *domainModel* tag. The elements of the IMS-VDEX specification are matched to the attributes required by a DM language, described above, as follows.

- The *vocabulary* part of the IMS-VDEX specification corresponds to the *Domain Model*.
- The *term* tag is used to model the *concepts*.
- The *relationship* tag directly matches with the *relationships* between concepts.

Below an example shows a simple Domain Model in the DM Language. The DM is called *Astronomy* and contains the concept *Jupiter*, with the possible content alternatives *Jupiter Picture* and *Jupiter Description* each with a URL linking to the content. The DM also contains the concept *Solar System* and using the *relationship* tags, it is indicated that the *Solar System* is composed of *Jupiter*.

```
<vdex orderSignificant="true" profileType="hierarchicalTokenTerms"
  language="en" xsi:schemaLocation="http://www.imsglobal.org/xsd/imsvdex_vlp0
  imsvdex_vlp0.xsd" xmlns="http://www.imsglobal.org/xsd/imsvdex_vlp0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <vocabName><langstring language="en">Astronomy</langstring></vocabName>
  <vocabIdentifier>http://www.giuntlabs.it/currmap.xml</vocabIdentifier>
  <term>
    <termIdentifier>500</termIdentifier>
    <caption><langstring language="en">Jupiter</langstring></caption>
    <description>
      <langstring language="en">Largest planet in milkiway</langstring>
    </description>
    <? possible concept attributes introduced by the author?>
    <sequence>
      <content>
        <caption>
          <langstring language="en">Jupiter picture</langstring>
        </caption>
        <location>href='http://tinyurl.com/dave3u'</location>
        <label> Picture </label>
        <? resource attributes/metadata?>
      </content>
      <content>
        <caption>
          <langstring language="en">Jupiter description</langstring>
        </caption>
        <location>href="/jupiter_spots.html'</location>
        <label>Body text </label>
        <? resource attributes/metadata?>
      </content>
    </sequence>
  </term>
</vdex>
```

```

    </content>
  </sequence>
</term>
<term>
  <termIdentifier>501</termIdentifier>
  <caption>
    <langstring language="en">Solar System</langstring>
  </caption>
  <? attributes and resources?>
</term>
<relationship>
  <sourceTerm>501</sourceTerm>
  <targetTerm>500</targetTerm>
  <relationshipType>is-composed-by</relationshipType>
</relationship>
</vdex>

```

6.4.2 CRT Language

As described before, the CAM model makes a clear distinction between adaptation-based and content-based relations. The latter are part of the domain knowledge, whereas the former can be pedagogical relations which define a teaching strategy. Thus, the CRT is the language used to describe the pedagogical relations. The CRT language [4] is not meant to be processed by humans by hand, but is a description language of the adaptive behaviour, which is encapsulated and represented by a graphical symbol in the CAM environment. The CRT language is included in the *crtModel* tag of the CAM language, and is also an XML-based language, like the DM language. This design decision was taken due to the portability of XML-based languages, as all these tool-components have not only to communicate with each other, but also to communicate to delivery tools, via a dedicated *event bus* (collection of web services, which allows arbitrary GRAPPLE components to communicate) and web services. The CRT language also contains snippets of behaviour, included in a system independent adaptation language, for example, GAL [142]. The adaptation language description is not written by lay person authors, but by persons combining programming knowledge with pedagogical knowledge. For the scope of the CAM component and language, the number and type of entities that connect to a CRT are important, as well as any restriction on combinations. Moreover, the metadata describing

the CRT behaviour in layman's terms is vital for the non-programmer author to be able to use CRTs efficiently in the CAM component. Below an example of a prerequisite CRT is shown, together with explanations of each of the elements. The main parts of the representation are:

- (1) General
- (2) Adaptive Behaviour
- (3) Constraints
- (4) Sockets
 - a. Directed
 - b. Undirected
- (5) Associated DM Relations

6.4.2.1 General

The example starts with some general information. This information is stored in the common header, described previously in section 6.4.

The header contains the title and description of the CRT, the time it was created (*creationtime*) and last updated (*updatetime*), and the id of the author (*authoruuid*). It also contains the *authorisation*, the value of this can be either *nopermissions*, *read*, or *readwrite*, indicating respectively that only the original author may access the CRT, all other authors may only read the CRT, or everyone can edit the CRT.

```
<model>
  <header>
    <title>prerequisite</title>
    <description>this is a prerequisite CRT </description>
    <creationtime>1253890007331</creationtime>
    <updatetime>1253890007331</updatetime>
    <modeluuid>cf5de7f5-12b5-4720-92e5-736cac59985b</modeluuid>
    <authoruuid>dfa9b95c-f26f-4883-855e-a886f76f27a</authoruuid>
    <authorisation>readwrite</authorisation>
    <modeltype>crt</modeltype>
  </header>
```

In the GRAPPLE project, extensions are defined for authoring of Virtual Reality and Simulations. After the header the CRT continues in the body. The header can contain a *comment*, explaining the CRT in pedagogical terms, After this the *crt dialect* is set, indicating either *crt*, *vr crt*, *sim crt*, to indicate whether this is a normal CRT or whether it is a CRT for use with any of the extensions.

Next it can contain optional *parameter* tags. Parameters are used in the CRT, in the code, but get instantiated in the CAM, when the author instantiates a CRT. The parameters have a *name*, *type* and *default value*. An example use can be found in the rollout strategy in section 6.2.1. A parameter could indicate after how many times the concepts in a socket should be shown and the concrete number can be changed when the author uses a CRT in the CAM, eliminating the need for a CRT for every possible value.

Next the CRT describes a default visual representation, containing a colour in a hexadecimal representation and a shape, the name of a Flex drawing class, which can be changed when a CRT is used in a CAM.

```
<body>
  <crt>
    <comment/>
    <crt dialect>CRT</crt dialect>
    <parameter name="" type="" />
    <visualrepresentation>
      <colour>#000000</colour>
      <shape>diamond</shape>
    </visualrepresentation>
    ...
  </crt>
</body>
```

6.4.2.2 Adaptive Behaviour

The *adaptivebehaviour* tag is the place where the adaptation code is kept. There are two tags, *code*, containing the code, and *usermodel*. The code can be of type GAL [145]. It could also contain other code for the purpose of flexibility of formats. The *usermodel* tag

indicates which user model variables are in use. For each variable, the *name*, the *socket* they are used in and whether they are publicly available and can be stored in the GRAPPLE systems are recorded. There is a *location* for the variable, in case it concerns a remote course on a LMS, and the variable has a type, range and default value.

```
<adaptationbehaviour>
  <code type="gale"><![CDATA[]]></code>
  <usermodel>
    <umvariable>
      <umvarname>knowledge</umvarname>
      <socket>source</socket>
      <public>true</public>
      <persistent>true</persistent>
      <location>
        <web/>
        <remotecourse>
          <remotecoursename/>
          <resource>
            <resourceuniqueid/>
            <resourcename/>
          </resource>
        </remotecourse>
      </location>
      <type>integer</type>
      <range>
        <from>0</from>
        <to>100</to>
      </range>
      <default><![CDATA[0]]></default>
    </umvariable>
  </usermodel>
</adaptationbehaviour>
```

6.4.2.3 Constraints

In the general case it is not possible to completely prevent termination and confluence problems with CRTs without making assumptions and restrictions. To prevent confluence problems we envisioned a layered system, with priorities attached to each layer. To prevent termination problems, in the general case any kind of loops, where a CRT connects to itself via a series of other CRTs, would need to be disallowed. This is very restrictive, hence we

allow loops, unless otherwise stated and note that it is up to the CRT authors to ensure this does not cause problems.

In the *constraints* part of the CRT, the *allowedinloop* tag indicates whether such loops are allowed. Furthermore we introduce constraints based on the attributes of the CRT. In this example we see that concepts in the socket with id cf5de7f5-12b5-4720-92e5-zzzzzzzzzzz need to have a User Model, an attribute called *language*, and that its *value* should be English. At deploy time, the authoring tool can check whether these attribute constraints hold for all concepts, as well as whether the CRTs which participate in loops are allowed to do so.

```
<constraints>
  <allowedinloop>false</allowedinloop>
  <attributeconstraints>
    <attrconstraint>
      <socketid>cf5de7f5-12b5-4720-92e5-zzzzzzzzzzz</socketid>
      <attributename>language</attributename>
      <requiredvalue>en</requiredvalue>
    </attrconstraint>
  </attributeconstraints>
</constraints>
```

6.4.2.4 Sockets

Each CRT has at least one socket. Sockets are containers which can contain a number of concepts when instantiated in the CAM. Each *socket* has a *type*, *name* and identifier. The minimum and maximum number of concepts it can contain are indicated in the *mincardinality* and *maxcardinality* tags.

```
<crtsockets>
  <socket type="source">
    <uuid>cf5de7f5-12b5-4720-92e5-zzzzzzzzzzz</uuid>
    <name>beginner</name>
    <mincardinality>1</mincardinality>
    <maxcardinality>*</maxcardinality>
  </socket>
  <socket type="target">
    <uuid>cf5de7f5-12b5-4720-92e5-pppppppppppp</uuid>
    <name>advanced</name>
```

```

        <mincardinality>1</mincardinality>
        <maxcardinality>*</maxcardinality>
    </socket>
</crtsockets>

```

6.4.2.5 Associated DM Relations

Relations between concepts in the Domain Model are conceptual, regardless of teaching strategy chosen. However an author of an adaptive course may want to use a certain teaching strategy for each of the occurrences of a relation in the DM. Therefore in the CRT, a CRT author can indicate that this CRT, this teaching strategy, is a good candidate to be used with a certain DM relationship. The CAM component will feature an interface, which would allow the author to get suggestions about suitable CRTs to use based on this information.

```

<model>
  <head> ... </head>
  ...
  <crt>
    <associateddmrelations>
      <relationshiptype>is_composed_of</relationshiptype>
    </associateddmrelations>
  </crt>
</body>
</model>

```

6.4.3 CAM Languages

Initially, we considered that one language to represent the course would be enough, but in reality this has to be decomposed based on the target user and role of the language.

Currently, the CAM language consists of three components:

- (1) The visual description, or *CAM-visual language*, in which the author defines the course graphically;
- (2) The XML format which is used by the authoring tool internally for storage and editing of CAM models;
- (3) The *CAM-external language*, the XML-format that is used for export and interfacing with the adaptation delivery engine.

6.4.3.1 CAM Visual Language

The visual representation is meant for authors to create adaptive story lines and expresses the CAM instance in a visual way. This, together with the visual representation of the DM, are the only languages that are intended for the non-programmer author to work with. This language is essential for ensuring the lowest possible threshold in authoring of adaptation.

The CAM languages are all based upon the well known concept of graphs. Hence, the main components are nodes and links. More specifically, the nodes are concepts from a DM instance and the links are the relationships from the CRT instance. There is an additional grouping operator as well.

To summarise, we have the following components that make up the graphical representation:

1. *DM Concepts*, as in the DM, with default representation a list of items belonging to a socket, represented as an ellipse:



Figure 33 A socket with one concept

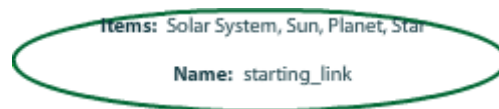


Figure 34 A socket with multiple concepts

2. *CRT instances*, with a graphical based representation and a number of anchor hooks locations, which sockets can be hooked into. The default representation is shown in Figure 35:



Figure 35 A CRT

There are provisions for changing colours and shapes, described in the CAM internal language, which is described in the next section.

6.4.3.2 CAM Internal Language

The CAM internal language is the main format for storage and manipulation of the CAM component. It is designed to be used by other authoring systems, while they interface with the CAM component. For instance, when inserting information from the DM XML language about a domain concept into the CAM model, this CAM internal language is used. It represents thus incomplete lessons in progress. Prior experience with the LAG language [43] shows that non-programmer authors prefer not to be at all involved at the level as described by this language. On the other hand, programmers or authors with a Computer Science background prefer the more ‘hands-on’ experience. Thus, for the latter authors only, the CAM XML language could be used directly to describe adaptive behaviour. Also, prior research showed that an XML-based language is preferable, as it is both more portable and perceived as easier to manipulate than a pure programming language, as in the development of the LAG-XLS language [136]. As the LAG-XLS language was only aimed at learning styles, however, and we wish to adopt a wider scope of adaptivity, a new language had to be created.

The CAM format can be used to describe not only direct CRT relations, but also more global information and the flow of a course, as follows.

- *Start* and *end* states can be described with a special CRT. The CRT is different only in that the GAL [142] code simply states “start” or “end”.
- *Main concepts* of a course, useful for visualising the course, can be indicated in the name or metadata of the sockets.

- *Learning goals and objectives* can be derived from the selection of start and end states and main concepts.

Below we illustrate the new CAM language via an example. We see this type of CAM output as XML descriptions of groups of concepts and *named typed* relations between them. For example, if concept A is to be seen before concept B, a CAM would be:

```
<model>
  <header> .. </header>
<body>
  <cam>
    <camInternal>
      <domainModel>..</domainModel>
      <crtModel> .. </crtModel>
      <crt>
        <uuid>cf5de7f5-12b5-4720-92e5-736cac59985b</uuid>
        <shape>diamond</shape>
        <colour>#C0C0C0</colour>
        <parameter name="p1" type="integer">1</parameter>
      <camSocket>
        <uuid>e9b45bd0-6013-11de-8a39-0800200c9a66</uuid>
        <socketId>cf5de7f5-12b5-4720-92e5-zzzzzzzzz</socketId>
        <position><x>100</x><y>250</y></position>
        <size>10</size>
        <shape>rectangle</shape>
      <colour>#006633</colour>
      <entity><dmId>201-de-8a39-0800200c9a66</dmId></entity>
    </camSocket>
    <camSocket>
      <caption>target</caption>
      <uuid>f539bae0-6013-11de-8a39-0800200c9a66</uuid>
      <socketId>2b5-4720-92e5-pppppppppppp</socketId>
      <position><x>100</x><y>400</y></position>
      <size>10</size>
      <shape>rectangle</shape>
      <colour>#006633</colour>
      <entity><dmId>11de-8a39-0800200c9a66</dmId></entity>
    </camSocket>
  </crt>
</camInternal>
</cam>
</body>
</model>
```


As we can see above, the CAM language consists of two parts: the *header* and the *body*. The header is the common header for all authoring tool components, as described in section 6.3.1. The *body* part contains the information specific for the CAM model. The body part contains the following information.

- The *domainModels* and *crtModels* in use, as described in sections 4.1 and 4.2
- A number of *CRT* tags, containing the instantiation of CRTs from the *crtModel* with actual concepts from the *domainModel*.

The *CRT* tag contains the following information.

- A unique identifier the *uuid*.
- The *shape* and *colour* of the CRT, for displaying the CRT in the CAM component, this is optional, defaults will be used if the tags are omitted.
- There can be a number of parameters, which are optional, and a way to instantiate some CRT values in the CAM. For example in a rollout strategy, see section 6.2.1, the number of times after which a certain concept is shown can be a parameter, instantiated when the author uses the CRT in the CAM.
- An optional *position* element. The position of a CRT needs to be given if the CRT is not binary (1 or >2 sockets). For binary CRTs the position of the graphical connecting relation arrow or link is calculated, based on the position of the sockets.
- A number of *camSockets*.

The *camSockets* contain the instantiation with the actual Domain Model concepts. They are represented via the following information.

- A *caption*, a *name* for the socket, for both the author's information, as well as to label the domain concepts which are pasted into this socket (for example, if the label says 'beginners', then the concepts are all labelled beginners, and the behaviour associated will be that of beginner level concepts). By placing concepts in multiple sockets, either belonging to the same CRT or to different ones; it is possible to assign to them multiple labels and for them to be involved in multiple adaptation strategies.
- A unique identifier, the *uuid* for the socket.

- A *socketId*, the unique identifier of the specific socket in the *crtModel*.
- A *position*, the position where the socket should be displayed in the CAM component.
- An optional *shape* and *colour*, the shape and colour a CAM socket should have; a default will be used if the *shape* and *colour* tags are omitted.
- A number of *entity* tags, pointing to entities as described below.

The entity contains the instantiation with the actual concepts, and they contain the following information.

- Zero or one *dmID*, the ID of the concept in the Domain Model that is assigned to the entityID.
- Zero or more *labels* for the resource of a concept and an optional location for a resource. These are for identifying which resources will be bound in the final package into the concept, and thus into the socket. They are either given directly, via a location, or indirectly, via labels that are to be used by the adaptation engine to insert the resources at runtime. This setup was considered most flexible for the adaptation engine, as it allows for resources to be updated later in the process, as long as the overall conceptual structure is constant. This also allows thus for a very high level of dynamic and adaptive behaviour, which is not completely scripted (as is in previous models such as AHAM and LAOS). For compatibility however, the direct binding of resources is still allowed.
- A number of *relationshipType* elements, where a requirement for a concept to be involved in a certain relationship in the DM model can be expressed (for example, the entity should have participated in an IS-A relation). This is useful if no domain concept is given, and instead, the domain concepts involved have to be retrieved via some description – in this case, via the relations they participate in the Domain Model. Again, this is added for the sake of flexibility. The author is not obliged, to have to specify directly and concretely the resources involved, or even the concept involved. Thus, domain concepts involved in a socket can also be specified indirectly. In the following, some graphical information about the display of the

entity can also be specified, as follows. This information is optional and can be left out, in which case the default values from the CRT should be used.

- The relative *position* in the socket instance.
- The *size* of the entity in the socket.
- The *shape* of the entity in the CAM instance.
- The *image* of the entity in the CAM instance.
- The *colour* of the entity in the CAM instance.

The description of the subject domain and behaviour semantics of the CRT called ‘prerequisite’ needs to be separately imported from the DM and CRT repositories. These descriptions are then included in the *domainModel* and *crtModel* parts. The complete XML Schema specification of the CAM Internal Language definition can be found in Appendix II.

6.4.3.3 CAM External Language

The CAM external language is the portable format for the output for the GRAPPLE Adaptation Tool. In essence, it is a scaled down version of the CAM-internal language, with only the information relevant for delivery. Therefore it does not have any information for visualising the CAM model, but it does have all other information about the components present from the various models: CAM, CRT and DM. The schema can be found in Appendix IV.

6.5 Design of the GRAPPLE Authoring Tool Shell and Web Services

The different models (DM, CRT and CAM) each have an authoring tool for authoring, bound together via a Shell tool. The goal of the GAT authoring environment is ultimately to make it possible for authors (teachers) to create adaptive courses. In order to present a coherent approach and not to frustrate the authors with an array of different tools it is very important that the GAT offers the tool components under one umbrella. In the implementation, we have decided to take this one step further and to create one completely integrated tool. For an author there is only one tool, the GAT, which offers DM, CRT and

CAM editing functionalities. The fact that the models and tool components have been developed in parallel by different teams is completely transparent to the user.

The file *Settings.xml* contains the address of the web services used by the GAT, as well as the names of the models, CRT, CAM, DM. There are two web services, one called *GrappleAuthoring* with the methods, *loadModel*, *saveModel*, *getModels*, *logInUser* and *registerUser* for loading and retrieving models and registering and logging in users.

In the GRAPPLE architecture, communication between components, such as the GAT and the delivery engine, is done via a so called event bus. The event bus is a web service, which allows arbitrary (GRAPPLE) components to communicate. Answers to requests come in the form of web service calls. However, in general it is not possible to turn a user's browser into such a web service, which can accept answers to requests. Therefore a proxy is needed. This proxy is the second web service and is called *eventEventListenerService*. It has the methods *callGebByProxy* and *eventEventListenerOperation* for accepting answers from the event bus.

In the future, new models can (and will³⁹) be added, by performing the following simple steps:

- adding their name in the *Settings.xml* file and
- making sure that this name is used in their *modeltype* tag in the header,
- ensuring that there is a folder of the appropriate name in the modules folder in the trunk and
- that there is an <name>.mxml file in that folder containing the model, and
- ensuring that a new model should extend the same class and react to the same events as other models.

³⁹ In a future step of the GRAPPLE project, integration with the Virtual Reality tool and the Simulation tool, built by other partners in the same work package, are planned.

A guide for tool implementers for integrating their tool with the GAT shell can be found in Appendix V. The shell uses the *dockable flex*⁴⁰ library and the *birdseye ravis*⁴¹ library. The main functionality it offers is a fully functional toolbar. Implementers, implementing tool components, are expected to extend *closablevbox* and react to events.

6.6 Design of the CAM Component

The CAM component is a component in which authors are able to specify CAM instance models thus, CAM is a *meta-modelling system*). The author is able to define all layers of their specific CAM model with this tool. The tool consists of three components corresponding to the mandatory layers: DM editing; CRT editing and CAM editing, held together via an integration component. The tool is implemented as a web-based application and was developed in Adobe Flex. It uses the GAT shell, which contains general infrastructure for manipulating windows, opening/saving models and the toolbar, which interacts with each component. In Figure 36 we see a screenshot of the authoring tool. While normally, the different tool components are seen in different tabs, they can be moved to floating windows. In the screenshot, the three main components are visible in floating windows, from left to right the DM component, CRT component and CAM component. In the DM component we see a graph, with several nodes and links. The nodes represent the concepts in the subject domain and the links represent semantic relationships, without any attached behaviour. Authors can create conceptual Domain Models, by inserting the concepts and their links in the CRT component.

The CRT component gives the opportunity to edit the adaptive behaviour of a CRT. Authors specify a title and description of the CRT. Many authors with less technical skill will use the CRT component mainly to look at the descriptions and the allowed User Model variables in use by a CRT. Authors can, if they wish to, customise a CRT or create one

⁴⁰ <http://code.google.com/p/dockableflex/>

⁴¹ <http://code.google.com/p/birdeye/>

from scratch, specify the number of sockets, create a list of User Model values that can be used in the code and write the actual adaptive behaviour in GAL code [142]. The CAM component instantiates CRTs, which originally have socket placeholders, with actual domain concepts, via the equivalent of a drag and drop functionality between the DM component and the CAM component. In the screenshot we see the nodes, which represent sockets and the links. The links with a diamond label represent the CRTs. The grey round labels indicate that sockets share concepts. This is useful for displaying possible transitivity. This appears for instance in a prerequisite type of relation, where if we represent a CRT prerequisite between concepts A and B, and then a CRT prerequisite between concepts B and C, then it is useful to have a connection showing that both CRTs share concept B, and thus processing and adaptive behaviour may be linked. Such visualisation is useful for authors also to see possible errors in their planning – for example, if CRTs were not supposed to share concepts, than it is immediately obvious that an error occurred. Authors can insert CRTs from the list of existing CRTs and copy concepts from a Domain Model to a DM component window to instantiate the CRTs.

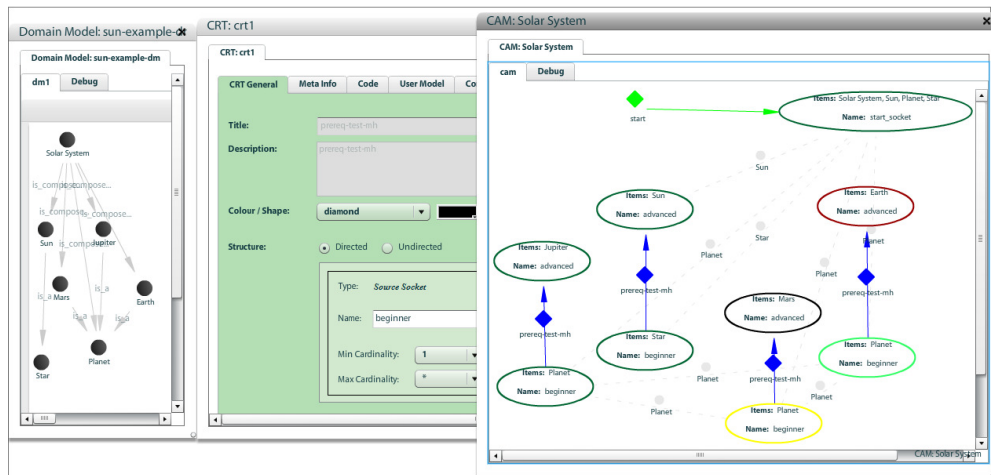


Figure 36 Authoring tool

After the implementation, the tool has been evaluated with several groups of users, based on different criteria. The evaluations are presented in the next section.

6.7 Evaluation of GRAPPLE Authoring Tool

This section describes the evaluations of the authoring tool. The aim of these evaluations is twofold. First of all the aim of the evaluations is to verify whether the tool can be used to

specify courses and adaptive strategies and especially whether technically experienced authors can do this successfully. Secondly the aim of the evaluations is to identify areas of the tool that can be improved, in order to inform the directions for future development, and to identify what should be done to get closer to our goal of a tool that allows technically experienced authors to successfully specify courses.

In order to establish whether technically experienced teachers can easily create courses using our tool, we conducted a quantitative evaluation. For this the evaluation objective is broken down into a number of hypotheses. A questionnaire with closed questions is used and statistical analysis of these questions indicates whether the hypothesis should be accepted or not. The questionnaire, hypothesis and results are discussed in section 6.7.1. In order to identify areas of the tool that need to be improved to get closer to our goal of a tool that allows technically experienced authors to successfully specify courses, we conducted a qualitative evaluation. This evaluation consisted of two separate evaluations. First of all, feedback was gathered from attendees of the ICALT 2009 conference, in July 2009, and of the ECTEL 2009 conference, in October 2009. The feedback was collected from about 12 lecturers and researchers in the areas of technology enhanced learning, at a paper presentation and at two demonstration sessions respectively. This resulted in a list of suggested improvements. Secondly an online survey with a number of open questions was conducted in collaboration with Work Package 9 partners in the GRAPPLE project [75]. It was filled in by the same respondents as the questionnaire with closed questions mentioned above. The results were processed in a more formal way. The answers were assigned to categories and counted, to show what is seen by the respondents as the main directions of future improvements. The questionnaire and analysis of the results are discussed in section 6.7.3.

For both the open and closed questions of the online survey we had 10 responses from researchers internal to the GRAPPLE project and 11 responses from researchers external to the project. Of the internal respondents 2 described themselves as Software Engineers, 2 as Researchers and 5 as (PhD) students; the other 2 participants did not specify their

profession. The age of the external respondents ranged from 23 to 34 with the average age being just over 27. 5 of the external respondents were male, 4 female and 2 did not specify their gender. Of the respondents 3 indicate never to use learning environments, 3 indicate to use them less often than once a week and 3 indicate to use them several times a week. Two respondents indicate to have no experience with learning environments, 3 to have about a year's worth of experience and 4 have several years of experience using learning environments. One respondent indicated to be an expert on authoring, while 3 respondents had no experience at all, 4 had some experience and 1 had advanced experience. Of the internal participants 5 were male, 3 female and 2 did not specify. The age ranged from 27 to 46. The average age of over 33, is higher than that of the external participants. 5 respondents described themselves as researchers in the area of TEL, 2 as teachers, 2 as (PhD) students and one as a Computer Scientist. 4 respondents never use learning environments, 3 use them several times a week and 1 respondent indicate to use learning environments on a daily basis. 6 respondents indicate to have several years' experience with learning environments, while one more respondents indicates to have several months experience and finally one respondent has no experience at all. 2 respondents have no authoring experience, 3 have some experience and another 3 are expert authors. Full details about the background of our respondents can be found in appendix VII.II.

6.7.1 Quantitative Evaluation

The formal evaluation was conducted as an online survey. The background of the respondents is as discussed above. The aim of this part of the evaluation is to verify whether the tool can be used to specify courses and adaptive strategies and especially whether technically experienced authors can do this successfully.

The ideal tool, which allows technically experienced authors to successfully specify courses easily, will be simple, well integrated and consistent. The tool itself should not be experienced as cumbersome. If the tool allows authors to easily specify courses, it is reasonable to assume that authors would want to use the tool again, would like to use it

frequently and would recommend it to other authors. This implies that they accept the tool and the ideas behind it. The learning process of the tool will also have to be manageable. So authors will not have to learn a lot and will do so easily and quickly. Afterwards, when authoring, the authors will not require a lot of support, and are confident to use the tool. These observations lead to the following hypotheses.

H_{sus1a} Respondents would like to use the tool frequently.

H_{sus2a} The tool is not perceived as complex.

H_{sus3a} The tool is easy to learn.

H_{sus4a} Respondents do not require a lot of support.

H_{sus5a} The tool is well integrated.

H_{sus6a} The tool is not inconsistent.

H_{sus7a} Respondents learn to use the tool quickly.

H_{sus8a} The tool is not cumbersome.

H_{sus9a} Respondents are confident to use the tool.

H_{sus10a} Respondents do not need to learn a lot to use the tool.

H_{UA1a} Respondents would like to use this system in the future.

H_{UA2a} Respondents would recommend this system to their colleagues.

H_{acc-a} The respondents accept the ideas, concepts and functionality of the system.

In order to be able to identify areas of improvement for the tool, it is useful to know where respondents agree that in fact the opposite of the hypotheses mentioned above is true. Therefore we will also be investigating the following hypotheses.

H_{sus1b} Respondents would not like to use the tool frequently.

H_{sus2b} The tool is perceived as complex.

H_{sus3b} The tool is not easy to learn.

H_{sus4b} Respondents do require a lot of support.

H_{sus5b} The tool is not well integrated.

H_{sus6b} The tool is inconsistent.

H_{sus7b} Respondents do not learn to use the tool quickly.

H_{sus8b} The tool is cumbersome.

H_{sus9b} Respondents are not confident to use the tool.

H_{sus10b} Respondents need to learn a lot to use the tool.

H_{UA1b} Respondents would not like to use this system in the future.

H_{UA2b} Respondents would not recommend this system to their colleagues.

H_{acc-b} The respondents accept the ideas, concepts and functionality of the system.

Based on these hypotheses, a questionnaire was designed, consisting of 12 closed questions. Note that as usability is an important aspect in the identified hypotheses, the first 10 questions were the questions taken from the standard SUS (System Usability Scale, [19]) questionnaire for system usability. The questions asked were the following. Answers could be given ranging from 1 to 5 on a Likert⁴² scale, with 1 representing *Strongly disagree*, 5 representing *Strongly agree*.

- SUS1. I think that I would like to use this system frequently.
- SUS2. I found the system unnecessarily complex.
- SUS3. I thought the system was easy to use.
- SUS4. I think that I would need the support of a technical person to be able to use this system.
- SUS5. I found the various functions in this system were well integrated.
- SUS6. I thought there was too much inconsistency in this system.
- SUS7. I would imagine that most people would learn to use this system very quickly.
- SUS8. I found the system very cumbersome to use.
- SUS9. I felt very confident using the system.
- SUS10. I needed to learn a lot of things before I could get going with this system.
- UA1. I would like to use this system in the future.
- UA2. I would recommend this system to my colleagues.

⁴² See <http://www.socialresearchmethods.net/kb/scallik.php>

6.7.2 Results of Quantitative Evaluation

The average usability score from the SUS questionnaire is 47.98. Given the possible range of 0 to 100, this outcome indicates a medium usability, or even slightly below that. The external participants were more positive (average score 53.41) than those involved in other parts of the GRAPPLE project (average score 42.00). Interestingly, this difference is statistically significant ($T=2.3$; $p<0.05$). This could be because respondents internal to the project have overall higher expectations than people outside the project. The different background could also play a role. The internal respondents are on average slightly older and more experienced researchers, academics and teachers.

The average results of the SUS questions are shown in Figure 37. The figure lists the scores for internal and external respondents side by side. An ideal system would show a perfect star shape in this type of diagram, with peaks at SUS questions 1, 3, 5, 7, 9 and dips at questions 2, 4, 6, 8 and 10, due to the fact that the SUS questionnaire alternates questions formulated in a positive way, with questions formulated in a negative way, to even out bias. We can see that, the results of the internal respondents almost show a shifted star shape, with the peaks and dips in the wrong places. This indicates internal respondents face serious usability problems. The only exception seems to be that they do in fact see the tool as well integrated. The results gathered from the external respondents are again slightly more positive, though also here we do not see the expected star shape. We see that external respondents want to use the tool frequently, find it easy to learn, find the tool very well integrated and consistent and learn the tool quickly. However they also find it complex, need a lot of support, find it cumbersome, and are not very confident when using the tool and need to learn a lot. This indicates that, while external respondents are a bit more positive about the tool, they too experience serious usability problems.

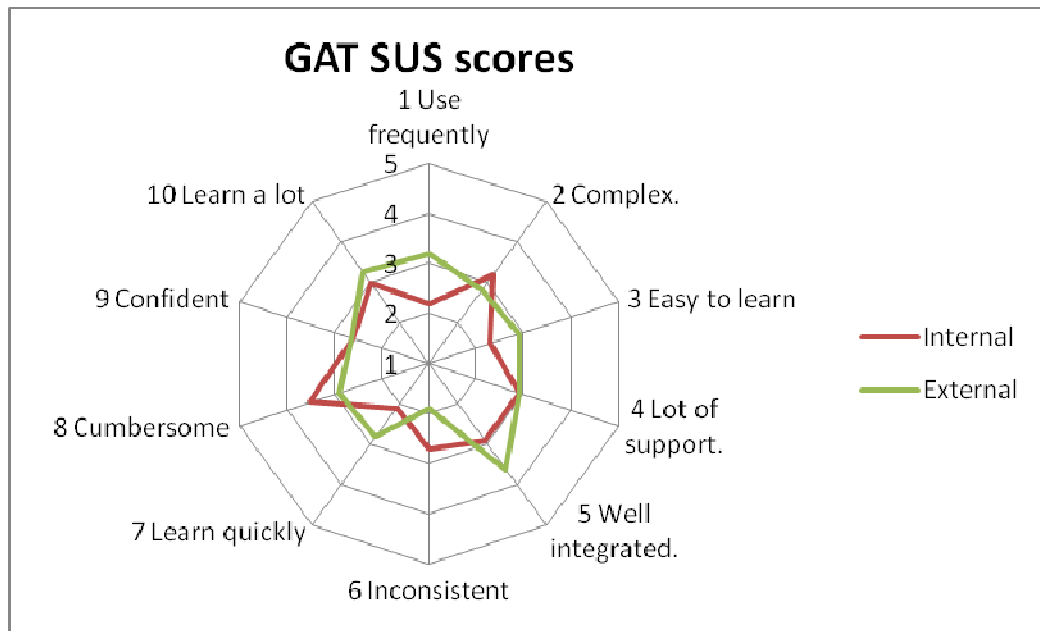


Figure 37 SUS score GAT

In order to verify our hypotheses, we now analyse these results statistically. We will both establish whether the hypotheses hold for internal respondents, whether they hold for external respondents and whether there is a significant difference in the answers given by both groups.

In order to calculate SUS scores, the answers are normalised, unlike the raw results as presented in Figure 37. For the questions with a positive bias, the normalised value is equal to (the answer -1), whereas for the questions with a negative bias, the normalised value is equal to (5 – the answer). For the analysis of our hypotheses we use the normalised scores. For the questions UA1, UA2 we also used the normalised data. Thus, in order for a hypothesis to be confirmed, the mean of the related normalised closed question needs to be above 2, so the mean difference compared to 2 should be ≥ 0 , on a normalised 0 to 4 scale. An additional requirement is that this result should not have been caused by chance. To verify this, we use a one-sample, one tailed T-test against an expected population mean of 2, for both the internal and external respondents. The alternate hypothesis for this test is ‘the mean in our data is equal to 2’. With a one tailed test we can be sure, that a rejection of the null hypothesis is caused by a deviation of the results on one side. This is the most suitable test, as we want our hypothesis to be rejected when the mean is significantly less

than 2, but do not mind if it is significantly greater, as this just indicates a stronger confirmation of what we wanted to evaluate. So the null hypothesis is '*the mean in our data is smaller than 2*'. Hence all alternative hypotheses are our 'a-hypotheses' (hypotheses $H_{sus1a} - H_{acc-a}$) and the null hypothesis are the 'b-hypotheses' (hypotheses $H_{sus1b} - H_{acc-b}$). This test works under a number of assumptions.

- The observations are independent from one another.
 - Our evaluation setup consisted in contacting people directly with the web-address of the tool, user guide and questionnaire. It is therefore reasonable to assume that the observations are independent.
- The sample was drawn from a normal population.
 - It is reasonable to assume our data fits these criteria, as the external participants have been randomly selected from researchers in the area, and the internal participants have been randomly selected from researchers internal to the project it is fair to assume the sample is representative and normally distributed.

In Table 5 and Table 6 for both internal and external respondents, we see the T value and the degrees of freedom (DF) and the mean difference, calculated as the difference between the mean of the respondents' answer and the neutral value of 2. Finally we see the P value. We assume a confidence of 95%. If the probability is $P < 0.05$, the null hypotheses, our 'b-hypotheses' (hypotheses $H_{sus1b} - H_{acc-b}$), must be rejected and our alternate hypotheses, the 'a-hypotheses' (hypotheses $H_{sus1a} - H_{acc-a}$), are likely to be true.

It would seem that we should reject the null hypothesis for H_{sus1b} , H_{sus3b} , H_{sus7b} , H_{sus8b} . This would seem to indicate that H_{sus1a} , H_{sus3a} , H_{sus7a} , H_{sus8a} are likely to be true. So that means that the internal respondents would like to use the tool frequently, find it easy to learn, learn the tool quickly, find it not cumbersome, would recommend it to others and would like to use it again in the future. In the SUS diagram suggests we can see that this is however by a very small margin. Because of the small sample size, individuals who feel strongly have a relatively large impact on the SUS diagram. Our use of one-tailed T-test means that the results would suggest that we cannot discredit the null hypotheses H_{sus2b} , H_{sus4b} , H_{sus5b} ,

H_{sus6b} , H_{sus9b} , H_{sus10b} , H_{UA1b} , H_{UA2b} , and H_{acc-b} . So this would suggest the internal respondents find the tool complex, need a lot of support, not very well integrated, inconsistent, are not confident in using it and need to learn a lot. They also would not like to use it again, would not recommend it and do not accept its functionality and ideas at present.

Table 5 GAT hypotheses results internal respondents

Hypothesis	T	Df	Mean Diff	P
H_{sus1b}	-2.228	9	-0.800	0.026
H_{sus2b}	-0.688	9	-0.200	0.254
H_{sus3b}	-2.689	9	-0.700	0.012
H_{sus4b}	0.198	9	0.100	0.423
H_{sus5b}	-0.361	9	-0.100	0.363
H_{sus6b}	0.896	9	0.300	0.197
H_{sus7b}	-3.857	9	-0.900	0.002
H_{sus8b}	-3.000	9	-0.500	0.007
H_{sus9b}	-1.309	9	-0.400	0.111
H_{sus10b}	0.000	9	0.000	0.500
H_{UA1b}	0.818	9	0.300	0.217
H_{UA2b}	-0.612	9	-0.200	0.277
H_{acc-b}	0.183	9	0.050	0.429

An important caveat is that the sample size is very small. We have no data about which work packages the respondents come from. The respondents were selected by asking for volunteers. So it is possible that projects members in work packages that rely upon the tool were more motivated to fill in the questionnaire than those in other work packages. So it is

doubtful whether these results are transferable to the whole population of GRAPPLE participants. Also by asking for volunteers we may have influenced the results as people who feel strongly, either for or against particular questions surrounding the tool, are more likely to voice their opinion than those for whom its usability is just ok, but not exceptional. What we can see though, that the internal respondents have issues with the user experience, and have problems learning to use the tool. Both example courses, and improved user guide and indeed an improved user experience are clearly needed.

Table 6 GAT hypotheses results external respondents

Hypothesis	T	Df	Mean Diff	P
H _{sus1b}	0.803	10	0.181	0.220
H _{sus2b}	0.614	10	0.181	0.276
H _{sus3b}	-0.430	10	-0.090	0.338
H _{sus4b}	0.232	10	0.090	0.410
H _{sus5b}	3.130	10	0.636	0.005
H _{sus6b}	3.464	10	1.090	0.003
H _{sus7b}	-0.614	10	-0.181	0.276
H _{sus8b}	0.430	10	0.090	0.338
H _{sus9b}	-1.174	10	-0.363	0.133
H _{sus10b}	-.607	10	-0.272	0.278
H _{UA1b}	1.336	10	0.454	0.105
H _{UA2b}	2.667	10	0.727	0.012
H _{acc-b}	2.283	10	0.636	0.023

The external respondents seemed more positive about the tool, looking at the SUS diagram. In the table above, we can see that it would seem that external respondents accept H_{sus5a}, H_{sus6a}, H_{UA2a} and H_{acc-a}. This would suggest that the external respondents find the tool well integrated and consistent, would recommend the tool, and would imply user acceptance of

the tool and functionality. The results also suggest that the null hypotheses must be accepted for H_{sus1b} , H_{sus2b} , H_{sus4b} , H_{sus7b} , H_{sus8b} , H_{sus9b} , H_{sus10b} and H_{UA1b} hence that external respondents would not like to use the tool frequently, find it complex, difficult to learn, need a lot of support, do not learn it quickly, find it cumbersome, are not confident when using it, need to learn a lot and would not like to use it again. These results are more in line with the SUS diagram, with the exception of whether or not they want to use the tool frequently. However a caveat here is that the sample size is very small. The sample size is also drawn from a small pool of conference attendees, who are already interested in this area of research. If we look at the background of the external respondents we see that they are relatively young and inexperienced and will have little experience as a teacher. Thus the results may not directly translate to the wider population of technically experienced teachers. We again see however that there are problems with the usability of the tool as well as its learning curve.

We have seen in the beginning of this section, that the average SUS scores for external respondents are significantly higher than for internal respondents. We will now investigate whether this is also the case for each of the hypothesis individually. Since the samples are not taken from the same subjects we cannot use a paired T-test. Instead we must use an independent samples T-test⁴³. This test relies upon the assumption, that the variances of the samples are equal or close enough, whether this is the case can be determined by using Levene's test⁴⁴. If the P value of Leven's test < 0.05 we cannot assume equal variance and have to use a different test. We see that this is not the case for any of our hypotheses. If the results obtained from the internal and external respondents are significantly different, the P value should be < 0.05 . Otherwise we cannot disprove our null hypothesis and there is no significant difference. From Table 7 we can see that the results for H_{sus1b} , H_{sus5b} and H_{sus8b} are significantly different. External respondents want to use the tool significantly more frequently, find it significantly better integrated and significantly less cumbersome.

⁴³ See <http://comp.uark.edu/~whlevine/psyc2013/t-test.html>

⁴⁴ See <http://www.itl.nist.gov/div898/handbook/eda/section3/eda35a.htm>

Table 7 Difference between internal and external respondents

Hypothesis	Levene's Test		T	Df	Mean Diff	P
	F	P				
H _{sus1b}	1.444	0.244	2.359	19	0.981	0.029
H _{sus2b}	0.062	0.806	0.917	19	0.381	0.370
H _{sus3b}	0.344	0.565	1.831	19	0.609	0.083
H _{sus4b}	0.801	0.382	-0.014	19	-0.009	0.989
H _{sus5b}	0.007	0.932	2.171	19	0.736	0.043
H _{sus6b}	0.100	0.756	1.721	19	0.790	0.101
H _{sus7b}	2.429	0.136	1.879	19	0.718	0.076
H _{sus8b}	0.001	0.978	2.166	19	0.590	0.043
H _{sus9b}	0.014	0.908	0.083	19	0.036	0.934
H _{sus10b}	0.418	0.526	-0.452	19	-0.272	0.656
H _{UA1b}	0.025	0.875	0.309	19	0.154	0.760
H _{UA2b}	0.322	0.577	-1.118	19	-0.472	0.277
H _{acc-b}	0.021	0.887	-0.801	19	-0.313	0.433

6.7.3 Qualitative Evaluations

The main aim of the qualitative evaluations is to identify areas of the tool that can be improved, in order to inform the directions for future development, and to identify what should be done to get closer to our goal of a tool that allows technically experienced authors to successfully specify courses. In addition to this we also seek information to further verify whether the tool can be used by technically experienced authors to successfully create courses.

The qualitative evaluation was conducted in two steps, first of all informal feedback was gathered from attendees of the ICALT 2009 conference, in July 2009, and of the ECTEL 2009 conference, in October 2009 as described in the introduction of section 6.7. Secondly,

an online survey was used. As described in the introduction of section 6.7, the survey was filled in by the same group of respondents as the qualitative survey.

Our main hypothesis for qualitative feedback from the evaluation at demonstration sessions at the ICALT 2009 and ECTEL 2009 conferences and for the qualitative evaluation of the online survey is:

- H_{qual} *Users are able to understand how the GAT authoring tool works, find the interface usable and agree with the ideas behind the tool.*

In sections 6.7.3.1 and 6.7.3.2 we briefly describe the setups and results for both evaluation steps and draw conclusions with regard to our main hypothesis, as well as further sub-hypotheses. In section 6.7.4 we present the results of the qualitative evaluations and in section 6.7.5 we discuss the findings based on these results.

6.7.3.1 Informal Qualitative Evaluation at Demonstration Sessions

The informal evaluation at demonstration sessions took place at the ECTEL 2009 and ICALT 2009 conferences, respectively. The feedback was gathered by demonstrating the tool to various interested conference participants and peer researchers, with positions such as PhD students, researchers and professors, and asking them to try and use the tool. In section 6.7.4 we summarise the feedback gathered at these sessions.

For a more comprehensive analysis of the qualitative feedback from these evaluations, we can further divide H_{qual} into the following sub-hypotheses.

- $H_{\text{inf-shell}}$ *Users see the tool as one tool: thus, the Shell tool integrates the different components well.*
- $H_{\text{inf-cam}}$ *Users are able to understand how the CAM component of the GAT tool works (a), find the interface usable (b) and agree with the ideas behind the component (c).*
- $H_{\text{inf-dm}}$ *Users are able to understand how the DM component of the GAT tool works (a), find the interface usable (b) and agree with the ideas behind the component (c).*

- $H_{\text{inf-crt}}$ *Users are able to understand how the CRT component of the GAT tool works (a), find the interface usable (b) and agree with the ideas behind the component (c).*

6.7.3.2 Formal Qualitative Evaluation at Online Survey

For the online survey we gathered 21 computer scientists, researchers and PhD students from the field of Technology Enhanced Learning. As mentioned above, this evaluation was conducted together with Work Package 9 partners in the GRAPPLE project. Respondents were asked to answer the following open-ended questions:

UFQQ1: What did you like best about the system and authoring tool?

UFQQ2: What did you like least about the system and authoring tool?

UFQQ3: What should be improved and how?

The overall sample can be divided into two groups:

- Participants involved in (other parts of⁴⁵) the GRAPPLE project.
- External participants.

Ten participants involved in (other parts of) the GRAPPLE project took the survey. The participants were recruited via the mailing list of the whole consortium. In addition to this, external participants were also recruited at the ECTEL demonstration and tutorial sessions on the authoring tool. It was considered useful to compare a group who had no prior knowledge of GRAPPLE ideas and tools, such as the external participants, with a group who had a broad idea, but no prior experience with the tool, such as the GRAPPLE project participants who were however not involved in the Authoring tools design and implementation. Out of 20 invited external researchers in the area of Technology Enhanced Learning, 11 filled in the online survey.

The material used for the online survey consists of the *authoring tool set* and the online *user guide* available through the help button.

⁴⁵ These participants were outside of Work Package 3, the Authoring work package, but still in the GRAPPLE project.

For the qualitative feedback from this evaluation we can divide H_{qual} into the following sub-hypotheses.

- H_{visual} *Users appreciate the visual interface for authoring.*
- $H_{\text{functional}}$ *Users like the functionality the tool offers.*
- $H_{\text{understandable}}$ *The tool is easy to understand.*
- H_{usable} *The tool is not too difficult to use.*

Thus, we have divided the part of H_{qual} stating that ‘users agree with the ideas behind the tool’ into two hypotheses, $H_{\text{functional}}$, dealing with the ideas reflected in the functionality, and $H_{\text{interface}}$, dealing with the ideas specifically expressed via the visual interface. The other two hypotheses, $H_{\text{understandable}}$ and H_{usable} , correspond to versions (a) and (b) of the previous division of sub-hypotheses in section 6.7.3.1, respectively.

6.7.4 Results of qualitative evaluations

In this section we first present the informal feedback gathered, followed by the results of the online survey.

6.7.4.1 Informal qualitative feedback

Feedback on the GAT/Shell tool in general:

- The shell nicely integrates the different tool components, to the extent that all participants in our demonstration sessions considered the tool to be one single tool rather than a collection of different tools.
- Some low-level debugging and usability comments were also registered, indicating the following.
 - The Escape key should also exit the selection mode alongside CTRL+L.
 - It would be nice to be able to put a DM and CAM side by side and drag and drop concepts across.
 - The functionality of copying concepts via right clicking on a selection is desirable.

- The naming of the tool components seems confusing; many people asked: What are '*DM, CRT, CAM*'? As an explanation, *domain, relation and course* seemed to be understood by most.

Feedback concerning the DM component:

- People using the tool at our demonstration sessions appreciated the DM component interface and were able to model a conceptual domain for the final course.
- The DM component does not offer a way to create content, one can only link to existing HTML pages.
- Importing outside domains into the DM component would be very beneficial, for example from ontologies from outside the tool, from IMS-LD or from other standards. This would also increase reusability.
- Some low-level debugging and usability comments were also registered the following.
 - When selecting, some nodes sometimes turn red, without apparent reason.
 - The functionality to right click on a concept and press copy, such as in the CAM, is desirable.
 - When you click on a concept and go to insert relation, the concept clicked should be selected as source, by default (there should be no need for an extra step in the specification). It would also be nice to have the inserting of relations in the list of functionalities under the right mouse button.
 - It would be useful to be able to click on a concept name to edit it.

Feedback concerning the CAM component:

- People using the tool at our demonstration sessions appreciated the CAM component interface and were able to model a conceptual domain, with some help and explanation of the underlying theory.

- Some low-level debugging and usability comments were also registered the following.
 - All links appear directed, even the undirected CRTs.
 - There may be issues with the scalability of the CAM view, when the course becomes large and contains hundreds of concepts, the view becomes cluttered. Introducing hierarchies or levels was suggested.

Feedback concerning the CRT component:

- From informal tool use and feedback gathered at demonstration sessions we noticed that the concept of CRTs is considerably more difficult to grasp than the concept of the DM and the CAM, the name itself also seems to lead to confusion.
- Some low-level debugging and usability comments were also registered, as follows:
 - The naming of the options ‘public for LMS’ and ‘Public for GRAPPLE’ in the UM variables part is confusing, a better naming convention has to be found⁴⁶. This has already been changed to ‘public’ and ‘persistent’. Further experiments need to show if this is easier to understand.
 - The list of sockets sometimes displays XML rather than the names of the socket. This has been fixed in the new version.
 - The title of a model does not appear in the tab.

6.7.4.2 Formal Qualitative Evaluation Results

As said, the questionnaire was filled in by 21 people, 11 external to the GRAPPLE project and 10 involved in other parts of the GRAPPLE project. Respondents were asked the following open questions.

- UFQQ1: What did you like best about the system and authoring tool?
- UFQQ2: What did you like least about the system and authoring tool?
- UFQQ3: What should be improved and how?

⁴⁶ These were introduced as they are needed by the adaptation engine implementation team, and the LMS integrating team.

The collected qualitative data has been content-analysed in order to establish answer categories and frequencies which reflect the gathered data. For aggregating and synthesising the open answers and establishing categories for them, the approach used and outlined in [150] was applied. This approach is based on and inspired by the principles of the content analytical research tradition of grounded theory [139]. At this stage, gathering qualitative data is very valuable, as it will guide further developments of the tools and enable diagnosing unexpected issues. However, specifically in combination with the questions asked, a quantitative analysis would not result in meaningful information.

Below we give a summary of the answers given by the participants, the number of times a respondent gave a similar answer (the frequency) and the answer statements given. This summary only shows answer categories that appear more than once. The full data can be found in Appendix VI. The aim of this more formal qualitative feedback was to quickly discover problems with both the ideas and concepts behind the tool as well as the usability of the tool.

UFQQ1: What did you like best about the system and authoring tool?

Respondents external to GRAPPLE

Answer Category	Frequency
usability/interface	5
Help	2

Respondents involved in (other parts of) the GRAPPLE project

Answer Category	Frequency
visual representation – DM	5
interface – tabs	2
Functionality	2

UFQQ2: What did you like least about the system and authoring tool?

Respondents external to GRAPPLE

Answer Category	Frequency
complexity	2

Respondents involved in (other parts of) the GRAPPLE project

Answer Category	Frequency
CRT concept	3
multiplicity of tabs	3
unexpected reactions	2
authoring approach as a whole	2
DM: add relationship	2
CAM: filling sockets	2
complexity of tools	2
adaptation unclear	2
no testing capability	2

UFQQ3: What should be improved and how?

Respondents external to GRAPPLE

Answer Category	Frequency
provide example	3

Respondents involved in other parts of the GRAPPLE project

Answer Category	Frequency
improve help	3
provide predefined examples	2
add testing capability	2
coherency/consistency	2
enhance graphical representation of CAM	2
improve usability of operations	2

6.7.5 Qualitative Evaluation Findings

6.7.5.1 Informal Evaluation at Demonstration Sessions

The qualitative feedback on the GAT/shell is mainly in support of sub-hypothesis $H_{\text{inf-shell}}$, however some interface issues still have to be fixed. For example users cannot put different models side by side, and drag between them. It is important to note that this is merely an indication, as the feedback only consists of informal comments of a small group of volunteers.

The qualitative feedback on the DM component is mainly in support of sub-hypotheses $H_{\text{inf-dm}}$ (a) and (c). However, clearly, some small interface issues still have to be fixed, as presented above, thus $H_{\text{inf-dm}}$ (b) is only partially confirmed. It was however clear that the concept of the DM is easily understood, and authors are able to create a DM with the tool successfully. It is important to note that this is merely an indication, as the feedback only consists of informal comments of a small group of volunteers.

The feedback on the CAM shows that it is possible for authors to create a CAM using the tool. This is considerably more difficult than creating a DM and authors do require a lot more help and explanation. From this feedback we cannot conclude whether $H_{\text{inf-cam}}$ (a) and (c) are supported. As it stands, creating a CAM is too complicated. Moreover, some bugs however remain and have to be fixed, and thus $H_{\text{inf-cam}}$ (b) cannot be currently confirmed.

The feedback on the CRT component shows that the concept of a CRT is difficult to understand, and authors are having real trouble understanding the concept of a CRT. This also means that it is very difficult for them to create CRTs. The gathered feedback therefore cannot support sub-hypothesis $H_{\text{inf-crt}}(\text{a})$. This also influences the acceptance of hypotheses $H_{\text{inf-crt}}(\text{b})$ and $H_{\text{inf-crt}}(\text{c})$, which cannot be confirmed. This evaluation shows that the CRT tool is not usable for a technically experienced author. Changing the name CRT from '*concept relationship type*', which sounds somewhat abstract to many teachers and instructional designers, to, for example, '*type of pedagogical relationship*' may help.

However it is likely that a combined approach will work best, improving the GAT/SHELL along the lines suggested. Hiding the more complicated authoring functionality, when the GAT starts up will help novice users, while the advanced authors, who will want to modify or create CRTs, will know how to do this.

It is important to note that these conclusions are drawn from informal feedback of a small group and can only be an indication. However in terms of our evaluation objectives, we have been able to get an indication that authors can successfully author DMs, but struggle a bit with CAMs and have major difficulties authoring CRTs. This would indicate that currently technically experienced authors cannot successfully author courses using the tool. We have identified a number of improvements in relation to the second objective. Apart from lower level debugging information, we have seen that the help needs to be made clearer, real life examples should be created and the overall user experience should be simplified for novice authors.

6.7.5.2 Formal Qualitative Evaluation

It is important to note that while the online survey has been processed in a formal manner, the results are only an indication. Results are categorised and counted, to provide a more formal approach. The nature of the questions however, means that statistical analysis cannot be applied to prove or disprove our hypotheses. For example, where respondents did

not mention the help as most or least favourite, we cannot draw any conclusions about how much they appreciated the help.

The most liked features of the system among external respondents are the user interface, the tools usability, and the help functionality. Internal respondents liked the visual representation of the tool, especially when editing DMs best. This would seem to indicate in the direction of accepting H_{visual} . It would also suggest that for external respondents H_{usable} would be accepted. Two people mentioned the functionality as their most liked feature. However looking at their actual comments, one is quite critical of how well the system works, while the other comments on a very narrow part of the functionality, the possibility to copy and paste concepts. Therefore we cannot comment on whether $H_{\text{functional}}$ is likely to hold.

The external respondents mainly have issues with the overall complexity, although they did not specify in more detail which issues they faced. This seems to point to issues with the steep learning curve of the tools. Internal respondents mainly faced difficulties understanding the concept of CRTs and found that there was a multiplicity of tabs and acronyms, which confused them. This is an important issue that needs to be addressed. Looking at the feedback, we can see that the concept of a CRT is difficult to understand. However the naming, using various acronyms makes it more difficult to understand the tool. Also the abundance of open tabs and the absence of clear cut instructions make the tool more difficult to understand. Hence future development should focus on a simpler authoring interface for novice users, using a clearer naming scheme and featuring clear cut instructions.

Additionally, some unexpected system reactions were also mentioned. These unexpected reactions were caused by bugs, some of them also listed, for the various tools involved, in section 6.7.4. Resolving the bugs does however have a large impact on the systems usability as a whole and future developments should tackle these bugs.

It is important to mention that some respondents questioned the authoring approach as a whole, the unclearness of adaptation specification, and the fact that there was no testing capability to check whether a course, once built, actually works. At the time of evaluation it was not yet possible to deploy a course and see it in action, this should be addressed in the future versions, so that authors can check whether something they created actually works. Better examples and a more extensive help could improve the attitudes towards the approach and clearness of the adaptation.

The feedback about what respondents liked least seems to suggest we need to, reject hypotheses $H_{\text{understandable}}$ and H_{usable} .

The formal open question directly asked respondents what should be improved and how. The most prominent suggestions are:

- Improving help and documentation. Some respondents suggested, of interactive and tool-component specific help or tooltips.
- Improving available predefined examples.
- Adding the ability to test (deploy) courses.
- The usability of the tool.
- The graphical representation of the CAM.

In terms of our evaluation objectives, we have been able to get an indication that H_{visual} can be accepted and $H_{\text{understandable}}$ and H_{usable} rejected. Based on the feedback we cannot establish whether authors actually liked the functionality, whether $H_{\text{functional}}$ is likely to hold. Therefore this would indicate that the tool as a whole at present cannot be used by technically experienced authors to successfully create courses. We also identified a number of improvements that show which direction future development should take. We have seen that the help and documentation needs to be improved, usable examples need to be added, the ability to deploy an test courses need to be added and the graphical representation of the CAM should be improved. The naming of the different parts of the tool confuses authors,

as well as the abundance of tabs. The authoring interface should be simplified and novice users should, at first, not be confused with advanced editing.

6.7.6 Summary of Evaluation Findings

In the evaluations in sections 6.7.3 and 6.7.5 we have investigated whether tools can be developed that allow the specification of an adaptive course including the adaptation strategies and whether technically experienced authors can successfully create courses and adaptation strategies with these tools. This relates to our fourth research question in section 1.2.1. We have also identified the directions future work should take to improve the tools and make a step towards the goal of providing a tool, with which technically experienced authors can successfully create courses and adaptive strategies.

As we have seen in section 6.7.2, it seems that respondents internal to the GRAPPLE project look likely to want to use the tool frequently, find it easy to learn, learn the tool quickly and find it not cumbersome, would recommend the tool to other authors, would like to use it again and generally accept the tool. However the SUS diagram suggests that, these hypotheses have only been accepted by a very small margin. It is important to note that the small sample size was very small and authors did not have much training. The data would also suggest that internal respondents did find the tool complex and require a lot of support. They find the tool not very well integrated and inconsistent and are not confident when using the tool. They also have to learn a lot to be able to use the tool and would not like to use it again, would not recommend it and do not accept its functionality and ideas at present.

External respondents seem to find the tool well integrated and consistent, would recommend the tool, and it seems to accept the tool and functionality. However they would not like to use the tool frequently, find it complex, difficult to learn, need a lot of support, do not learn it quickly, find it cumbersome, are not confident when using it, need to learn a lot and would not like to use it again. These SUS diagrams would seem to suggest external respondents would like to use the tool frequently, however the sample size is very small, so one or two people with a strong opinion influence the final results a lot.

To summarise it would seem that both internal and external respondents have serious usability problems and are not able to successfully author courses and adaptive strategies. There is some indication that at least among external respondents, the functionality of and ideas behind the tool are accepted. The qualitative feedback points in the same direction. While users seem to be able to create DMs, creating CAMs is more difficult and they really struggle with creating CRTs. However the functionality seemed to be perceived in a positive way. Only one respondent questioned the concepts and ideas behind the tool; however absence of critique does not automatically imply acceptance of the ideas. These results would seem to indicate that it may be possible to improve the tool to a level where authors can successfully create courses with it. We have seen that internal and external respondents significantly disagree with each other on whether they want to use the tool frequently, whether the tool is well integrated, and whether the tool is cumbersome.

In terms of future developments, we have seen in section 6.7.4.2 that the help and documentation needs to be improved. It is necessary to provide usable examples of courses and adaptive strategies. Authors need to have the opportunity to deploy and test their courses. Instead of acronyms clearer names should be used. The abundance of tabs and complicated editing options available should be reduced. And finally, the authoring interface should be simplified and graphical representation of the CAM should be improved. Novice users should not be confused with advanced editing.

6.8 Related Research

In this section we look at a number of other methods of AEH authoring related to the CAM approach, and analyse differences and similarities.

6.8.1 Competence Based Learning Tools

6.8.1.1 iClass Competence-based Knowledge Space Theory (CbKST) Tool

The FP6 research project iClass⁴⁷ developed several skill-based learning tools, the Competence-based Knowledge Space Theory (CbKST) Tool, for planning, competence assessment, self evaluation and learner knowledge presentation. The Domain Model Editor is a tool for structuring knowledge domains and for defining curricula. It provides visual guidance in different learning phases and supports self-regulated learning. In Figure 38, an overview of the iClass Domain Model is shown. It consists of skills, concepts, action verbs, activities (learning objects), and assessment items (problems) [4], [137].

Like in the CAM, a set of concepts represents the knowledge domain. An action verb describing the skill level in combination with a set of concepts models a skill. Action verbs are chosen from the Bloom taxonomy [8]. Similarly to the CAM component, additional pedagogical relationships can be defined; however the only relationships which are allowed are *prerequisites* and *associations* between *activities*, or between *problems*.

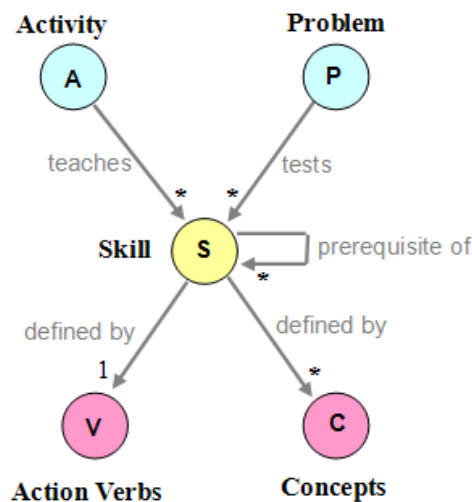


Figure 38 Knowledge Domain Model

In general, we can claim that our approach is more versatile, based on the fact that it does not restrict authors to modelling competencies, and it allows, in principle, for almost any

⁴⁷ <http://www.iclass.info>

behaviour to be specified, not just prerequisites. Our approach could therefore also be transferable to domains other than education. In the GRAPPLE project, tools are being developed, for visualising students' progress as well as the progress [111]; this will explicitly support self-regulated learning.

6.8.1.2 TENCompetence Personal Competence Manager

The aim of The European Network for Competence (TENCompetence)⁴⁸ is to Provide Lifelong Competence Development. The reason for using the paradigm of competence development is that learning can take place not only in a formal academic setting, but in many other less formal ways as well. A competence approach to lifelong learning ensures that the pursuit of a learning goal does not happen in a vacuum, but is bound instead to a precisely defined purpose such as an occupation, a profession, a market or a particular life or work situation.

The project features a Personal Competence Manager, a space where users can access tools for organising participation and authoring of competence development programmes within social networks focused on competence development. The manager allows the user to access and manage the different aspects from various sources.

The system built in the TENCompetence project is based around the needs of users who want to develop their competences, rather than around the (management) needs of educational, governmental or industrial institutions. There are various roles available and a user can do the following.

- Develop his/her competence independently
- Follow a specific competence development course
- Design competence development courses
- Define competence development routes
- Participate in competence development networks

⁴⁸ <http://www.tencompetence.org>

- Switch between roles; however, these switches are all user-driven, and not system-driven. Thus, there is no generated adaptivity, hence users need a strong sense of where they want to go, in order not to face the ‘lost in hyperspace’ [63], [38] problem described previously in section 1.1.

Our novel graphical model for authoring of Adaptive Hypermedia approaches the issue from a different angle. In AH, the emphasis is very much on authoring individual adaptive courses, which automatically adapt to the user’s needs. One of the aims of the GRAPPLE project is to make different AH courses work together rather than in isolation. Building upon this line of thought, competences from an approach such as taken in TENCompetence, can be a very useful source of information about the user, to base adaptation on. At the same time, a competence management based approach can be of added value to the AH community. The user can gain a sense of his personal development and of which AH courses to take to reach an overall goal.

6.8.2 Mash-ups of Learning Environments: ROLE

The Responsive Open Learning Environments (ROLE) project⁴⁹ aims to deliver an environment for flexible composition of learning services. Learners themselves can adapt the combination of components and create their own personal learning experience. This requires users to have a strong idea of what they want to achieve and which services would be suitable for this. This clearly contrasts with our approach, where in fact the number of available choices is limited to the most suitable ones, as defined by authors of adaptation and as generated based on system-driven adaptation.

6.8.3 My Online Teacher 2.0 (MOT2.0)

My Online Teacher 2.0 (MOT2.0) [73] follows up from MOT, see section 2.7.1, and is built around an extension to the LAOS [41] framework. The extension adds a social layer to capture (and adapt) information from collaborative authoring and authoring for collaboration.

⁴⁹ <http://www.role-project.eu>

As MOT2.0 builds on MOT and the LAOS model, it features a separation of concerns, and separation of content features from pedagogical and adaptation features, as does our approach. The differences with our approach are:

- MOT2.0 is aimed mainly at collaboration, whereas our approach is not explicitly aimed at collaboration.
- MOT2.0 does not make a very strict distinction between authors, learners and students, whereas in the CAM model and GAT tool, the difference is more pronounced.

6.8.4 My Online Teacher 3.0 (MOT3.0)

My Online Teacher 3.0 (MOT3.0) has briefly been mentioned in section 2.7.1 and is a modernisation of the MOT authoring tool. It still comprises a Domain Model, a Goal and Constraints Model, Presentation Model, User Model and Adaptation Model. Concepts in its Domain Model and Goal and Constraint Model are assumed to have a tree structure. MOT3.0 builds upon the strong point of MOT, its reusability. Content can easily be reused in the form of the Domain Model, as can adaptive behaviour in the form of adaptive strategies. MOT3.0 adds imports from formats, which are popular throughout education, such as PowerPoint and Media Wiki.

The differences to our approach are the following.

- Our approach offers a visual way of modelling the content, as well as the adaptation. This is an important improvement upon the use of separate files containing programming code.
- Our approach separates adaptation at a lower level of granularity. The CRTs can describe snippets of adaptive behaviour, and thus non-programming authors would have access to specification of adaptation with lower granularity, instead of using pre-defined full LAG strategies created with PEAL, see section 2.7.1.1.

6.8.5 GRAPPLE Simulation Authoring Tool

Activities are an important part of the learning process [16]. Using activities, the responsibility for learning is placed upon the learners who have to actively engage with the learning material. This active learning can take place in various [108] forms, for example,

role-playing, social activities and field trips. Learners learn best when the learning experience suits their individual needs [36], and therefore it makes sense to consider adaptive learning activities in addition to adaptive presentation and navigation. The GRAPPLE Simulation Authoring Tool focuses on procedural simulations [80], [81]. In order to minimise the programming overhead, the notion of a workflow [144] is taken as a starting point for the adaptive educational simulations. The authoring tool uses workflow constructs like sequencing, splitting, and joining to sequence and order the services (elementary components of an activity) in educational simulations. These constructs can then be adapted to individual learners. Workflow languages for process modelling and coordination have very similar requirements to a language needed to represent the adaptive educational simulations. Workflow management automates business procedures, passing documents, information, or tasks between participants. The process follows a set of rules or procedures [154]. The main difference between workflow management and adaptive educational simulations is that workflows usually manage “real-world” processes whereas adaptive educational simulations merely simulate the process.

The Simulation authoring tool follows a layered model, featuring a separation of concern as a key element in order to promote reuse.

- The *Resource Model (RM)* models all assets (concept instances and service instances) that are available for use in an adaptive course or adaptive simulation. The assets consist of, for example text images, sound, video, chat services, email services, forum services, exercises, references and datasets. Assets can be included in a closed-corpus repository or retrieved from elsewhere, for example from the Web. Each model has a required set of pedagogical metadata, for instance, the difficulty, the language and the media format. Services also need to specify their inputs and outputs.
- A *Domain Model* for the purpose of Adaptive Simulations, which consist of a *Concept Domain Model (CDM)*, which is very similar to the DM in the CAM model, and a *Service Domain Model (SDM)*, which is also called *Service Workflow Model (SWM)*. The CDM extracts concepts. The concepts are described

in the IMS-VDEX format, in the same way as the DM in the CAM model. The SDM provides a similar abstraction for services. It models abstract services, the concrete services are defined in the RM. In the authoring tool, services are implemented using Portlets and the Web Services for Remote Portlets (WSRP) Specification [120].

- The *Pedagogical Relationship Types (PRT)* are sub-divided into *Concept Relationship Types* and *Concept-Service Relationship Types (CSRT)*. CRTs describe abstract pedagogical relationships between (placeholder) concepts. This is again the same idea as that of CRTs in the CAM model, and the same format is used to describe them. CSRTs are relationships between concepts and a service. The author can freely mix CRTs and CSRTs in an Adaptation Model in the authoring tool, but CSRTs follow their own XML format.
- Similarly to the CAM layer in the CAM model, the *Adaptation Model* is a workspace that combines everything else. Here CRTs and CSRTs are placed and instantiated with concepts and services from the DM.
- *Learning Activities* can be defined by reusable groupings of concepts and services.

6.8.6 GRAPPLE Virtual Reality (VR) Authoring Tool

In the GRAPPLE project, the possibility is also offered to include what is known as Virtual Reality. This can range from simple 3D objects to complete virtual environments. In a VR environment, certain aspects can also be adapted. More specifically, in a VR environment, adaptive navigation, as described in Brusilovsky's taxonomy [21] can be used.

The VR environments authored with the GRAPPLE VR Authoring Tool [57] can support the following.

- *Direct guidance*. In a VR environment, the link is replaced by a viewpoint. The best viewpoint, according to the current user's background, is given to the user.
- *'Link' Hiding*. In a VR environment, different viewpoints are created for a specific object. These viewpoints reside in a *scene sphere*. The user cannot view any object out of this sphere. Using so called Gaze Fixation, the user can walk past the object

with a system-generated orientation, so the user can still control their position, but not their orientation, in the case a viewpoint is hidden.

- *'Link' Sorting*. In a VR environment this can be achieved by introducing Point of Interest navigation that adjusts the viewer's motion speed logarithmically in relation to the distance from an object of interest.
- *'Link' Annotation*. Annotation can be achieved in VR environments by drawing an arrow pointing to the interesting object or by using the flashlight technique that highlights the object.

Based upon the above observations, a VR in the GRAPPLE VR Authoring Tool is composed of the following elements.

- The *scene* consists of lights, objects, viewpoints, cameras and physical properties (for instance gravity).
- *Objects* are the entities, which populate the scene. These are 3D or 2D objects with properties such as shape, texture, colour, position and orientation.
- Objects in a VR environment can have *behaviour*, for example objects can move, rotate or change size. More complex behaviour is also possible in order to simulate real objects.
- A user can interact with objects in a VR environment. *Interactions* are, for example, picking up or moving objects.
- VR environments often allow collaboration and *communication* between users. However, as in the GRAPPLE project collaboration is not explicitly considered, the GRAPPLE VR Authoring Tool does not explicitly support this, although extensions could be defined in the future.
- *Sound* is also a component of a scene.

Adaptation can happen for each of the components described above. The GRAPPLE VR Authoring Tool focuses on adaptation of objects, behaviour and interaction. The tool introduces the notion of Adaptation Types (AT). An AT describes the various properties that can change for an object or for a scene. On a technical level, the GRAPPLE VR Authoring Tool is realised by defining an extension of the CRT and CAM components of the CAM model in order to suit the specific information for VR. Thus, similarities exist

between the approaches at the model level, as planned in collaboration, but the area of application is quite different: VR environments versus Web-based learning.

6.9 Conclusions

In this chapter we have introduced a novel graphical approach to authoring of Adaptive Hypermedia, featuring both a graph-based approach as well as a strict separation between content and adaptation. The CAM model and the GAT tool based on the model, allow non-technical authors to create a conceptual domain and to use predefined relations, in order to create a course. In the CAM model, the adaptive behaviour is represented in abstract reusable relationships between groups of concepts (called CRTs). In this chapter we have described the languages used to express the parts of the CAM model and we have introduced the design of the GAT tool based upon the model.

Finally, we have evaluated the tool. From this evaluation we have seen that authors are mostly able to understand the system and seem to like using it and generally accept the ideas represented by the tool set. However, there are a number of small interface issues, as well as the tool seems to have quite a steep learning curve. This can be improved by providing better example courses and making the conversion to adaptive systems available, so that authors can see their courses in action. Moreover, the usability of the system can be improved by bringing, where possible, the usage of the tools in line with user expectations, based on tools with similar functions.

7 Summary and Conclusions

7.1 Introduction

Throughout (formal or informal) education it is clear that students benefit from personalised attention by their teachers. In an ideal world, every student would receive a great deal of private tuition throughout their studies. However, not everyone is able to afford this intensive private tuition. As a result, teachers usually teach (relatively large) groups of students and are not fully able to give each student enough personalised attention. Adaptive Educational Hypermedia systems have the potential to offer personal attention in an educational environment and thereby improve students' learning outcomes. The great advantage of personalisation systems is that they do not have any limitations on the number of students they can deliver personalised courses for. We are, however, not advocating the replacement of teachers by AEH systems, but merely arguing that they could be a very valuable addition to the learning process.

Despite these clear benefits, AEH has not really taken off and made its way into mainstream use as, for example, Learning Management Systems have.

- An important factor in this matter is the *current state of the authoring (course creation) process of AEH*.
 - Creating courses in AEH systems is considered by many to be still difficult and time-consuming. Contrary to this, it is sometimes argued [84] that creating an AEH course is easier and less time-consuming than creating a static web-page that simulates the same behaviour.
 - Whilst the latter may be true, in reality, a comparison to a linear web-page that does not simulate any adaptive behaviour is more appropriate, as it is closer to the reality of the teaching practice.
 - In reality, we believe that Adaptive Hypermedia authoring will always remain slightly more difficult than creating non-adaptive web pages, and

the challenge is to reduce the amount of added difficulty to acceptable levels.

- An additional factor is the *lack of standards*, which means that a course created for one AEH system often cannot be used in another system very easily. Conversions exist, but the different systems do not always map onto each other, resulting in a loss of content or, more commonly, adaptive behaviour.
- *Reusability* is another issue that complicates the uptake of AEH. Since there is a lot of effort required to create AEH courses, teachers are more likely to do so if they can reuse parts of the content and behaviour for other purposes.

This led us to ask, how we can improve the authoring process of Adaptive Hypermedia in such a way that it allows technically experienced adaptive course authors to successfully specify adaptive courses (see section 1.2.1). While the eventual aspiration of this research is supporting a teacher with non technical background, this thesis has focused on providing support for technically experienced adaptive course authors. This question was broken down into more specific questions, which have been answered in chapters 3, 4, 5 and 6.

Tools can be developed which harvest multiple alternative content. Course authors feel the content both highly relevant for the course and is aligned with the teaching strategy for the course, as we have seen in chapter 3. We explored a connection with a Semantic Desktop setup in this context. Assuming a teacher is an expert in the field he is creating courses on, the teacher's desktop may contain many of related articles. If a teacher starts creating standard linear courses, automation methods can recommend articles to be added (from his desktop) for advanced learners. It is important to give teachers the final say as to what gets added to their course. The teacher could have manuscripts in preparation and the upcoming exam for a course on his desktop. Adding these automatically is obviously undesirable. Recently we are seeing others picking up on the topic of automatic content addition in AEH authoring. For example, the MOT3.0 system [67] includes import functions from PowerPoint and Media Wikis, such as Wikipedia.

Another important issue indirectly contributing to the difficulty in course creation in AEH is the lack of *standards*. The lack of standards means that courses created in one AEH system are often not transferable to another system and sometimes even not to the next incarnation of the same system. This means that with every new system, teachers would have to redo their efforts to create courses.

At the moment AEH cannot fully be expressed in IMS-LD and vice versa, as we have seen in chapter 4. Both have similar goals, but IMS-LD is a widely accepted standard. Therefore courses designed in the IMS-LD specification can be played by various different IMS-LD compliant players. We have seen that, while IMS-LD does not support all adaptive features of AEH, something can definitely be learnt from the standardisation efforts in the IMS-LD community. We are now seeing more work being done on connecting AEH to standards such as [72], indicating the importance of the issue.

An important aspect of authoring of AEH, which is currently relatively weak, is that of *reusability*. Application of standards can address this problem. Reusability however is not just about standardisation. In Chapter 5 we have seen that a meta-language can be designed and a tool can be built, which increases the reusability of adaptation strategies independent of the course content. These meta-strategies could be created by technical experts and reused by teachers.

As we saw in chapter 6, an authoring tool can technically be built that allows specifying adaptive courses, including adaptation strategies. However evaluations show that work remains to be done, to make these tools usable for technically experienced course authors, especially with regards to the creation of adaptation strategies. In this chapter we introduced a novel authoring model, the Concept Adaptation Model, and the tool developed based upon this model, the GRAPPLE Authoring Tool. The novel approach lies in combining a *graphical authoring* approach with a model built around re-usability of both content and teaching strategy. The model is specifically targeted towards teachers without computer-science backgrounds. The behaviour is defined in generic relationships, which

authors can create and modify, but which most teachers would be able to use with little change.

The rest of this chapter is organised as follows section 7.2 summarises the contributions, limitations and indicates directions for further work, for our approach to lessen the content creation burden. Section 7.3 summarises the contributions, limitations and indicates directions for further work, regarding our comparison of AEH and the IMS-LD standard. In section 7.4 we give a summary of the contributions and limitations of our approach to improve reuse of adaptation strategies and indicate directions for future research. Section 7.5 summarises the contributions, limitations and indicates directions for further work regarding our novel graphical approach to authoring of AH and section 7.6 outlines our final conclusions.

7.2 Automatic Content Addition

7.2.1 Summary of Contributions

In section 1.2.1 we asked whether tools can be developed which harvest multiple alternative content (i.e. supplementing the available content for the course under construction, with content already stored/accessible on the author's desktop) which is both highly relevant for the course and is aligned with the teaching strategy for the course under construction. In chapter 3 we have introduced an approach to automate part of the content creation process. We have shown that it is possible to connect AEH authoring systems to the Semantic Desktop and semi-automatically extend lessons with articles for advanced learners. We have thus shown that tools can be built and have shown in evaluations that the alignment and fit is such that they are perceived as helpful and useful.

7.2.2 Limitations

The (semi-)automatic content addition as described in chapter 3 has a number of limitations. First of all, in order to extend a lesson with relevant content, the lesson itself needs to contain some content to match possible extensions against. This content should

contain at the very least a minimal description of all key concepts. In other words, a basic linear lesson is required.

Secondly, as can be clearly seen in the scenario, the method was initially designed for courses using a specific strategy, namely the *beginner-intermediate-advanced* strategy. The final prototype goes one step further, and has a selection field for strategies. This displays a list of strategies that can be applied, together with the weights and labels that should be used with them. While this works fine for the scenario mentioned before, it will not work well in cases where resources that are to be added should not all get the same weights and labels. Authors can always change weights and labels of individual resources that are to be added, but this requires a thorough understanding of both the mechanism of weights and labels and the strategy and how it actually works. Thus, as is perhaps to be expected, more complex scenarios ask more work from the author, and a higher level of understanding of adaptation.

7.2.3 Future Directions

In chapter 3 we have made some progress towards helping authors create content by means of (semi-)automatic content addition. However more remains to be done in this area. Connecting our methods to not just the author's Semantic Desktop, but also to an array of online sources, would be a logical step. Such sources could include Learning Object Repositories such as Ariadne [62] and academic article databases such as Google Scholar [76].

The way that added articles link into the course could also be improved, from a linguistic point of view. At the moment they are added as links, without any narrative. This disturbs the reading of a concept. Simple techniques may be enough, such as using a pool of linking sentences.

The author currently has to initiate the recommendation process. It may be more natural for authors, if the recommendations would be integrated at author-time rather than calculated upon demand.

Another avenue that is currently getting some attention is the import from everyday sources, such as Media Wiki and PowerPoint [67]. Although these methods make some assumptions as to the intended division of the content into concepts and attributes, a great advantage is that the formats have a very wide user base.

The import of e-learning content in standard formats (such as LOM [98], IMS-CP [99]) could be another way of helping authors lessen the content creation burden. The use of standards could drastically reduce the overhead of re-creating the same content in different systems.

Finally, (semi-)automatic content creation is only one part of the process. All the different components, such as metadata, links, adaptation strategy, etc. can be further explored in terms of what type of (semi-)automatic authoring process they permit. For instance, related content could be linked (semi-)automatically. Images can be (semi-)automatically labelled ‘visual’, to be seen by people who have a preference for visual material, etc. In this sense, our contribution to meta-strategies, as in chapter 7.4 below, is illustrating (semi-)automatic authoring for the adaptation process. Further research in this particular area is discussed separately.

7.3 E-Learning Standards

7.3.1 Summary of Contributions

In section 1.2.1, we wondered whether AH can be expressed in IMS-LD and vice-versa. In chapter 4 we have therefore compared the worlds of Adaptive Educational Hypermedia and IMS-Learning Design. The IMS-LD specification is a standard specification for online learning systems, with similar goals to AEH. Our exploration was guided by Brusilovsky’s

taxonomy [21] and has identified adaptation of modality, link sorting, adaptive link generation and interface based adaptation to only be partially supported by IMS-LD. These forms of adaptation are more fully supported and better support would be required in IMS-LD in order to be able to express AEH in IMS-LD. At the same time we established that proper support for activities and the specific roles, stakeholders may have in the learning process, adaptive user grouping and changes on the fly are often missing. AEH support for these types of adaptation would need to be added, in order to be able to fully express IMS-LD in AEH.

7.3.2 Limitations

Our study has only been a preliminary exploration and had quite a technical focus. We have explored which features AEH has that IMS-LD misses and vice-versa. However, we have not investigated how essential or even desirable these features are from an educational point of view. We have also restricted ourselves to one standard, the IMS-LD specification and have restricted AEH to what Brusilovsky's taxonomy describes, and where more concrete examples were necessary we restricted ourselves to the LAOS framework.

7.3.3 Future Directions

Standardisation is an important issue in AEH. The GRAPPLE project is creating various new representations, which are explicitly designed for various different AEH systems to be able to use them. The hope is that these will eventually lead to usable standards. In addition to this, it is necessary to explore other standards that are already available. This effort could lead to converters for a number of the most compatible standards for various AEH systems, which could in turn make interoperability a reality.

7.4 Improved reuse of adaptive strategies

7.4.1 Summary of Contributions

In section 1.2.1, we hypothesised that a meta-language could be defined and tools could be built, which increase reusability of adaptation strategies. In chapter 5 we have shown how a meta-level addition to the LAG adaptation language can be defined, to improve reuse

capabilities of adaptation strategies. These meta-strategies would solve some of the limitations that currently exist of writing truly generic strategies and improve the ease with which strategies can be combined. We have also shown how a tool to (re-)use such templated strategies can be built. This avenue of research is based upon a move away from expecting teachers to specify their own adaptive teaching strategy. Instead, a highly skilled expert creates very generic strategies (*meta-strategies*, or *strategy templates*), which the teachers can then use on their content with little effort. We are seeing that the issue is recognised and picked up by other researchers [134].

7.4.2 Limitations

This work explicitly expects that the teacher and the person specifying the generic strategy are going to be different in most cases. The addition makes strategies more re-usable and thereby strategies would have to be created or changed less often. Teachers can simply reuse what is out there, while experts can create highly re-usable strategies. The meta-addition does, however, potentially make writing these strategies more complex, hence requiring more effort on behalf of the person creating the meta-strategies. The work also has not been tested, so an implementation and experiments would be necessary to verify the underlying assumptions.

7.4.3 Future Directions

The most logical next step in this direction would be to follow up our research, with the implementation of a prototype, based on the proposed setup. This should then be followed by a thorough evaluation. The evaluation should not only investigate the usability of the tool, but especially the quality and re-usability of the strategies created with the tool. Finally, an evaluation should look at whether teachers are able to reuse such tools. For this, additional support may be needed, either in the form of explanations and guides, or in the form of tool-support.

7.5 Graphical Authoring: CAM

7.5.1 Summary of Contributions

Previously, in section 1.2.1, we asked whether tools can be developed that allow the specification of an adaptive course including the adaptation strategies and whether technically experienced authors can successfully create courses with these tools and whether they can successfully create adaptation strategies? In chapter 6 we introduced a novel graphical approach to authoring of AEH. The Concept Adaptation Model combines both a graphical approach, with a layered model, the latter based around reuse. Writing adaptive behaviour code can be done via pedagogical relations, called CRTs. The power of this approach lies however in the fact that most teachers would not need to write such CRTs, which are highly technical. Instead, they will be able to specify their conceptual domain and course, while only re-using generic pedagogical relationships from a library. The initial evaluation of the authoring tool built based upon this model showed mixed results however. While technically experienced authors can successfully create Domain Models, they struggle with creating the adaptation using the more concept Concept Relationship Types. Also authors are having issues with creating the overall course in the Concept Adaptation Model.

In section 6.7, we have seen that the tool we have built has quite a few usability problems. In terms of our original research question, we have shown that a tool can be built in which technically experienced authors can easily author DMs. While at present authors are not able to specify courses successfully, from the feedback gathered it seems that it may be possible to improve the tool to a level where this becomes possible. Whether the same is the case for adaptive strategies is not clear. The evaluations have shown that creating adaptive strategies remains a very difficult task for technically experienced authors.

7.5.2 Limitations

A graphical model is very much in line with the nature of hypertext, which is a set of nodes and links. However, creating the conceptual domain in a graph form is a departure from the way many people author content in editors like MS Word or PowerPoint slides. While we do not expect teachers to author complex behaviour themselves, we do expect them to be able to use the relations in a course and to fill placeholders with the relevant concepts. This should not be a serious hurdle for anybody with any sort of technical or mathematical background, but may require more effort from those without such a background. This can be alleviated in further versions of the tool by allowing for import from commonly used tools (such as MS Word or PowerPoint), either directly, or via other authoring tools for adaptation, such as MOT3.0 [67].

Another limitation is that the adaptive pedagogical behaviour still has to be defined by a person knowledgeable about an adaptation language.

7.5.3 Future Directions

Immediate future work will consist of improving the prototype of the authoring tool based on both the quantitative evaluation as well as on the formal qualitative evaluation. There are a number of minor interface issues and the tool has quite a steep learning curve. This may be caused by the lack of real example courses and the fact that at the moment authors cannot deploy their course to an AH system to see it in action. Therefore, the next step is to fix the interface issues. The deployment functionality has since been added.

The authoring tool will be integrated into the whole GRAPPLE architecture, which aims to incorporate AEH into major Learning Management Systems. As part of this integration process, a form of user-management will be added to the tool and converters are being developed for deployment of authored courses to the adaptive engine. As the adaptive engine will be integrated into an LMS, from the users' point of view they will be authoring adaptive courses for use in the LMS. That this is in fact done by integrating AEH systems into the LMS is, ideally, transparent to the user.

7.6 Final Conclusions

As has been shown in this thesis, based on literature, implementations and evaluations, Adaptive Educational Hypermedia can improve learning outcomes by offering a personalised learning experience. A very important obstacle in the way of general acceptance of AEH is the process of creating AEH courses. Our work has explored authoring of both the content creation and adaptation behaviour.

We have shown that on the content creation side, automation methods, which allow to semi-automatically extending existing courses, are promising. Future research should build on these results and investigate connecting to other sources and imports from popular non-educational formats, such as PowerPoint. Moreover, (semi)-automatic generation both of content, of links within conceptual graphs, of the content representation, as well as the (semi)-automatic inclusion of pedagogical metadata and links, need to be further explored.

In terms of potential imports from current standards, we have furthermore investigated similarities between AEH and IMS-LD and have found that there are common goals and some of the adaptive features in AEH are also supported in IMS-LD. This avenue needs further exploration, especially in terms of extending IMS-LD towards full adaptation.

Standardisation is long overdue in the AEH community and a good standard that can express at least all adaptation as described by the seminal work represented by Brusilovsky's taxonomy [21] will make it possible for authors to create a course once and use it in any AEH (delivery) system, making the initial time invested more worthwhile. Future research around standardisation is therefore needed.

Standardisation is however not the only way reusability can be improved for AEH. Another approach can be to have experts create highly generic and reusable adaptation strategies, which can be further combined. This will mean that the resulting strategies created by the experts can be reused and combined by many other authors. To illustrate this idea, we have

defined a meta-level addition to the LAG adaptation language, which improves upon the reusability of adaptation strategies. This is a departure from making authoring of the adaptive behaviour easier and accessible to novices and instead focuses on producing highly reusable adaptation strategies. We recommend that further research first of all evaluates whether teachers want to be able to create their own adaptation strategies, and whether thus making specification of adaptation easy, at the cost of expressiveness, is the way forward, or whether teachers are happy with highly reusable strategies created by experts. A factor to investigate is also who the experts are going to be that are going to implement these strategies. Using adaptation templates, for instance, which are only instantiated with a small number of parameters, might be an easier way to create adaptation strategies by non-experts. Such strategies may not be extremely flexible, but may give enough flexibility to allow authors and teachers to deviate to some significant degree from the ‘one size fits all’ approach.

Regarding the creation of adaptive behaviour, we have introduced a new graphical model for authoring of AEH, and built and evaluated an authoring tool based on this model. This model is based on previous research, and the different language formats are based on collaborative discussions with researchers in the GRAPPLE consortium. We recommend the GRAPPLE consortium to disseminate and promote these formats as they could represent the basis for the needed AEH standards.

In the future, both regarding the creation of content and the adaptive behaviour, automation at delivery time should also be investigated. It should be explored whether teachers are willing to give up some control over what students see and whether students would want to access content that is not created or even recommended by their teacher but, for example, recommended by the delivery system, based on semantic extraction or other techniques. It should also be investigated whether such environments would benefit from automatic adaptation, which is not based on the teacher’s pedagogical decisions but, for example, on which path peers have taken, in relation to their exam scores. Such a type of adaptation is

found in Web 2.0 based adaptive systems [73], where often the common denominator represents the desired path.

To conclude, we can say that authoring of AEH will likely remain more time consuming than creating linear text, either as lecture notes or as a web page. Therefore it is important that authors are well aware of the benefits offered by the supplementary effort. Hence, further research which illustrates this return on investment to the general public, and especially, to the authoring community, is therefore very important.

Appendix I. GAT User Guide

The *GRAPPLE Authoring Tool* is a web-based application in which authors are able to specify their instance models. They can create and maintain individual models – DM, CRT and CAM. GAT provides a common shell, with 3 components integrated into it:

- *Domain Model tool* for editing Domain Models
- *Concept Relationship Type component* for editing concept relationship types
- *Concept Adaptation Model component* for editing the whole concept adaptation model

Note: In the future this basic set will be enhanced with authoring tools for simulations and virtual reality, which will be addressed in upcoming versions of the GRAPPLE documentation.

In the CAM tool, concepts from the DM tool can be selected and the adaptive behaviour can be attached to these concepts. *Sockets* are containers where concepts can be placed in the CAM tool. For a concept to appear in the final adaptive lesson, it needs to be selected in the CAM. This implies the following requirement on the implementation: There should be a common shell for the DM and the CAM tools and a drag and drop facility from the former to the latter, to allow for one (or more) domain concepts to be selected and dragged into the CAM tool. The *Google Web Toolkit* (GWT) was chosen for this common shell. CRTs dragged into the CAM tool can be then populated with domain concepts from the DM tool, as long as the particular CRT allows for that particular type of concept. The tool allows several windows open at the same time. The dependencies between DM, CRT and CAM are handled by the tool, when copying and pasting information between the windows. In Figure 39 the three main components are visible in floating windows, from left to right the DM-tool, CRT-tool and CAM component.

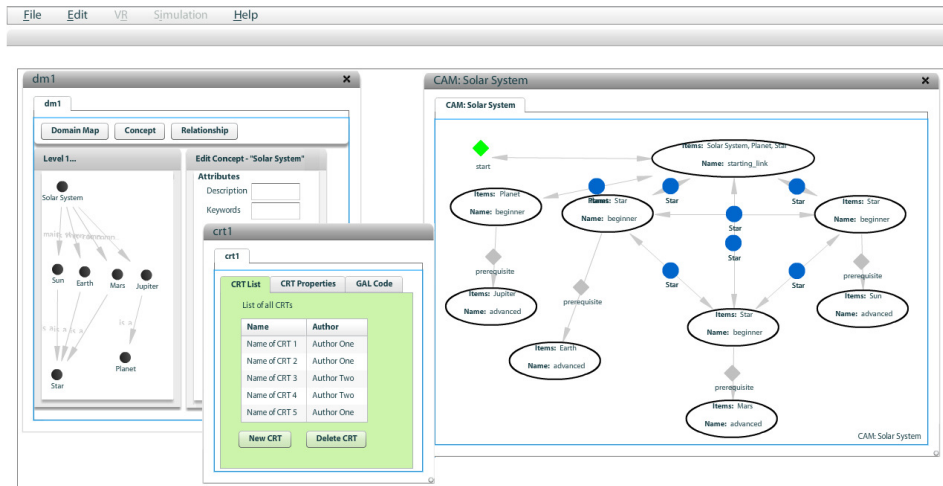


Figure 39 GRAPPLE authoring tool.

In the DM-tool we see a graph, with several nodes and links. The nodes represent the concepts in the subject domain and the links represent semantic relationships, without any attached behaviour. Authors can create conceptual Domain Models, by inserting the concepts and their links in the CRT-tool.

The CRT-tool gives the opportunity to edit the adaptive behaviour of a CRT. Authors specify a title and description of the CRT. Many authors with less technical skill will use the CRT-tool mainly to look at the descriptions of and possible User Model variables in use by a CRT. Authors can, if they wish to customise a CRT or create one from scratch, specify the number of sockets; create a list of User Model values that can be used in the code and write the actual adaptive behaviour in GAL code.

The CAM component instantiates CRTs with concepts. In Figure 39 we see the nodes, which represent sockets and the links. The links with a diamond label represent the CRTs. The round labels indicate that sockets share concepts. Authors can insert CRTs, from the list of existing CRTs and copy concepts from a Domain Model, in a DM-tool window to instantiate the CRTs.

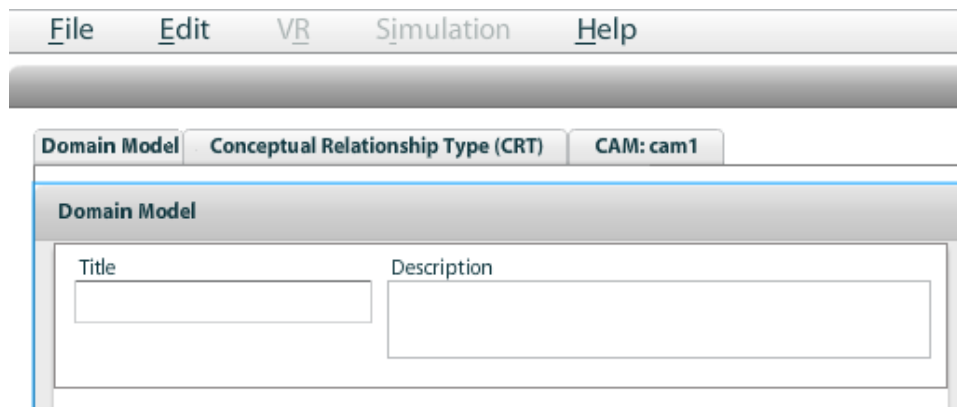
The CAM models can be deployed by the *GRAPPLE Adaptive Hypermedia Engine* (GALE). GAT communicates via web-services with a web server, which provides storage

and retrieval of the data. The CAM model groups domain descriptions and adaptation descriptions into adaptation strategies or adaptive story lines. These story lines need to be exported to an adaptation engine. This engine can be the adaptation engine developed in the GRAPPLE project or any other engine that can work or has converters for one or more of the exported adaptation languages. The CAM tool thus will export several languages of various levels: the CAM XML language, a combination of CAM and CRT language, the adaptation engine's own *GRAPPLE Adaptation Language*, and possibly also other languages.

The GRAPPLE authoring tool requires a recent web browser with Flash Player 9 or higher and an internet connection. The graphical tool is implemented in Adobe Flex 3 technology (<http://www.adobe.com/de/products/flex/>) and the Web Service is realised with Apache Axis2 framework (<http://ws.apache.org/axis2/>).

1.1. Domain Model Component

The Domain Model is the default panel available to the author when he opens the GAT tool, as displayed in Figure 40.



The screenshot shows the GAT tool interface. At the top is a menu bar with 'File', 'Edit', 'VR', 'Simulation', and 'Help'. Below the menu bar is a tabbed interface with three tabs: 'Domain Model', 'Conceptual Relationship Type (CRT)', and 'CAM: cam1'. The 'Domain Model' tab is selected and active. Inside this tab, there is a sub-header 'Domain Model' followed by two input fields: 'Title' and 'Description'. The 'Title' field is a small rectangular box, and the 'Description' field is a larger rectangular box.

Figure 40 Opening the GAT tool.

The author can create a new Domain Model (or Domain Model) by filling the Title and Description fields, as shown in Figure 40. Once the author has assigned a Title and Description to the new Domain Model, he can save it by selecting on the main bar File → Save, as shown in Figure 41.

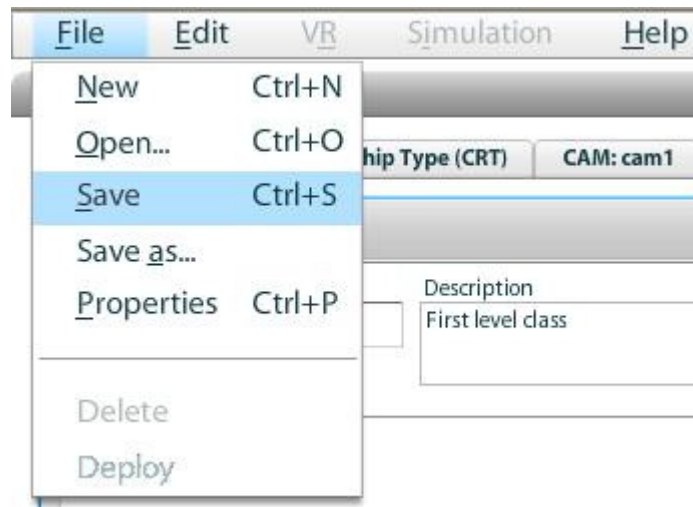


Figure 41 Save a new Domain Model.

The author can create a new Domain Model also by selecting on the main bar File → New, as shown in Figure 42.

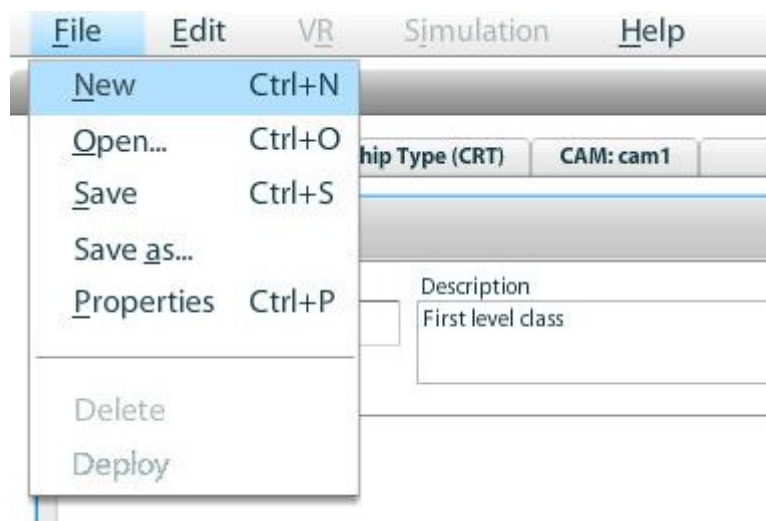
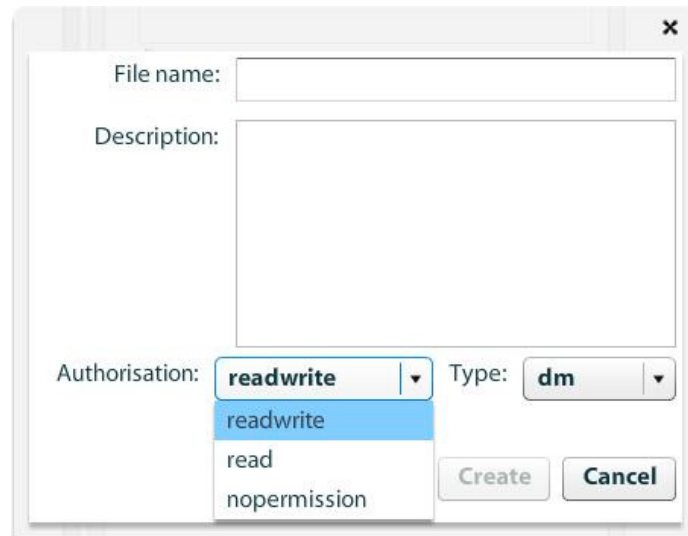


Figure 42 Create a new Domain Model.

This triggers a window that asks the author which file he is going to create.



File name:

Description:

Authorisation: **readwrite** ▼
readwrite
read
nopermission

Type: **dm** ▼

Create Cancel

Figure 43 Definition of a new Domain Model.

From here, the author can assign a Title and Description to the Domain Model, by selecting Type **dm** and the Authorisation level that can be:

- read-write: the author decides to make the Domain Model available to the other authors
- read: the author makes the new Domain Model read only to the other authors
- nopermission: the author does not provide any permission to the other authors.

Note: the Domain Model is always readwrite to the author who has defined it.

The author has created the Domain Model named Astronomy. Now he can start to populate it with concepts. By right-clicking on the Domain Model panel, he selects the option Add Concept, as shown in Figure 44.

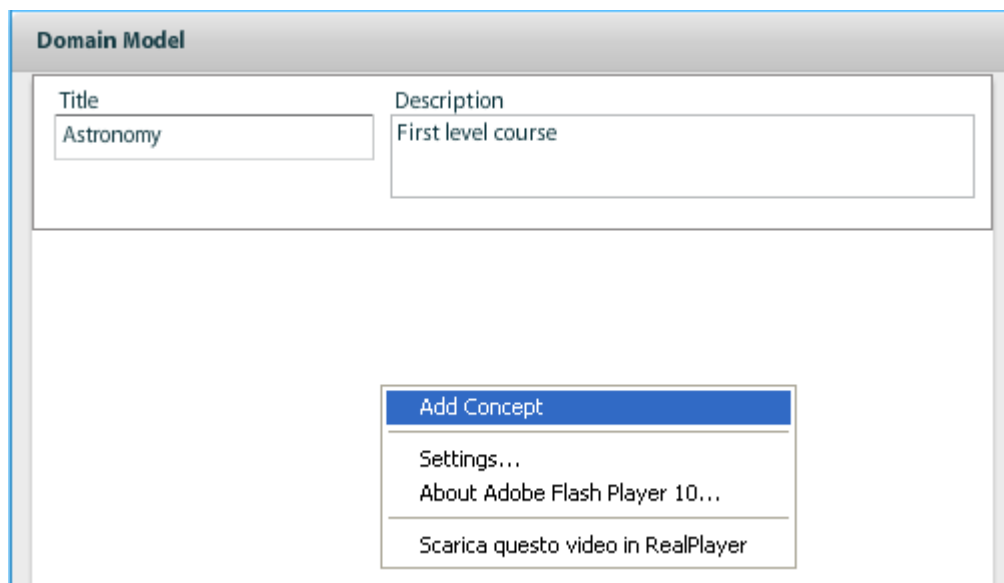


Figure 44 Adding a new concept.

The author edits the new concept by assigning a Name and related Description in the Concept Editor panel.

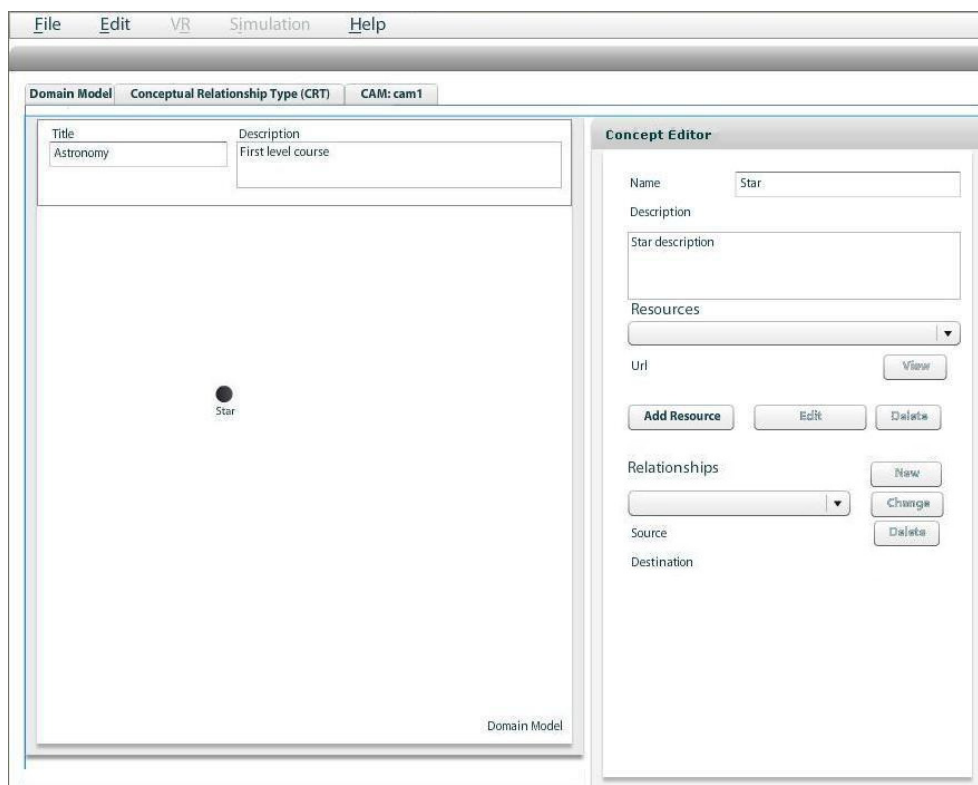


Figure 45 Creation of a new concept.

The author can assign resources to a single concept by selecting Add Resource button.

Resources

Url

Metadata

Title

Description

Add

Save Cancel

Figure 46 Resource addition.

Here it is possible to add the URL of the resource and assign a Title and Description to the single resource.

Resources

Url

Metadata

Title

Description

Add

Save Cancel

Figure 47 Resource window filling.

Once the author has saved, the new resource is visible in the Concept Editor where the concept Star is defined.

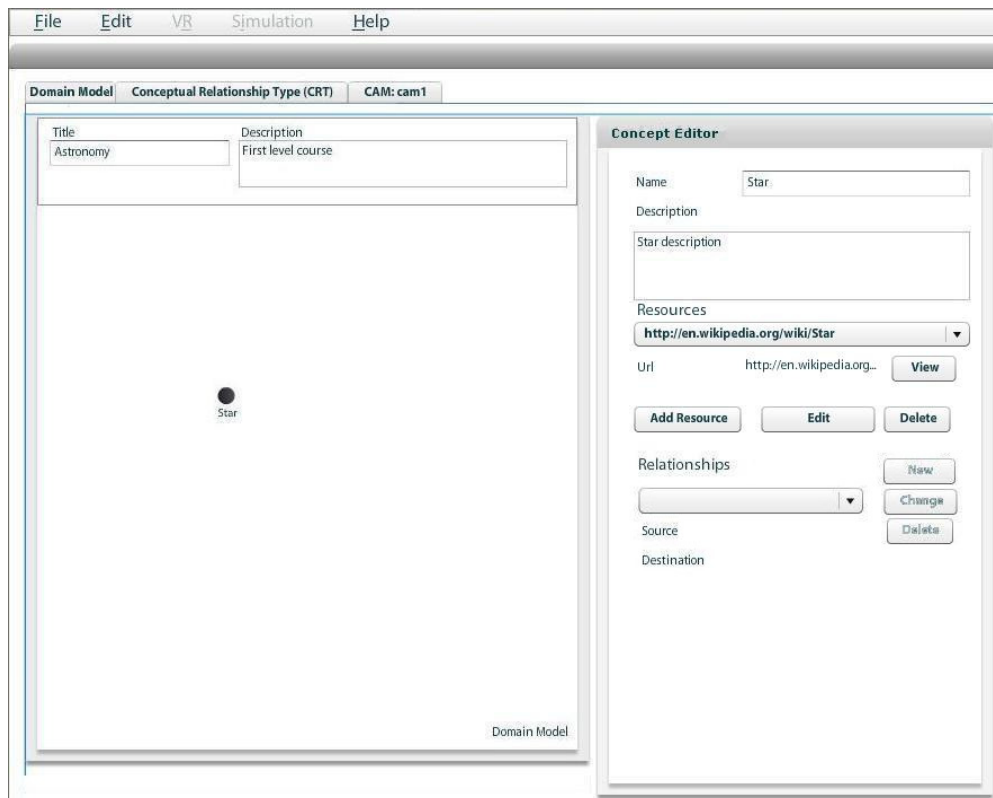


Figure 48 Astronomy Domain Model with only one concept Star and a related resource.

The single resource, available in the drop down list, can be always displayed, edited or deleted (View, Edit, Delete). The author can create many concepts related to the Astronomy subject and then, by using the Add concept selection, he can add other concepts like Planet, Satellite and System.

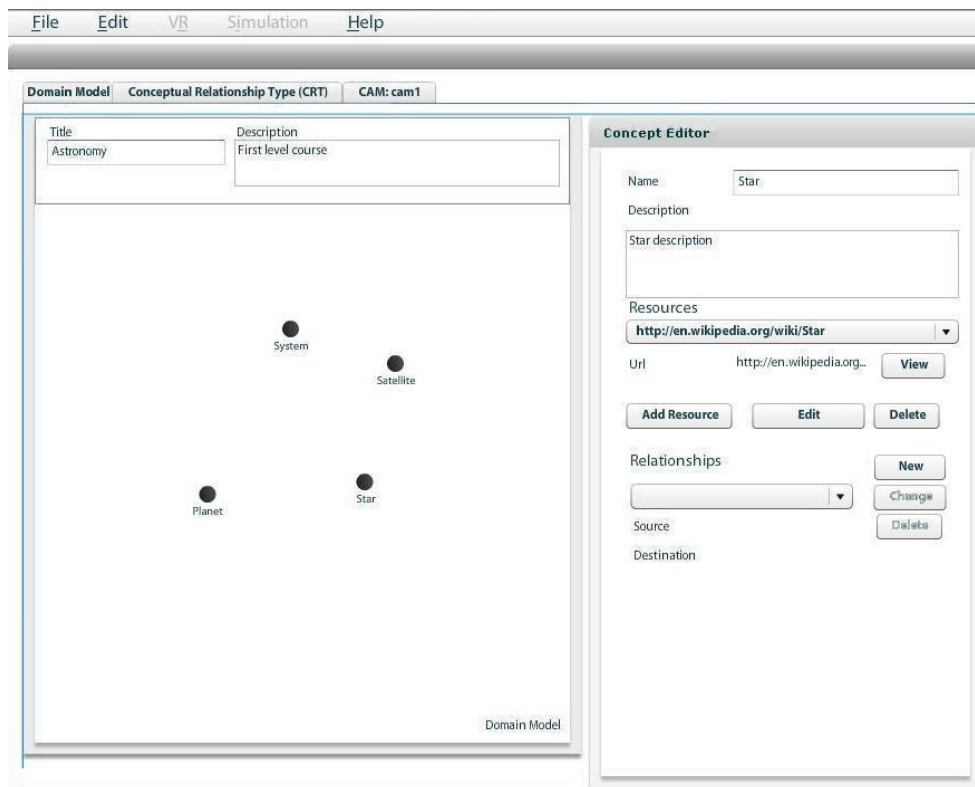


Figure 49 Astronomy Domain Model with four concepts, Star, Planet, Satellite and System.

The author can create relationships among the concepts in two ways:

- by right-clicking a concept in the Domain Model Editor panel,

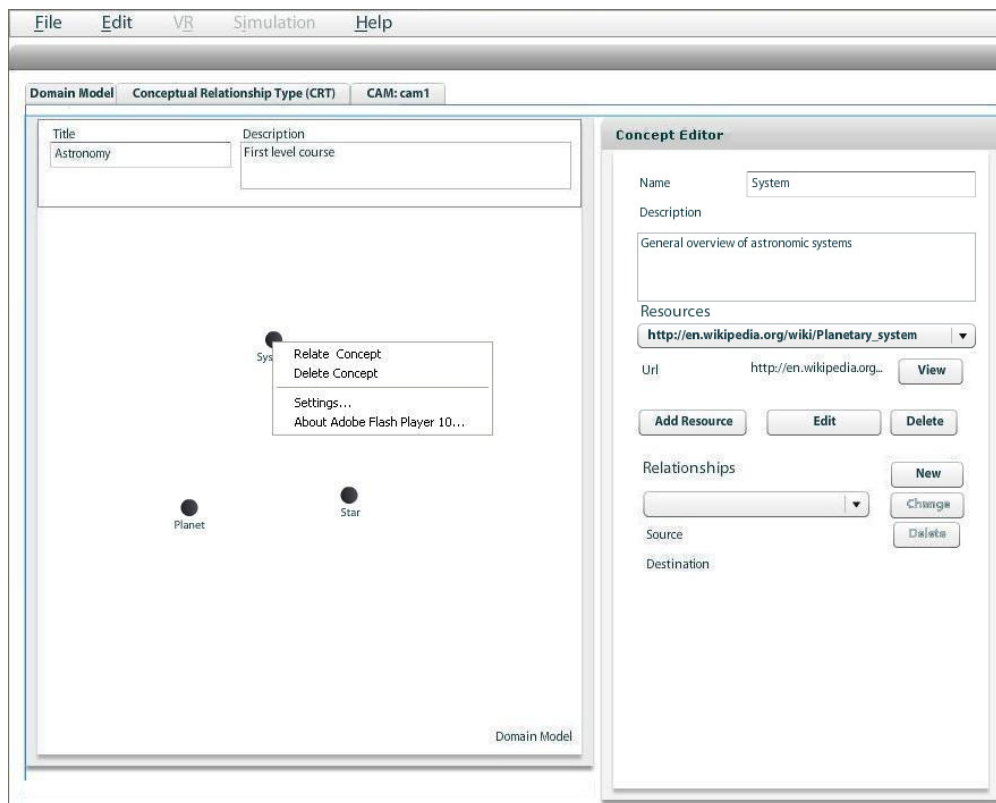


Figure 50 Relationship creation.

- by selecting the button New in the Concept Editor, as highlighted in Figure 51.

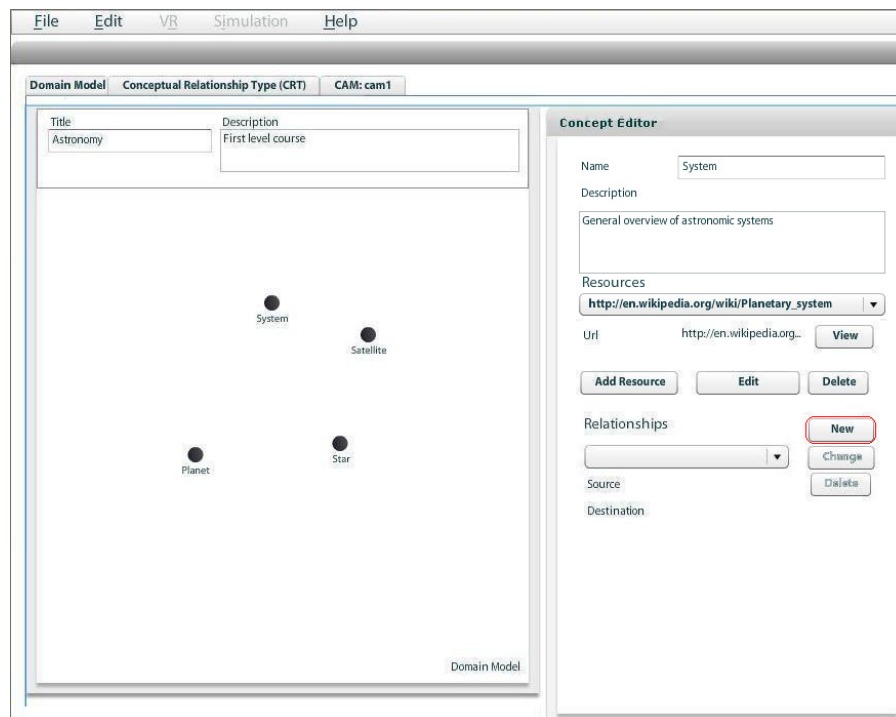


Figure 51 Relationship creation by using the New button.

In both options, the author can set the Source and the Destination of the new relationship. A set of predefined relationships are available from a drop down list: *is-composed-by*, *is-a* and *belong-to*.

Source: System

Type: is-composed-by (selected), is-a, belong-to

☐ New Relation

Destinat...

Save Cancel

Figure 52 Relationship definition.

Since the Planetary System is composed of stars, planets and satellites, the author can create three relationships between the concept System and the concepts Star, Planet and Satellite.

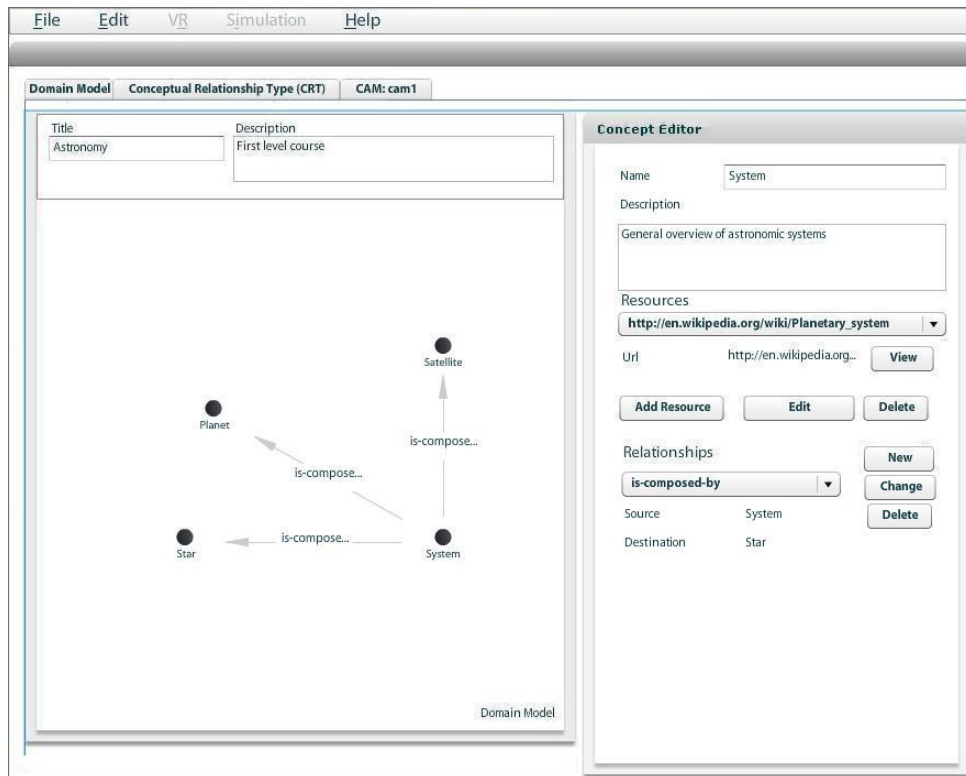


Figure 53 Concepts with relationships.

The author can modify or delete any time the relationships by selecting the buttons Change and Delete.



Figure 54 How to modify or delete relationships.

It is possible to add other concepts to the Astronomy Domain Model, like Moon, Earth and Sun and then add further relationships, such as Moon is-a Satellite, Earth is-a Planet and Sun is-a Star.

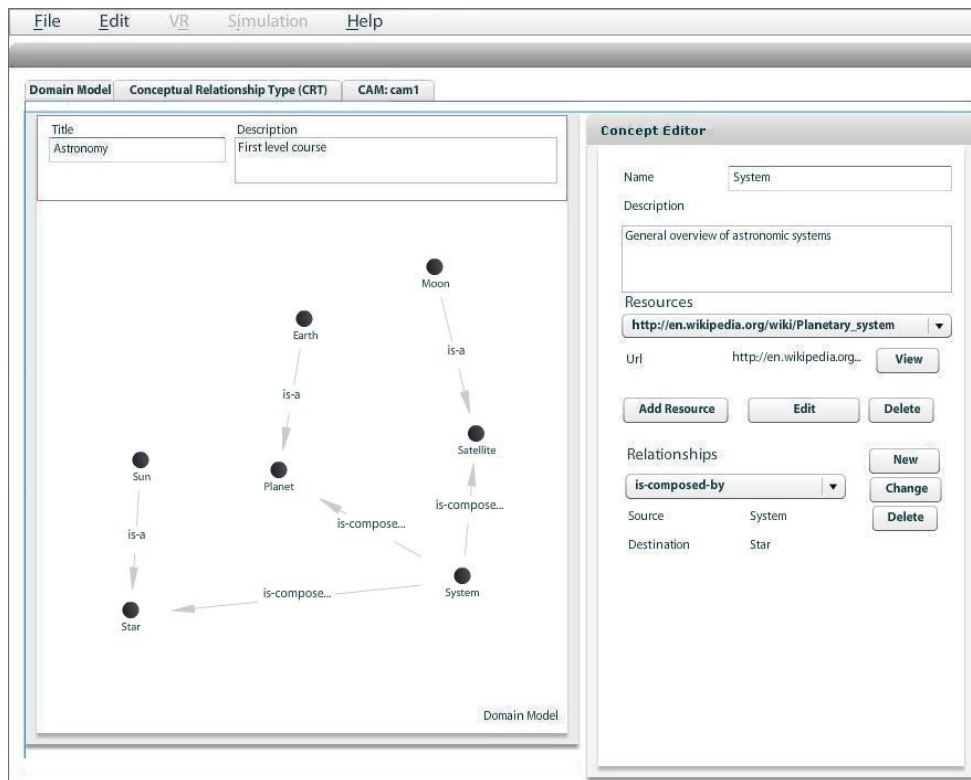


Figure 55 New concepts and relationships.

The author needs to relate the concept Earth with Sun and the concept Moon with Earth to indicate the Earth rotates around the Sun and the Moon around the Earth. In this case, it is possible to create a new relationship type, orbits.

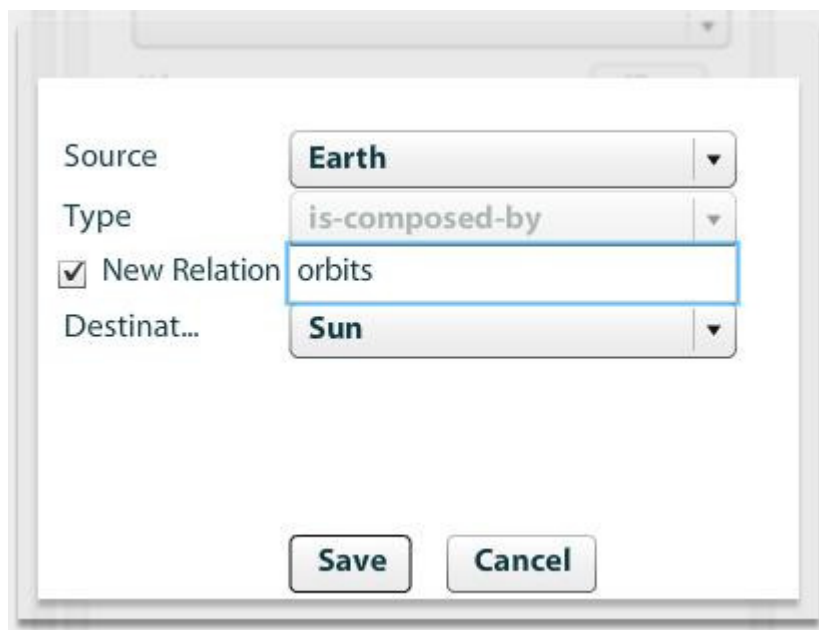


Figure 56 Creation of a new type of relationship.

This new type of relationship, *orbits*, is available to be used for other relationships in the same Domain Model.

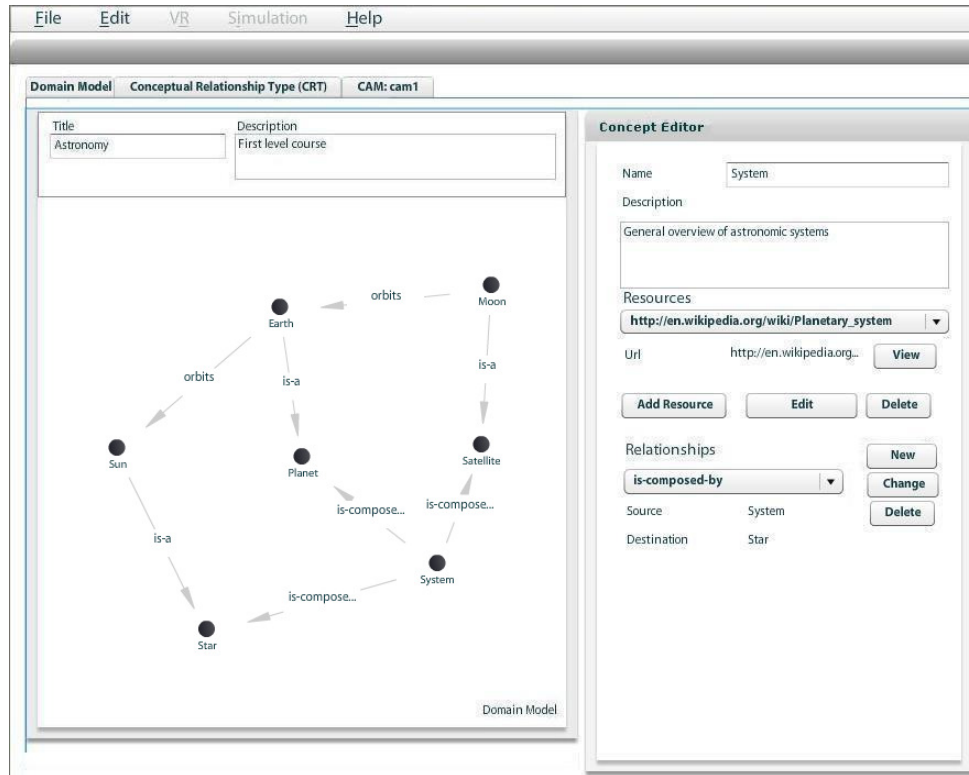


Figure 57 Final version of the Astronomy Domain Model.

I.II. Concept Relationship Type Component

The CRT authoring component consists of a component in the Web-based GAT tool which provides the possibility to create and define CRTs and a Web Service where the created CRTs can be stored and retrieved by other component. Authors can input the information described above in a graphical way and save them to the Web Service. For the reason of interoperability the CRT data is expressed in an XML-based format and the certain CRTs are saved as XML files on the server side into a database behind the Web Service. The information needed to specify a CRT includes:

- general information of CRTs, such as name and description, comment, creation time and author
- information how an instance of a CRT should be visually represented in the CAM
- the structure of the CRT defined by so-called sockets and their properties

- the GAL code which formally defines the adaptation behaviour
- the user variables which the GAL code is accessing
- the properties of the source and target socket which the CRT connects
- constraints, such as the information if sequences of CRT instances may form a loop (which would not make any sense for prerequisite relationships) or if concepts with specific attributes are excluded
- relations to Domain Model relations

The CRT component provides a graphical user interface to specify a CRT. The current version of the CRT component provides a rather simple user interface where the author has to directly manipulate the several fields. Later versions of the CRT component will provide more visual support to the author to make the manipulation easier. The most challenging part of the component development will be to provide as much as possible support to the author who wants to create and modify adaptation code. Programming adaptation code directly in GRAPPLE adaptation language excludes most content authors who do not have programming skills available from creating their own CRTs. Therefore, this component shall provide graphical techniques, which allows an inexperienced author to create new CRTs easily.

In the remainder of this section the usage of the CRT component and the meaning of the several properties of a CRT will be explained. The CRT component is organised with tabs where the various properties can be manipulated. For each tab a screenshot is presented in Figure 58. Opening, saving and creating new CRTs is done in the same way as for the Domain Model (Figure 58 - Figure 61).

The screenshot shows the 'General' tab of the 'CRT Tool'. It contains the following fields and options:

- Title:** A text box containing 'Prerequisite'.
- Description:** A text box containing 'This CRT defines a prerequisite relations which means that the source concepts should be show before the target concepts.'
- Colour / Shape:** A dropdown menu set to 'Diamond' and a color selection box showing blue.
- Structure:** Radio buttons for 'Directed' (selected) and 'Undirected'.
- Source Socket:** A sub-panel with:
 - Type: Source Socket
 - Name: A text box.
 - Min Cardinality: A dropdown set to '1'.
 - Max Cardinality: A dropdown set to '*'.
- Target Socket:** A sub-panel with:
 - Type: Target Socket
 - Name: A text box.
 - Min Cardinality: A dropdown set to '1'.
 - Max Cardinality: A dropdown set to '10'.

Figure 58 General-Tab: General properties can be specified.

In the "General Tab" (Figure 58), the following CRT properties can be specified:

- **description:** the description of the CRT, which should be a verbal description of the pedagogical meaning of the CRT;
- **visual representation:** the colour and the shape is specified how a CRT is represented in the CAM component;
- **structure:** the socket structure and the properties of sockets can be specified; the properties of the sockets include name of the socket and minimum and maximum number of concepts which can be placed in a socket; furthermore there are two structures: directed and undirected;
- a directed structure means that there is a source and a target socket which are used by the CAM to define a directed structure on concepts;
- an undirected structure means that concepts in sockets are not directed. Any number of sockets is allowed for this mode.

In the "Meta-Info Tab" (Figure 59), the following CRT properties can be specified:

- **comments:** general comments of the author can be specified here,

- **creation and modification time:** time stamps are automatically stored in these fields (cannot be changed by the user), and
- **author:** the author of the CRT is automatically stored (cannot be changed by the user),

The screenshot displays the 'Meta Info' tab of the 'CRT Tool'. The tab is highlighted in green. Below the tab header, there are several input fields. The 'Comment' field is a large text area containing the text 'Parts of this CRT should be improved.'. Below it, there are three read-only fields: 'Created:', 'Modified:', and 'Author:'. Each of these fields displays the same timestamp: 'Thu Aug 13 00:59:41 GMT+0200 2009'.

Figure 59 Meta-Info Tab: Meta information can be specified here.

In the "GAL Code Tab" (Figure 60), the following CRT properties can be specified:

- **GAL code:** the GAL code defines the adaptive behaviour (the meaning of the CRT on a technical level); the code has to be input in this text area as it is done for programming languages. This piece of code will be interpreted by the GRAPPLE adaptive engine.

CRT Tool

General Meta Info **GAL Code** User Model Constraints DM Relations

Adaptation behaviour code:

```
// GAL code can be typed in here
// The GAL code defines the adaptive behaviour
```

Figure 60 GAL Code Tab: The GAL code can be input here.

CRT Tool

General Meta Info GAL Code **User Model** Constraints DM Relations

UM Variables:

- visited
- knowledge

New Remove

General

Name: knowledge

Socket:

☒ public ☒ persistent

Value:

Type: Float

Range: from 0.0 to 1.0

Default: 0.0

Location:

Web:

Course Name:

Resource Name:

Resource ID:

Figure 61 User Model Tab: User model variables used in the GAL code can be specified here.

In the "User Model Tab" (Figure 61), the user model variables can be specified which are accessed by the GAL code. A list of variables can be created, for each variable the following properties can be specified:

- ***name***: the name of the user model variable;
- ***socket***: the socket which the variable refers to [optional];
- ***public***: any system with access to the User Model service will be able to access this variable;
- ***persistent***: The current value will be stored (as opposed to computed at runtime); this can be beneficial for performance, but there may be issues with privacy policies;
- ***type***: the type of the variable, can be one of integer, float, string, enumeration, or Boolean. The type determines which values can be expressed by this value;
- ***range and default***: the range limits the range of values according to the specified type;
- ***location***: the location determines where the user model variables occurs (if it occurs in a LMS outside of GRAPPLE engine).

In the "Constraints Tab" (Figure 62), constraints can be defined which limit the usage of CRTs in the CAM component. The following constraints can be specified:

- ***loops***: this flag defines if it is allowed to create loops in the CAM component with this CRT;
- ***attributes***: this table restricts which concepts can be added to a given socket; in the example below only concepts may be added which have the value 'true' for the attribute 'visited';
- ***user model variables***: this list specifies how which CRTs in the 'neighbourhood' are allowed; CRTs accessing user model variables given in this list are not allowed as 'neighbours' in the CAM.

CRT Tool

General Meta Info GAL Code User Model **Constraints** DM Relations

Loops ☒ Loops allowed

Attributes

Socket	Attribute	Value
Source	visited	true

New Remove

User Variable

New Remove

Figure 62 Constraints Tab: Constraints can be specified here.

In the "DM-Relations Tab" (Figure 63), the relations to Domain Model can be specified:

- Domain Model relations: relations between concepts as used in Domain Models can also be exploited to connect concepts in a pedagogical way. The Domain Model relations are given as list by adding a blank relation and modifying it in the list by clicking on it.

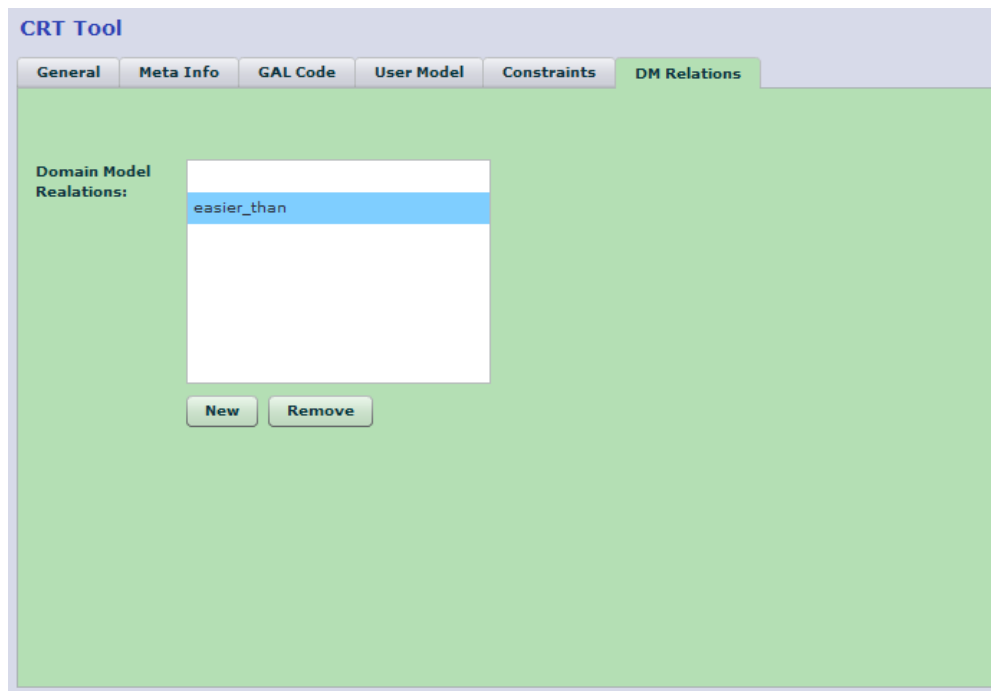


Figure 63 DM-Relations Tab: Relations to Domain Model can be specified.

I.III. Concept Adaptation Model Component

The Concept Adaptation Model is initially the third panel available to the author when he opens the GAT tool, as displayed in Figure 64. The CAM effectively contains a course. The CAM consists of all Domain Models, CRT descriptions used and their instantiations.



Figure 64 Opening the GAT tool.

The author can create a new CAM by selecting File → New on the main bar, as in Figure 65:



Figure 65 Creating a new CAM.

The author has to name the new CAM and then he can start creating the CAM model, by pointing on the CAM editor and right-click. As shown in Figure 66, he can insert a new CRT instance.

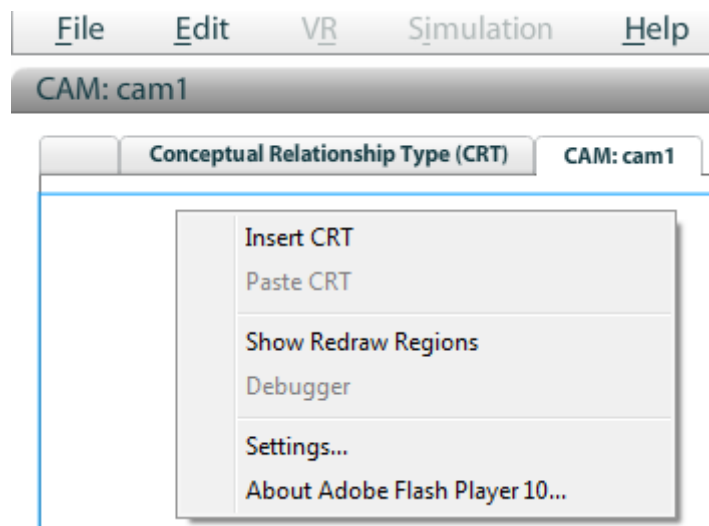


Figure 66 Inserting a CRT instance into a CAM.

I.III.I. Inserting CRTs

The author is now presented with a dialog for selecting the CRT model he wishes to insert an instance of. (See Figure 67).

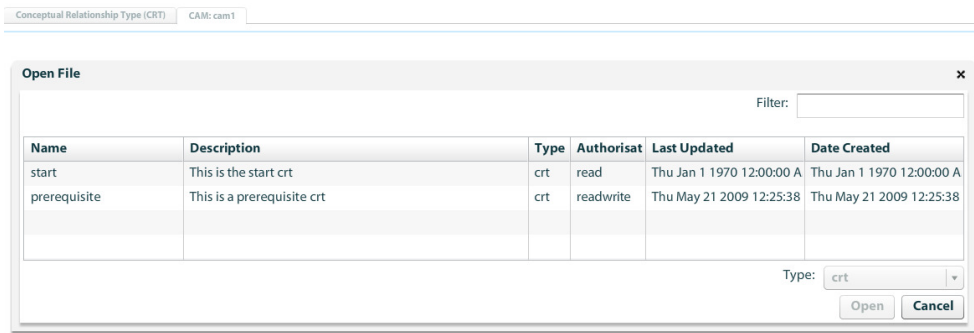


Figure 67. Choosing a CRT to insert.

After choosing a CRT to insert, A CRT node with its sockets is inserted. Initially the sockets do not contain any concepts. See Figure 68.

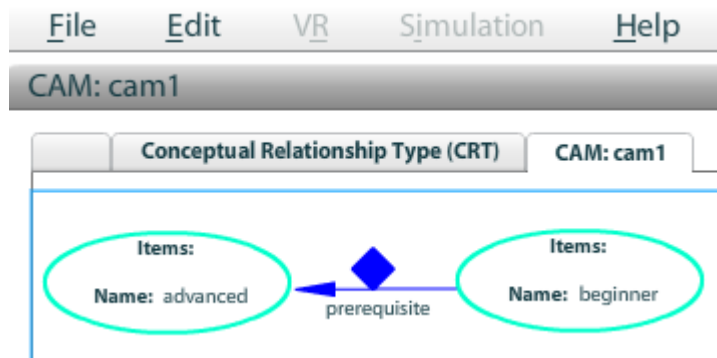


Figure 68 The prerequisite CRT, which has 2 sockets, inserted into the CAM.

As an alternative, CRTs can also be copied and pasted. This is done clicking the right mouse button on a CRT node and selecting copy CRT. Then clicking the right mouse button on an empty space on the canvas and choosing paste CRT. This pastes the CRT instance, including modified colours and shapes, but the sockets will not contain any concepts. (See Figure 70). Note that CRT instances can be copied from other CAM windows as well as the current window.

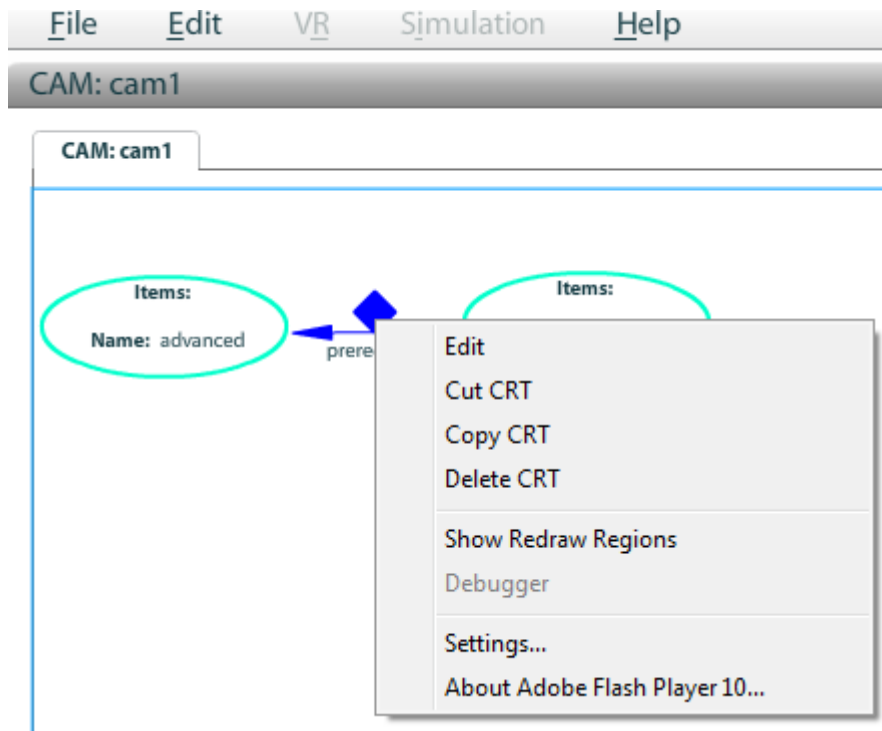


Figure 69 Copy a CRT instance. Other actions are delete and cut (copy then delete).

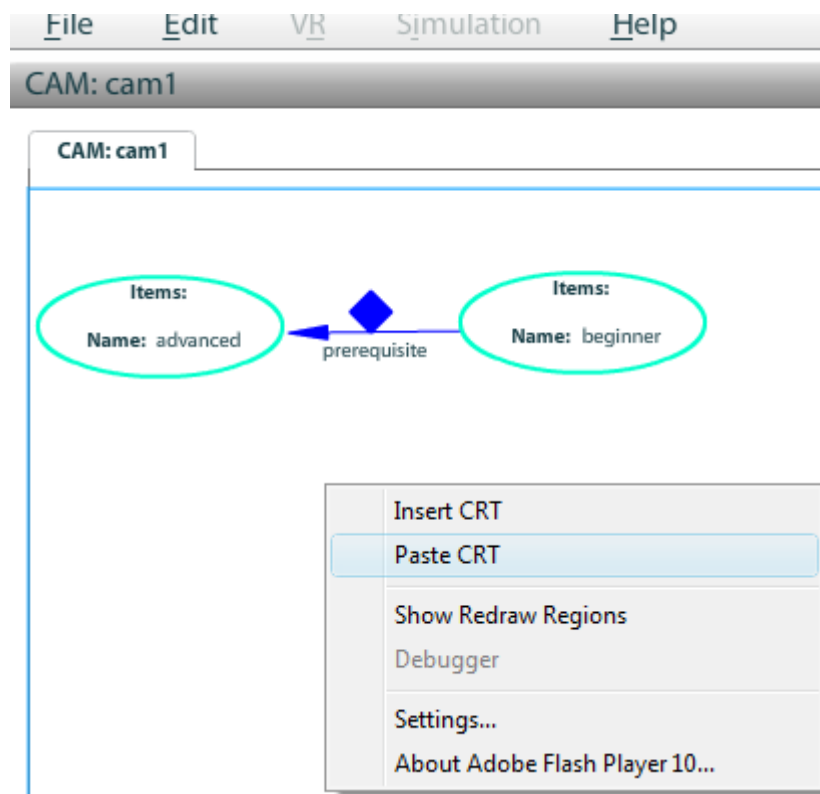


Figure 70 Past CRT is now enabled.

I.III.II. Filling Sockets with Concepts

Now the author has to fill all sockets with at least 1 concept. There are various ways of doing this.

- Copy-paste concepts from a CAM window (note: this can be a different CAM window from the current one) see Figure 71, Figure 72 and Figure 73.
- Copy-paste concepts from a Domain Model window see section I.I, Figure 72 and Figure 73.
- Through the edit socket interface see Figure 73, Figure 74 and Figure 75. This also allows insertion of external Locations, see Figure 76.

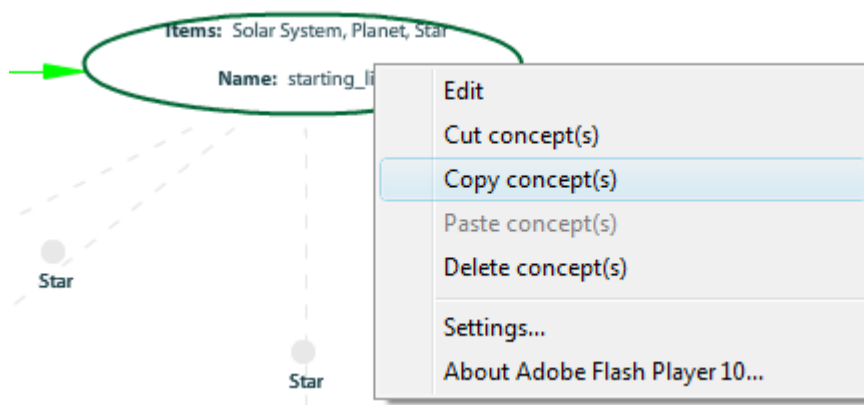


Figure 71 Copy concepts from a socket in a CAM window.

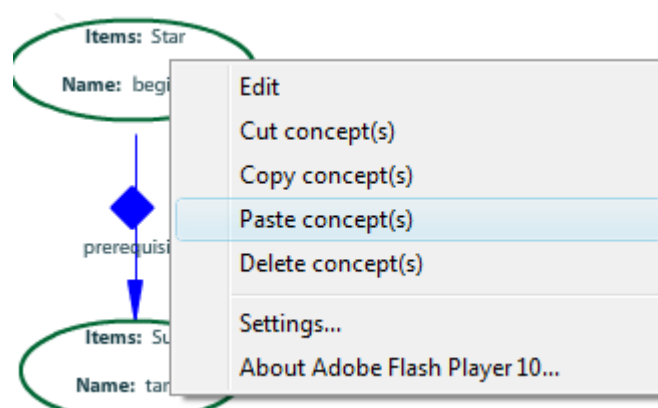


Figure 72 Pasting Concepts into sockets.

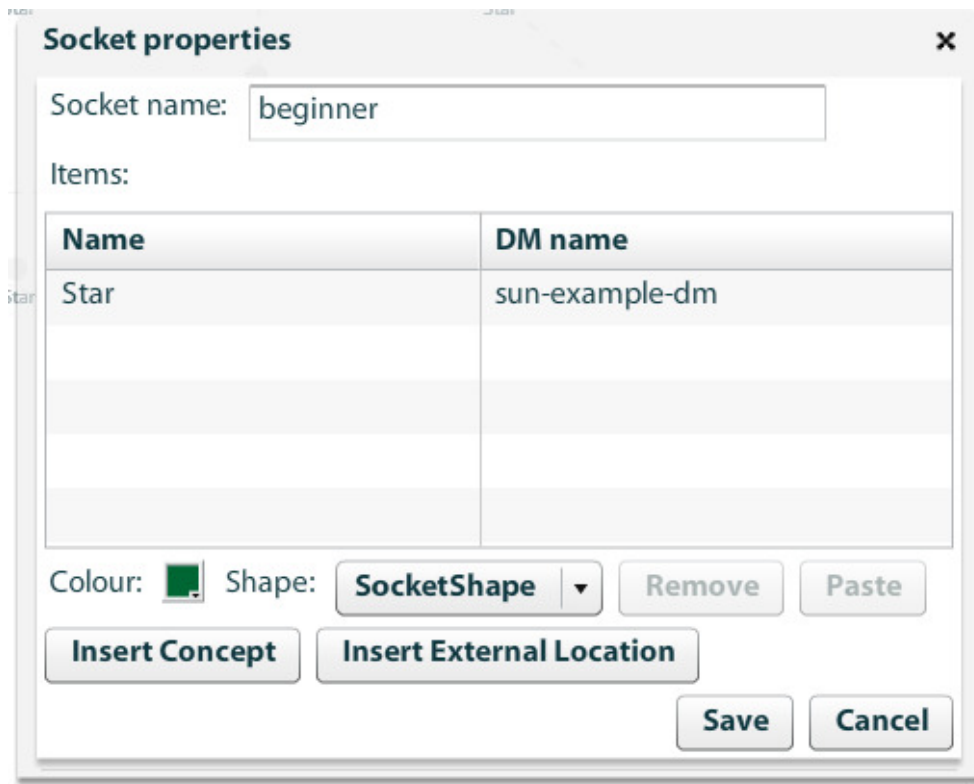


Figure 73 Socket editing screen, providing an alternate place to paste concepts and an interface to insert concepts.

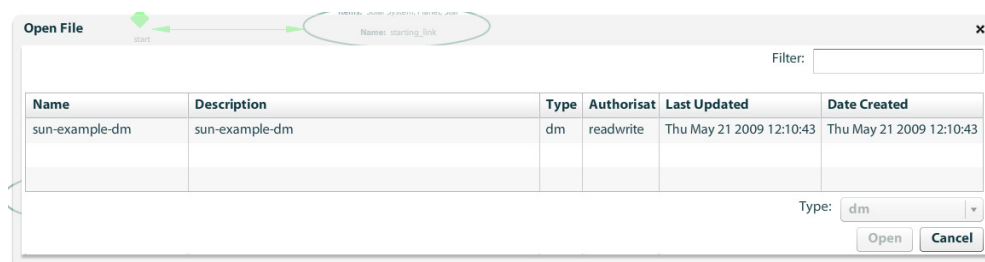


Figure 74 Selection interface for the Domain Model the concept comes from.

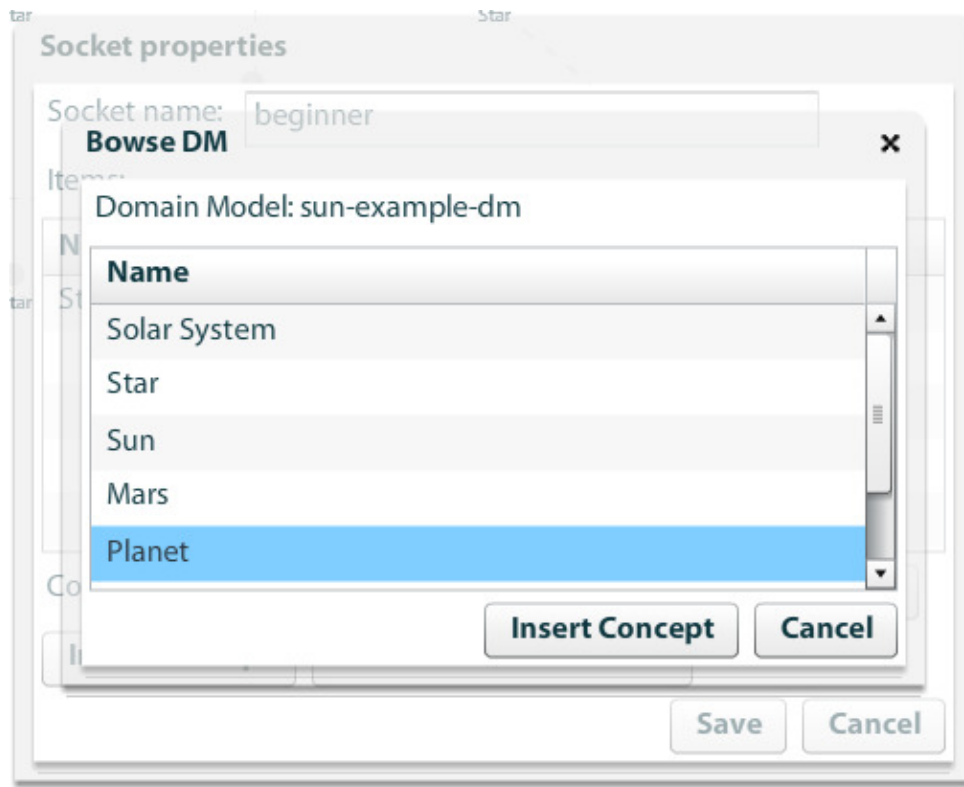


Figure 75 Selection of the Concept.

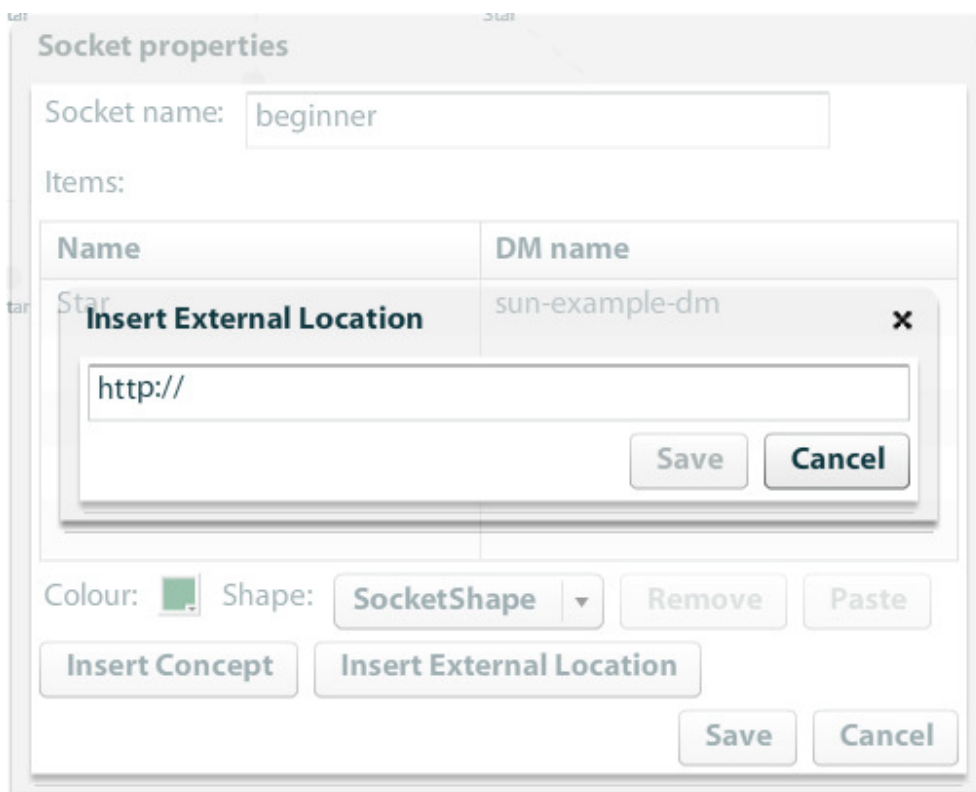


Figure 76 Insert External Location.

Note that the component will automatically link sockets that have some concepts in common, to indicate this relationship. (See Figure 77).

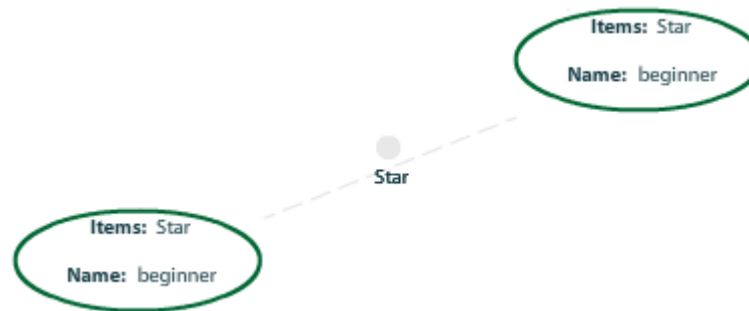


Figure 77 Two sockets containing the concepts Star

I.III.III. Editing CRTs and sockets

CRTs and Sockets can be edited by clicking the right mouse button on the socket or CRT node and choosing edit or by double clicking. The screen for editing properties of sockets is shown in Figure 73. It offers the possibility to change the name, shape and colour of the socket. It also allows pasting removing and inserting concepts and External Locations.

The screen for editing CRT instances is shown in Figure 78. The dialog allows changing the colour and shape of the CRT node. It also allows launching a new window of the CRT component with the description of this CRT instance loaded for editing. Finally it allows updating the CRT description in this model for this type of CRT from a CRT description stored in the repository.

So, in order to make changes to the description of the CRT and ultimately the adaptation code, in this CAM model an author has to open the CRT model in a CRT component window, either through locating it in the menu under File → Open in the toolbar or through clicking *Edit CRT* in the CRT editing screen. Then the author has to make the desired changes and store the CRT description – either under the same or a different name. Finally the author has to navigate to an instance of the CRT model that he wanted to change and click *Update CRT*. At this stage the author has to locate the CRT through a dialog which is

identical to the one under File → Open in the toolbar. This dialog however only lists CRTs that will fit in this place.

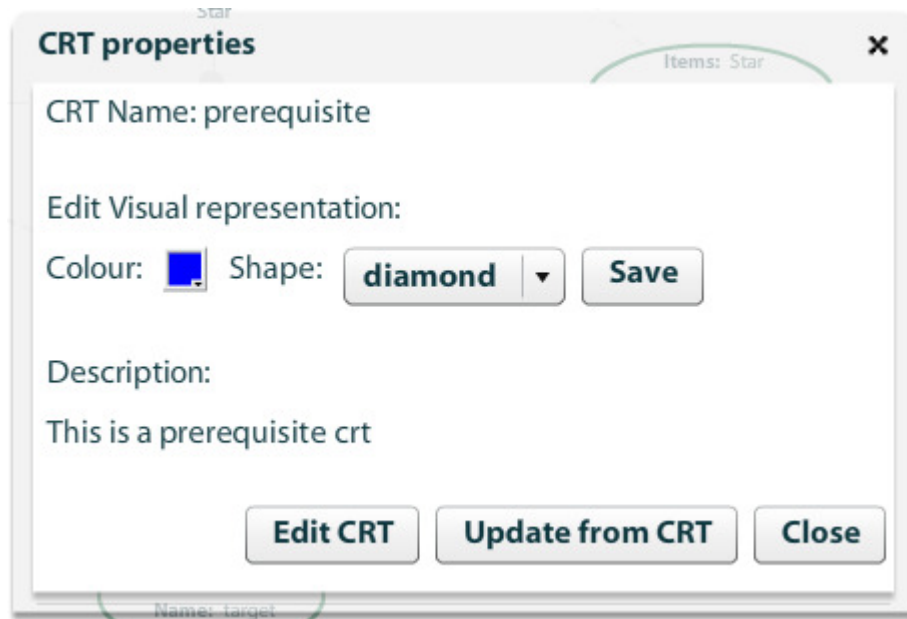


Figure 78 CRT editing screen.

I.III.IV. Deleting Concepts and CRTs

Concepts can be deleted by clicking the right mouse button on a socket and selecting delete concepts, see Figure 79. This deletes the concepts from the socket. Alternatively in the socket edit dialog, see Figure 73, the author can select any number of concepts and delete only the selected ones by clicking *remove* and then *save*.

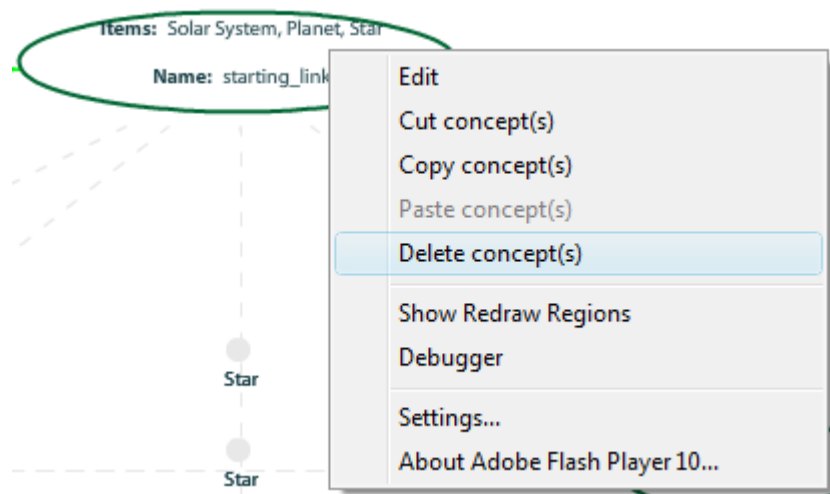


Figure 79 Deleting all concepts from a socket.

I.IV. GRAPPLE Authoring Tool Tutorial

This tutorial will illustrate the integrated authoring process step-by-step. The authoring process is explained referring to a demonstration scenario at the high school level. The example used is based on the domain of our solar system in the framework of a basic physics course. The domain will consist of the concepts *Planet*, *Jupiter*, *Earth*, *Mars*, *Star*, *Sun* and *Solar System*. As a teaching strategy we will take a globalist approach, which means that the learner should first look at the overall picture and later on details. First we will introduce general concepts *Solar System*, *Planet*, *Star* and then their specific instances *Jupiter*, *Earth*, *Mars* and *Sun*. Note that this choice in the tutorial does not imply this is the best teaching strategy it is merely to illustrate the authoring process.

I.IV.I. Creating a Domain Model – DM component

The first step in creating an Adaptive Story Line with the GRAPPLE Authoring Tool is creation of a Domain Model. The Domain Model will contain the concepts that eventually end up in the course and are shown to the student. First, you have to launch the GAT tool. The tool starts with 3 tabs: DM component, CRT and CAM component. The DM component tab is selected by default, see Figure 80.

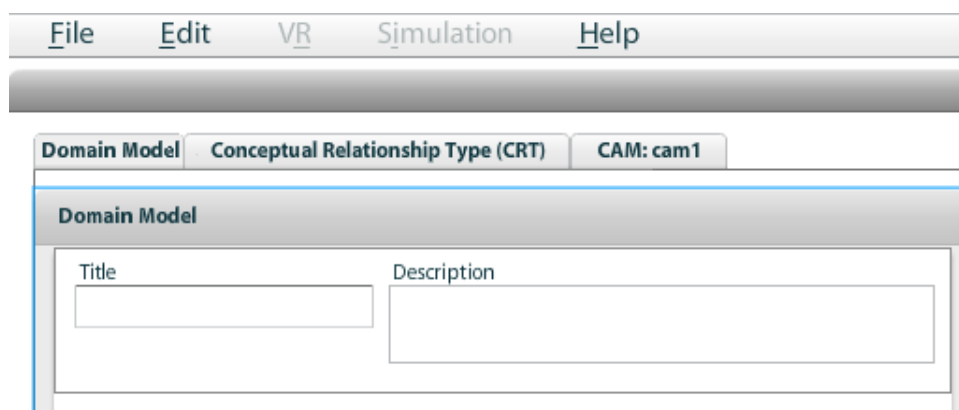


Figure 80 Opening the GAT tool.

Now you can give the course a title and description. Input *Astronomy* in the title field and specify the Description field. Now that the empty DM has a name, we can save it. On the menu bar select File → Save, as shown in Figure 81.

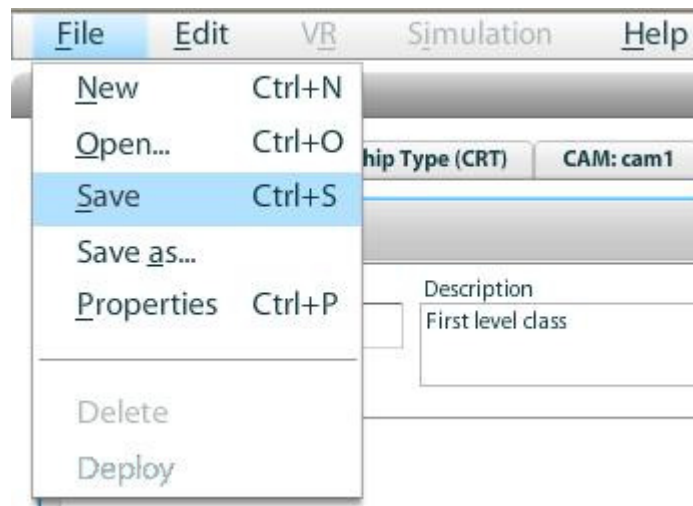


Figure 81 Save a new Domain Model.

Now you can change the properties of the model by selecting File → Properties from the menu bar, see Figure 82. Change the *Authorisation* to *readwrite* to give other authors full access to the DM.

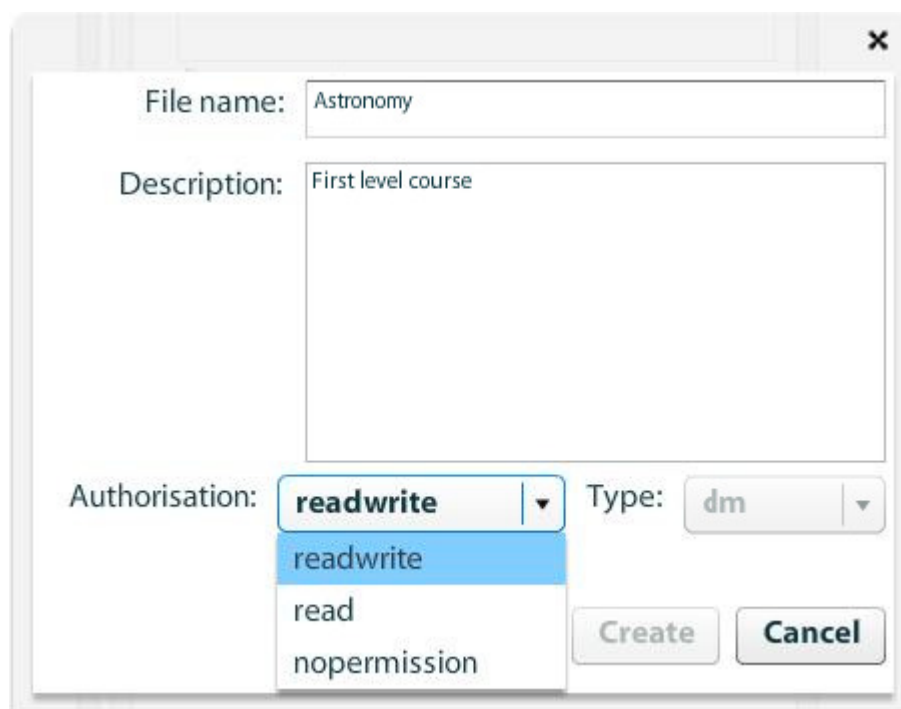


Figure 82 Definition of a new Domain Model.

Now we can start creating concepts for our new DM called Astronomy. We insert our first concept *Star* by right-clicking on the Domain Model panel and selecting the option Add Concept, as shown in Figure 83.

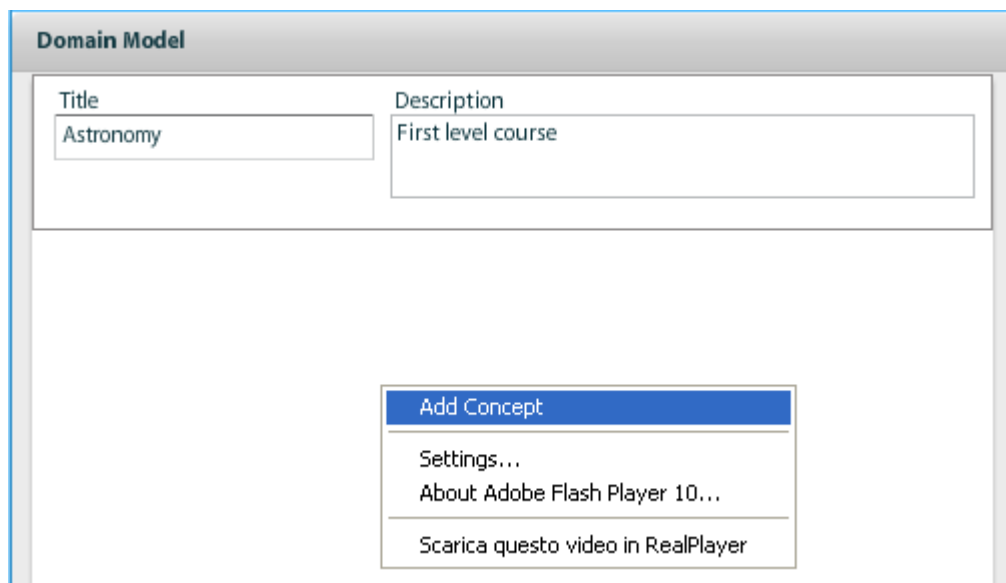


Figure 83 Adding a new concept.

After the first concept is inserted, click on the name field on the right hand side, and type the name *Star* as shown in Figure 84.

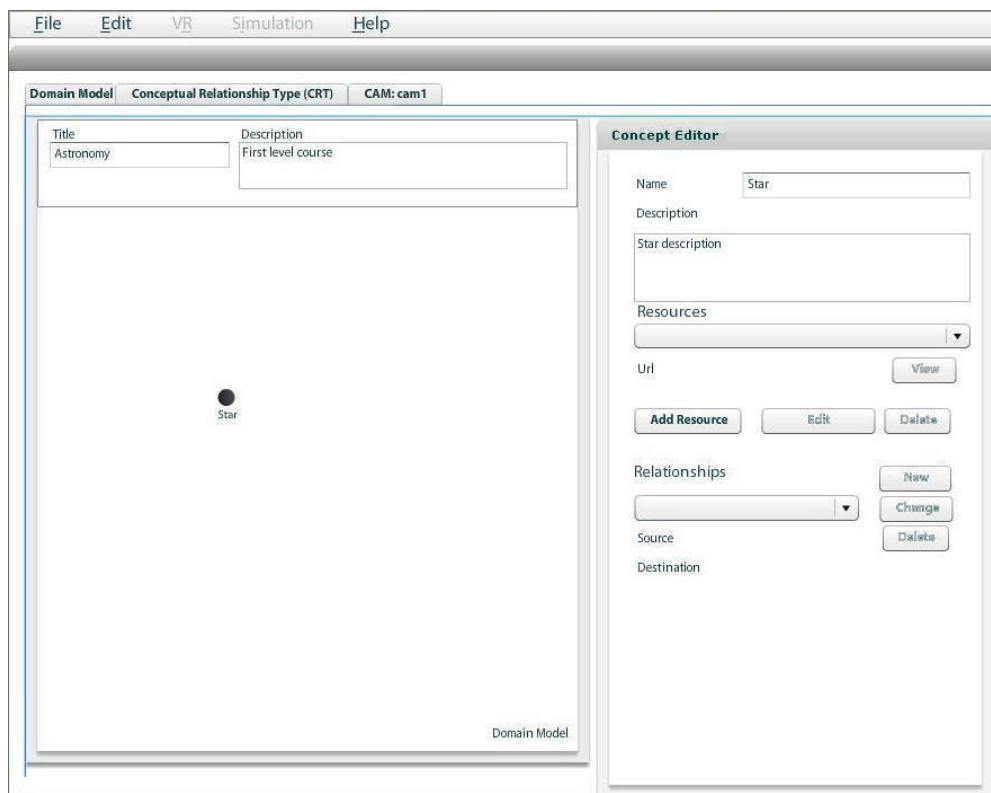


Figure 84 Creation of a new concept.

The next important step is to assign a resource to a single concept. Click the Add Resource button. In the screen shown in Figure 85 and Figure 86 you can insert any URL, for example insert the Wikipedia page on Star (<http://en.wikipedia.org/wiki/Star>) and press save.

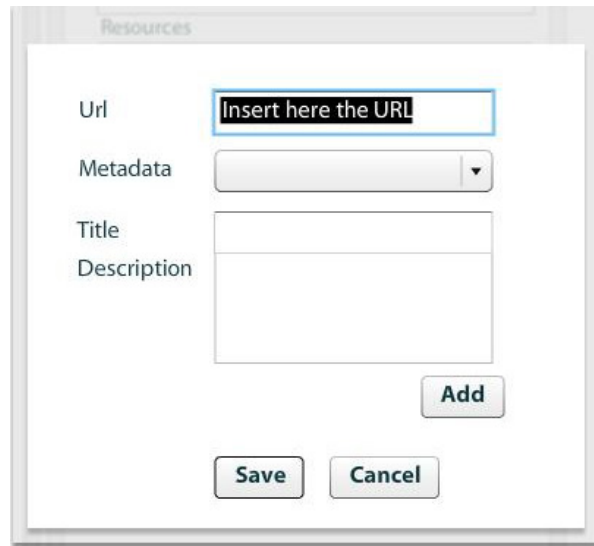


Figure 85 Resource addition.

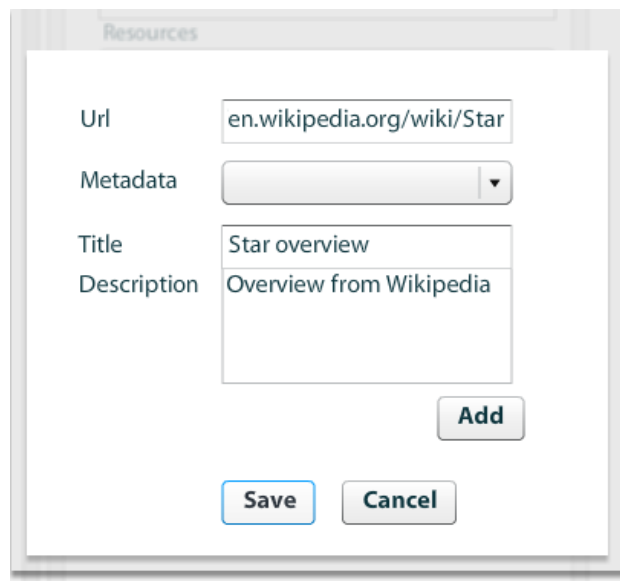


Figure 86 Resource window filling.

Now the concept Star is defined. Repeat the above process for the concepts *Planet*, *Jupiter*, *Earth*, *Mars*, *Sun* and *Solar System*.

The final step to complete our DM is to insert the relationships between the concepts. These relationships do not yet give any behaviour but are merely semantic links to aid the author. Click on *New* on the right in the section *Relationships* (Figure 84). A window will pop up in which you can select the relationship. Now select the Concept *Sun* as source and *Star* as Target, select the relationship *is-a* and click *Save* (Figure 87).

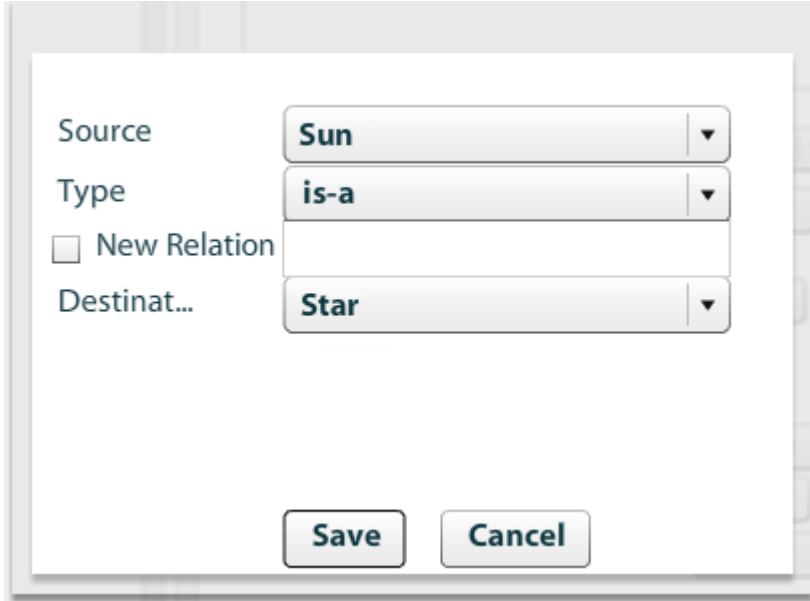
A screenshot of a dialog box titled 'Insert Relationship'. It contains four labels on the left: 'Source', 'Type', 'New Relation', and 'Destinat...'. To the right of 'Source' is a dropdown menu with 'Sun' selected. To the right of 'Type' is a dropdown menu with 'is-a' selected. To the right of 'Destinat...' is a dropdown menu with 'Star' selected. There is an unchecked checkbox next to 'New Relation'. At the bottom are two buttons: 'Save' and 'Cancel'.

Figure 87 Inserting is-a relation between Sun and Star

A set of predefined relationships are available from a drop down list: *is-composed-by*, *is-a* and *belong-to* (Figure 88).

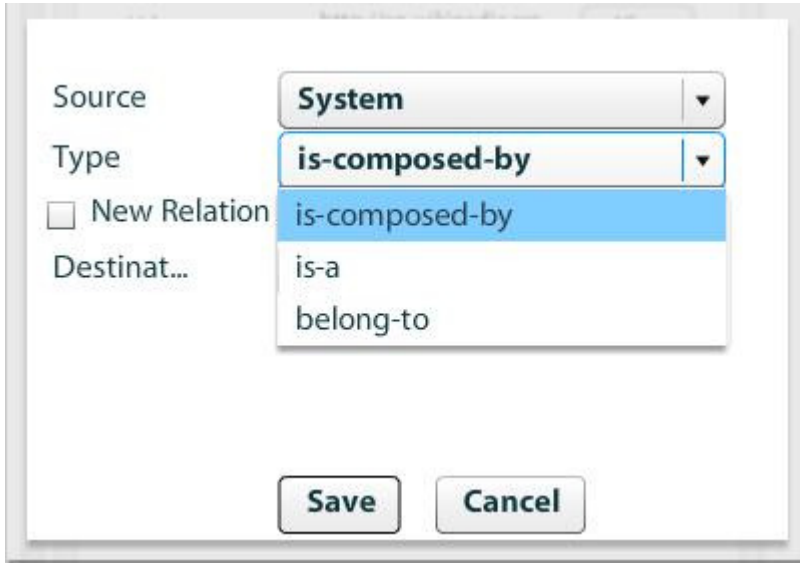
A screenshot of a dialog box titled 'Insert Relationship'. It contains four labels on the left: 'Source', 'Type', 'New Relation', and 'Destinat...'. To the right of 'Source' is a dropdown menu with 'System' selected. To the right of 'Type' is a dropdown menu with 'is-composed-by' selected. To the right of 'Destinat...' is a dropdown menu with 'is-a' and 'belong-to' visible. There is an unchecked checkbox next to 'New Relation'. At the bottom are two buttons: 'Save' and 'Cancel'.

Figure 88 Relationship definition.

In a similar fashion insert the relationship *is-composed-by* between the concepts *Solar System* and the concepts *Sun*, *Earth*, *Jupiter* and *Mars*. Also insert the *is-a* relationship between the source concepts *Earth*, *Jupiter* and *Mars* and the destination concept *Planet*. The DM now looks similar to Figure 89 and you have completed the DM.

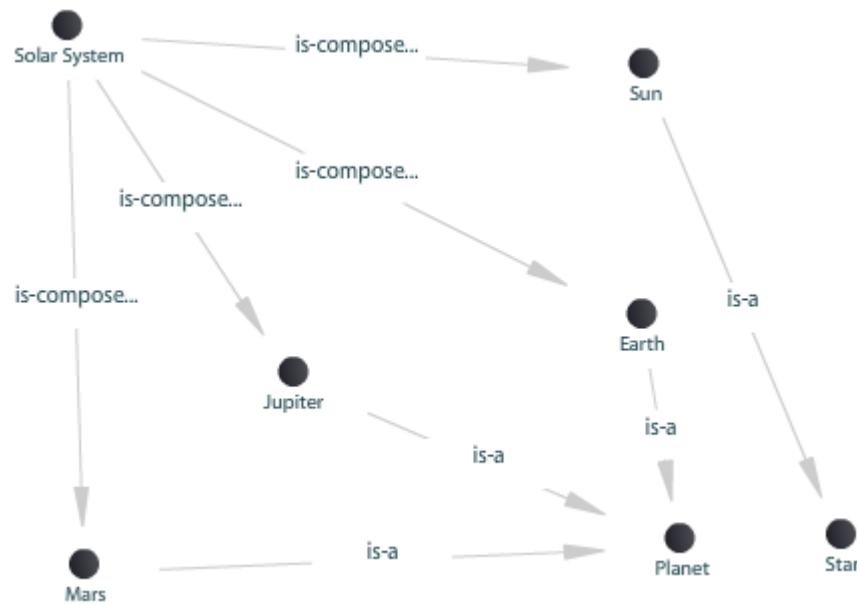


Figure 89 Solar System DM.

I.IV.II. Creating a Teaching Strategy – CRT component

The next step in the authoring process is creating a teaching strategy by means of Concept Relationship Types. The CRTs effectively define the teaching strategy. The bad news is that (some parts of) this step may be too complicated for some authors. The good news is that not everybody has to create new CRTs as one can choose from the existing ones.

In this part of the tutorial we are going to create CRTs in order to realise the globalist teaching strategy (first look at the overall picture, then on details). In order to achieve this teaching strategy we are going to create a prerequisite CRT. In the CRT component in the menu bar choose *File* → *New*, then as type CRT and as name *prereq* (see Figure 90). To make the CRT editable to other authors select *readwrite* as authorisation.

File name:

Description:

Authorisation: Type:

Figure 90 Creating new prereq CRT.

The CRT has two sockets and is directed. Both sockets have cardinality * (many). So the general tab of the CRT component will look like Figure 91.

CRT Tool

General Meta Info GAL Code User Model Constraints DM Relations

Title:

Description:

Colour / Shape:

Structure: ☒ Directed ☐ Undirected

Type: <i>Source Socket</i>	Type: <i>Target Socket</i>
Name: <input type="text"/>	Name: <input type="text"/>
Min Cardinality: <input type="button" value="*"/>	Min Cardinality: <input type="button" value="*"/>
Max Cardinality: <input type="button" value="*"/>	Max Cardinality: <input type="button" value="*"/>

Figure 91 CRT general tab.

Next we will insert the User Model variables that we will use in our CRT. Our CRT is based on the knowledge variable. The final CRT required is the CRT that will make sure the knowledge variable is effectively modelled. In this tutorial we choose a simplified model of knowledge where a student has full knowledge when he has accessed a concept and no knowledge otherwise. In the User Model tab insert the variable knowledge by clicking on *new* and typing in knowledge in the name field. Select public and persistent in order to be able to share this variable with other courses and other systems. Insert a range from 0 to 100 and a default of 0. Repeat the process for both the source and target sockets. See Figure 92.

The screenshot shows the 'CRT Tool' interface with the 'User Model' tab selected. On the left, under 'UM Variables:', there is a list containing 'Knowledge' and 'Knnowledge', with 'Knowledge' selected. Below this list are 'New' and 'Remove' buttons. The main area is divided into sections: 'General', 'Value:', and 'Location:'. Under 'General', the 'Name:' field contains 'Knowledge', the 'Socket:' dropdown is set to 'Source', and both 'public' and 'persistent' checkboxes are checked. Under 'Value:', the 'Type:' dropdown is set to 'Integer', the 'Range:' is from '0' to '100', and the 'Default:' is '0'. Under 'Location:', there are four empty text input fields for 'Web:', 'Course Name:', 'Resource Name:', and 'Resource ID:'.

Figure 92 CRT is based on the Knowledge variable.

A prerequisite relation should not occur in a loop. For example if *Solar System* is a prerequisite for *Star* then *Star* should not be a prerequisite for *Solar System*, or a prerequisite for a concept that is in a number of steps a prerequisite for *Solar System*. Therefore we open the Constraints tab and unselect *Loops allowed* (see Figure 93). Finally we will insert the adaptation code in the GAL language that actually makes the adaptive behaviour happen (see Figure 94).

CRT Tool

General Meta Info **GAL Code** User Model Constraints DM Relations

Loops ☐ Loops allowed

Socket	Attribute	Value

New Remove

User Variable

New Remove

Figure 93 Deselect loops allowed.

CRT Tool

General Meta Info **GAL Code** User Model Constraints DM Relations

Adaptation behaviour code:

```
FOR ALL (Select ?c
  WHERE
    ?c IN TARGET
    AND
    ((select COUNT(?s) IN SOURCE WHERE ?s.knowledge<50) == 0)
)
```

Figure 94 GAL code for prereq.

Our strategy relies upon the knowledge variable. Therefore we need to make sure the knowledge variable models students' knowledge of concepts of the course. We will do that using 2 unary CRTs (undirected CRTs with only 1 socket). First we will create the start CRT, which will set the knowledge variable for all concepts in its only socket to 100, this

will make sure that these concepts can be starting points of the course (Figure 95 to Figure 97).

CRT Tool

General | Meta Info | GAL Code | User Model | Constraints | DM Relations

Title: start

Description: This is the start CRT from the tutorial

Colour / Shape: Diamond [Green Swatch]

Structure: ☐ Directed ☒ Undirected

Type	Name	Min Cardinality	Max Cardinality
start_socket	start	*	*

start_socket : start [*,*]

New Remove

Figure 95 Creating the start CRT.

CRT Tool

General | Meta Info | GAL Code | **User Model** | Constraints | DM Relations

UM Variables:

- Knowledge

New Remove

General

Name: Knowledge

Socket: [Dropdown]

☒ public ☒ persistent

Value:

Type: Integer [Dropdown]

Range: from 0 to 100

Default: 0

Location:

Web: [Text Field]

Course Name: [Text Field]

Resource Name: [Text Field]

Resource ID: [Text Field]

Figure 96 The knowledge variable is also used by the start CRT.

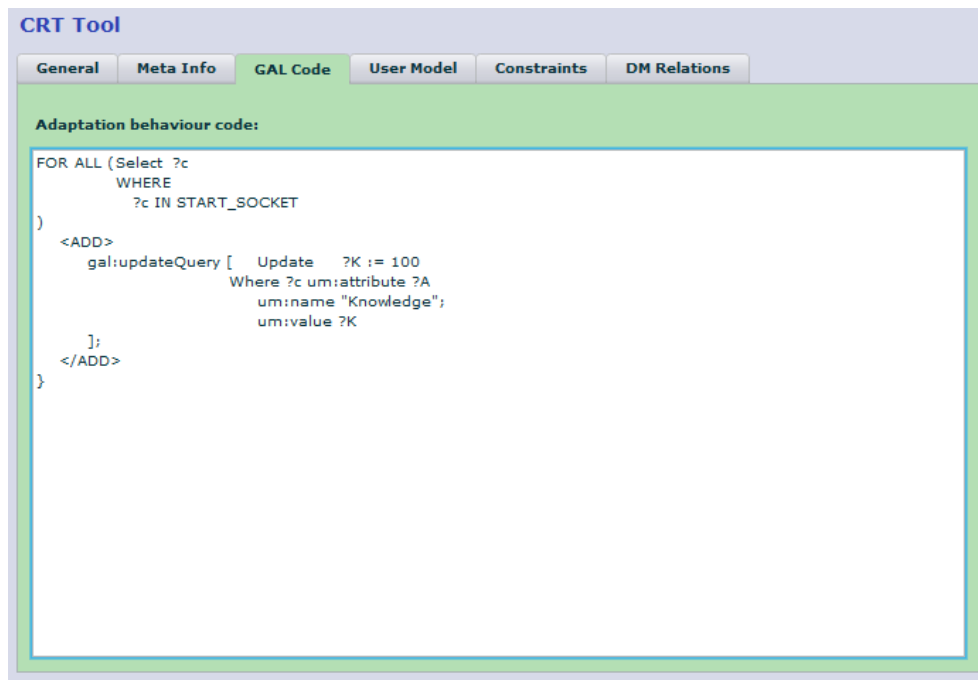


Figure 97 The GAL code for the start CRT assigns 100 to the knowledge for all concepts in the start_socket.

The final CRT required is the CRT that will make sure the knowledge variable is effectively modelled. This CRT is also unary (Figure 98- Figure 100).

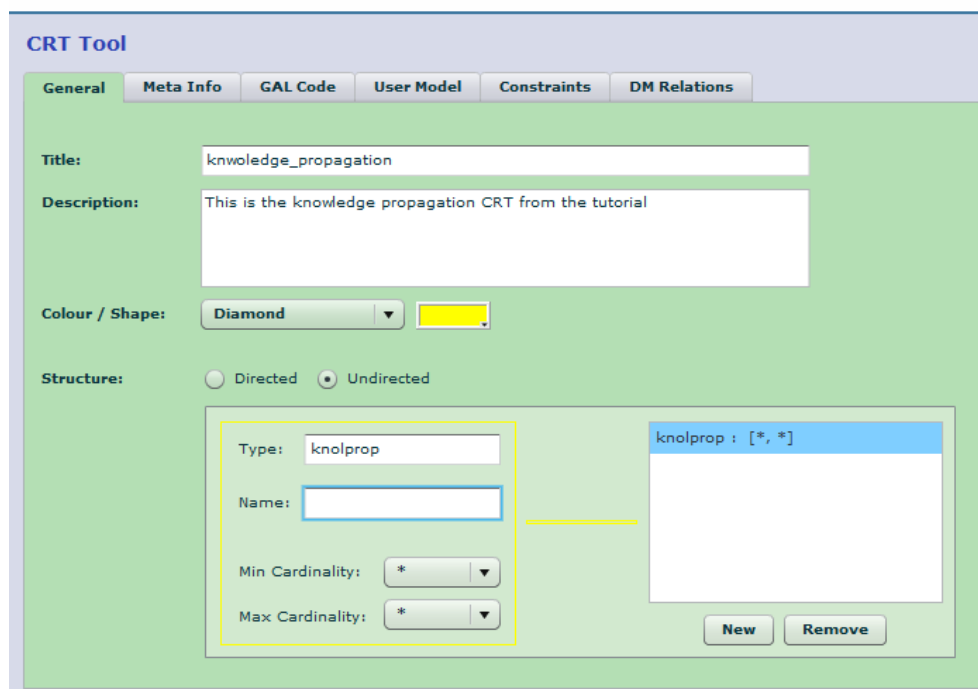


Figure 98 The knowledge propagation CRT.

The screenshot shows the 'User Model' tab in the 'CRT Tool'. On the left, under 'UM Variables:', there is a list containing 'Knowledge' and buttons for 'New' and 'Remove'. The main area is divided into 'General' and 'Value' sections. In the 'General' section, 'Name:' is 'Knwoledge', 'Socket:' is a dropdown, and there are checkboxes for 'public' and 'persistent', both of which are checked. In the 'Value' section, 'Type:' is 'Integer', 'Range: from' is '0' and 'to' is '100', and 'Default:' is '0'. At the bottom, the 'Location:' section has fields for 'Web:', 'Course Name:', 'Resource Name:', and 'Resource ID:'.

Figure 99 Also this CRT uses the knowledge variable.

The screenshot shows the 'GAL Code' tab in the 'CRT Tool'. It displays 'Adaptation behaviour code:' followed by a code block. The code is as follows:

```

FOR ALL (Select ?C
  Where ?c IN knolprop
) {
  <ADD>
    gal:onAccess [
      gal:updateQuery [ Update ?K := 100
                        Where ?c um:attribute ?A
                        um:name "Knowledge";
                        um:value ?K
                      ];
    ];
  </ADD>
}

```

Figure 100 GAL code for the knowledge propagation CRT.

I.IV.III. Creating an Adaptive Course – CAM component

In this final section of the tutorial we will see how we use the conceptual domain and teaching strategy to create the actual course. First, create a new CAM by selecting File → New on the main bar, as in Figure 101.



Figure 101 Creating a new CAM.

Name the CAM *Solar System* and change the *Authorisation* to *readwrite* to give other authors full access to the DM see Figure 102.

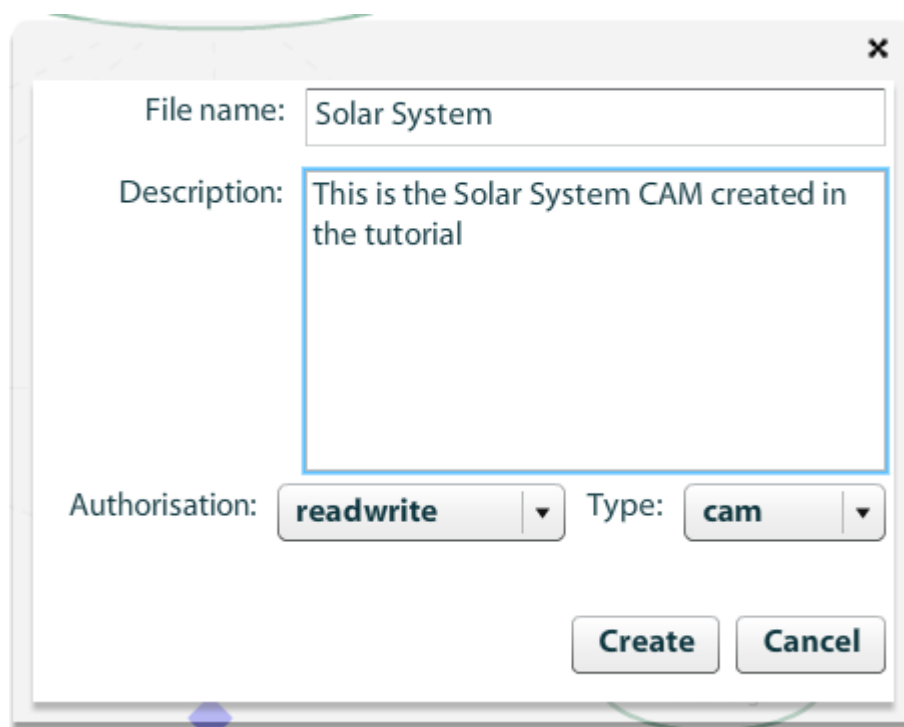


Figure 102 Creating Solar System CAM.

Now we can start creating the CAM model, by pointing on the CAM editor and right-clicking we can insert CRT instances, as shown in Figure 103 and Figure 104. We will insert 1 instance of the *start* CRT, 1 instance of the *knowledge_propagation* CRT and 2 instances of the *prereq* CRT. The result will look as in Figure 96.

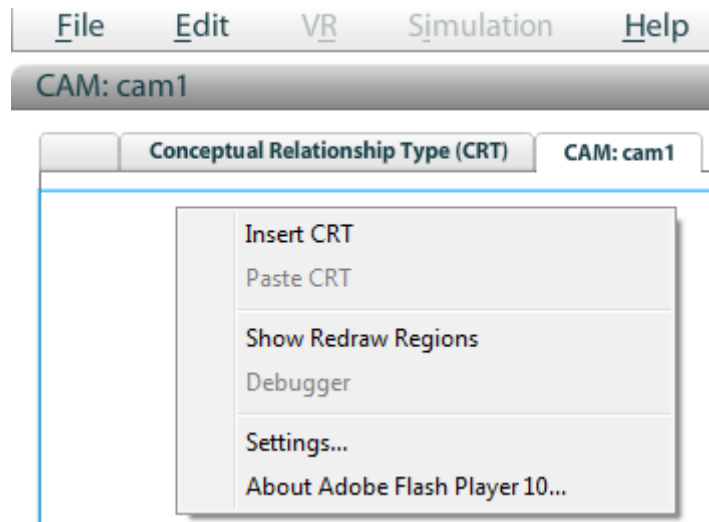


Figure 103 Inserting a CRT instance into a CAM.

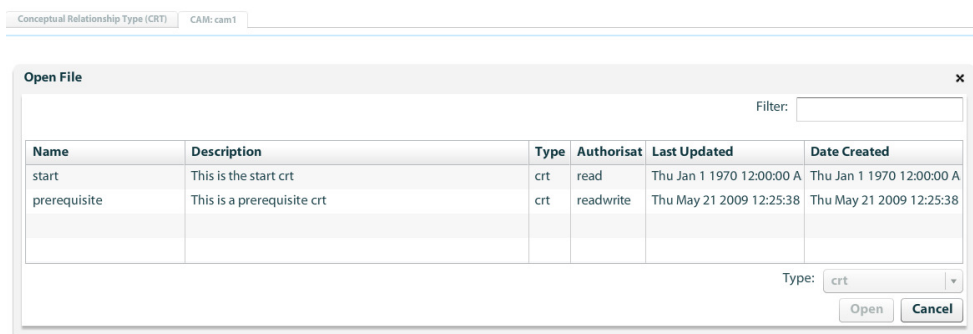


Figure 104 Choosing a CRT to insert.

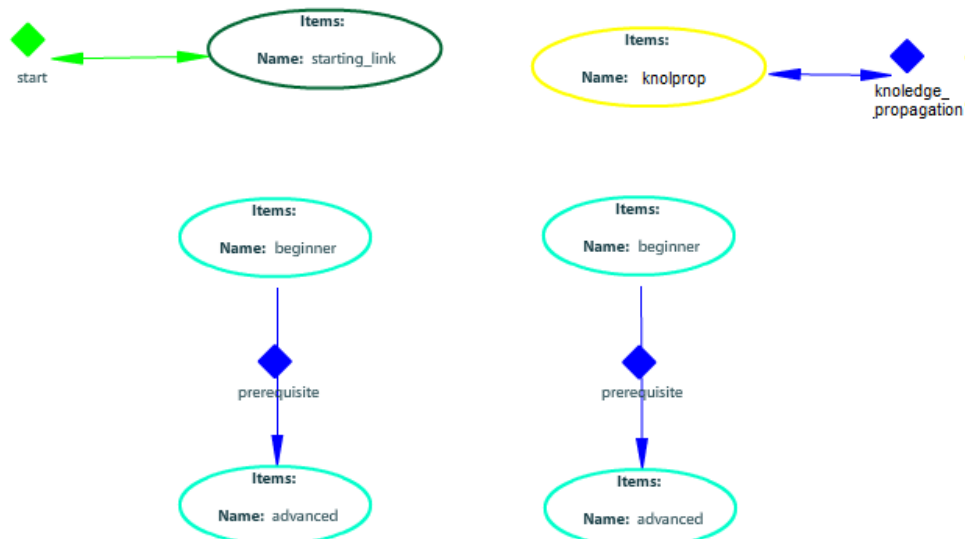


Figure 105 The CAM with the CRT instances inserted.

The final step is to insert the appropriate concepts in the appropriate sockets – first, the start CRT. We will insert the concepts *Solar System*, *Planet* and *Star* as starting points of the course. On the tab bar just below the menu bar navigate to the Domain Model tab (and if you have closed it, load the Solar System Domain Model using File → Open on the menu bar). In the DM tab right click on the concept *Solar System* and select *copy* as shown in Figure 106.

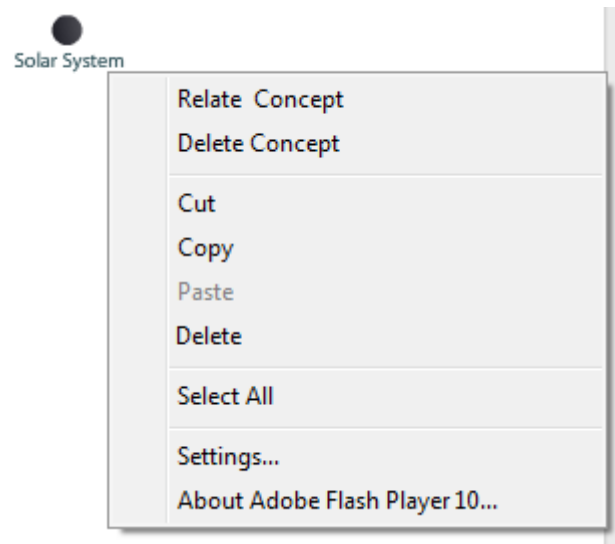


Figure 106 Copying the Solar System concept.

Now navigate back to the CAM tab which holds the Solar System CAM. Here right click on the socket belonging to the *Start* CRT and click *Paste concept(s)* as shown in Figure 107.

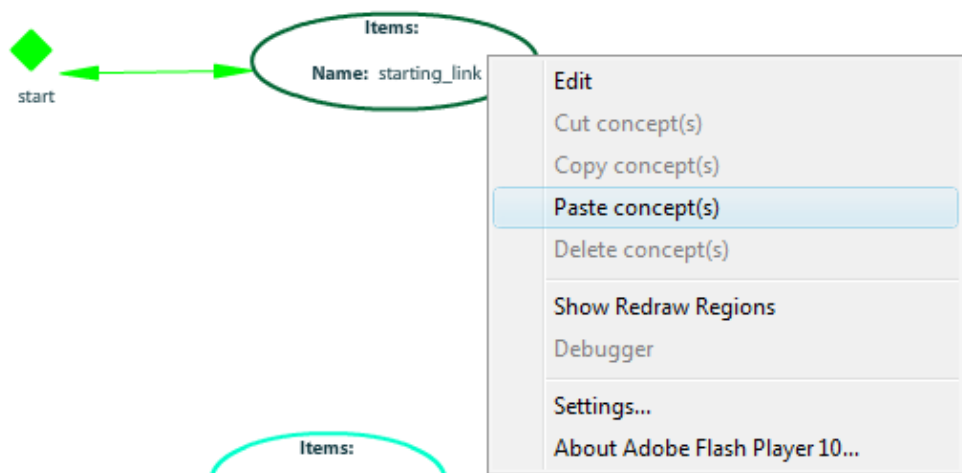


Figure 107 Inserting the Solar System concept.

In a similar fashion, insert all concepts into the knowledge propagation socket. Also insert *Star* in one of the source sockets and *Sun* in the target socket belonging to that CRT instance. Now insert the concept *Planet* in the other source socket and in the remaining target socket insert the concepts *Earth*, *Jupiter* and *Mars*. We have now completed the CAM (Figure 108). Do not forget to save the CAM by selecting *File* → *Save* from the menu bar.

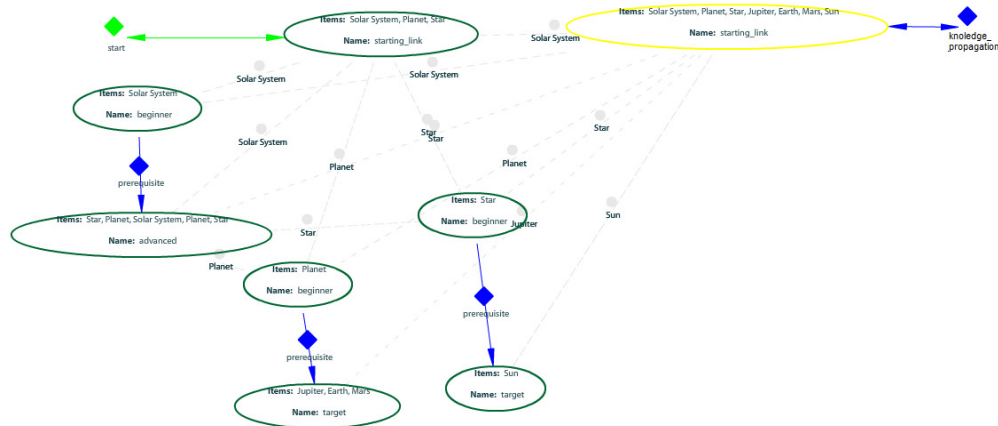


Figure 108 Completed CAM.

Appendix II. XML Schema of CAM Internal Language

This section contains the XML schema specification of the CAM XML format. CAMs which are created by the CAM component are valid against this schema specification. This specification is supposed to be used by all GRAPPLE tools and components which read CAMs. An example of a CAM which implements this specification can be found in Appendix III.

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.example.org/CAM-external-oct2009"
  xmlns:tns="http://www.example.org/CAM-external-oct2009"
  elementFormDefault="qualified"
  xmlns:Q1="http://www.imsglobal.org/xsd/imsvdex_v1p0"
  xmlns:Q2="http://grapple-project.org/GAT/" xmlns:Q3="http://www.grapple-
project.org">

  <import schemaLocation="dm-vdex-extensions.xsd"
    namespace="http://www.grapple-project.org"></import>

  <import schemaLocation="CRTOct2009.xsd" namespace="http://grapple-
project.org/GAT/"></import>

  <import schemaLocation="vdex.xsd"
    namespace="http://www.imsglobal.org/xsd/imsvdex_v1p0"></import>

  <element name="model" type="tns:modelType"></element>

  <complexType name="modelType">
    <sequence maxOccurs="1" minOccurs="1">
      <element name="header" type="tns:headerType" maxOccurs="1"
minOccurs="1"></element>
      <element name="body" type="tns:camBodyType" maxOccurs="1"
minOccurs="1"></element>
    </sequence>
  </complexType>

  <complexType name="headerType">
    <sequence>
      <element name="modeluuid" type="tns:UUIDtype" maxOccurs="1"
minOccurs="1">
        </element>
      <element name="modeltype" type="string" maxOccurs="1">

```

```

        minOccurs="1">
        </element>
        <element name="authoruuid" type="tns:UUIDtype" maxOccurs="1"
minOccurs="1"></element>
        <element name="authorisation" type="tns:authorisationType"
maxOccurs="1" minOccurs="1"></element>
        <element name="creationtime" type="dateTime" maxOccurs="1"
minOccurs="1"></element>
        <element name="updatetime" type="dateTime" maxOccurs="1"
minOccurs="1"></element>
        <element name="title" type="string" maxOccurs="1"
minOccurs="1"></element>
        <element name="description" type="string" maxOccurs="1"
minOccurs="1"></element>
    </sequence>
</complexType>

<complexType name="camBodyType">
    <sequence>
        <element name="cam" type="tns:camType" maxOccurs="1"
minOccurs="1"></element>
    </sequence>
</complexType>

<simpleType name="authorisationType">
    <restriction base="string">
        <enumeration value="read"></enumeration>
        <enumeration value="write"></enumeration>
        <enumeration value="readwrite"></enumeration>
    </restriction>
</simpleType>

<simpleType name="UUIDtype">
    <restriction base="string">
        <pattern
            value="[a-z0-9]{8}\-[a-z0-9]{4}\-[a-z0-9]{4}\-[a-z0-
9]{4}\-[a-z0-9]{12}">
        </pattern>
    </restriction>
</simpleType>

```

```

<complexType name="camType">
  <sequence>
    <element name="camInternal"
type="tns:camInternalType"></element>
  </sequence>
</complexType>

<complexType name="camInternalType">
  <sequence>
    <element name="domainModel" type="tns:dmModelType"
maxOccurs="1"
      minOccurs="1">
    </element>
    <element name="crtModel" type="tns:crtModelType"
maxOccurs="1"
      minOccurs="1">
    </element>
    <element name="crt" type="string" maxOccurs="unbounded"
minOccurs="0"></element>
  </sequence>
</complexType>

<complexType name="positionType">
  <sequence>
    <element name="x" type="int"></element>
    <element name="y" type="int"></element>
  </sequence>
</complexType>

<complexType name="camSocketType">
  <sequence>
    <element name="caption" type="string" maxOccurs="1"
      minOccurs="0">
    </element>
    <element name="uuid" type="tns:UUIDtype" maxOccurs="1"
      minOccurs="1">
    </element>
    <element name="socketId" type="tns:UUIDtype" maxOccurs="1"
      minOccurs="1">
    </element>
    <element name="position" type="tns:positionType"
      maxOccurs="1" minOccurs="1">

```

```

        </element>

        <element name="shape" type="string" maxOccurs="1"
minOccurs="0"></element>

        <element name="colour" type="string" maxOccurs="1"
minOccurs="0"></element>

        <element name="entity " type="tns:entityType"
maxOccurs="unbounded" minOccurs="0"></element>

    </sequence>
</complexType>

<complexType name="entityType">
    <sequence>
        <element name="dmID" type="tns:UUIDtype" maxOccurs="1"
minOccurs="0">
            </element>
        <element name="label" type="string" maxOccurs="unbounded"
minOccurs="0">
            </element>
        <element name="relationshipType " type="string" maxOccurs="1"
minOccurs="0"></element>
        <element name="position" type="tns:positionType"
maxOccurs="1" minOccurs="0"></element>
        <element name="size" type="int" maxOccurs="1"
minOccurs="0"></element>
        <element name="shape" type="string" maxOccurs="1"
minOccurs="0"></element>
        <element name="image" type="string" maxOccurs="1"
minOccurs="0"></element>
        <element name="colour" type="string" maxOccurs="1"
minOccurs="0"></element>
    </sequence>
</complexType>

<complexType name="dmModelType">
    <sequence>
        <element name="model" type="tns:dmModelType2"
maxOccurs="unbounded" minOccurs="0"></element>
    </sequence>
</complexType>

<complexType name="dmModelType2">
    <sequence>
        <element name="header" type="tns:headerType"></element>
    </sequence>
</complexType>

```

```

        <element name="body" type="tns:dmType"></element>
    </sequence>
</complexType>

<complexType name="dmType">
    <sequence>
        <element name="vdex" type="Q1:vdexType" maxOccurs="1"
minOccurs="1"></element>
    </sequence>
</complexType>

<complexType name="crtModelType">
    <sequence>
        <element name="model" type="tns:crtModelType2"
maxOccurs="unbounded" minOccurs="0"></element>
    </sequence>
</complexType>

<complexType name="crtModelType2">
    <sequence>
        <element name="header" type="tns:headerType" maxOccurs="1"
minOccurs="1">
            <sequence>
                <element name="body" type="tns:crtType" maxOccurs="1"
minOccurs="1"><complexType></complexType></element>
            </sequence>
        </element>
    </sequence>
</complexType>

<complexType name="crtType">
    <sequence>
        <element name="crtDialect" type="Q2:crtDialectType"
maxOccurs="1" minOccurs="1">
            <sequence>
                <element name="comment" type="Q3:contentType" maxOccurs="1"
minOccurs="1">
                    <sequence>
                        <element name="visualRepresentation" type="string"></element>
                        <element name="crtSockets" type="string"></element>
                        <element name="adaptationBehaviour" type="string"></element>
                        <element name="constraints" type="string"></element>
                        <element name="associatedDMRelations"
type="string"></element>
                    </sequence>
                </element>
            </sequence>
        </element>
    </sequence>
</complexType>

```

```
</complexType>  
</schema>
```

Appendix III. CAM Example

In this section an example of a CAM definition is given. The example is a simple solar system CAM as used in the user guide.

```
<model>
  <header>
    <modeluuid>8328858e-3161-419c-a96a-99863eb7a1c2</modeluuid>
    <modeltype>cam</modeltype>
    <authoruuid>null</authoruuid>
    <authorisation>readwrite</authorisation>
    <creationtime>1253140881961</creationtime>
    <updatetime>1254175085904</updatetime>
    <title>Solar System</title>
    <description>test</description>
  </header>
  <body>
    <cam>
      <camInternal>
        <crt type="crt">
          <camCrtUuid>B99C78C6-35E7-FEDE-9D0F-C506EB575985</camCrtUuid>
          <uuid>aaaaaaaa-12b5-4720-92e5-736cac59985b</uuid>
          <shape>diamond</shape>
          <colour>#00FF00</colour>
          <position>
            <x>672.5</x>
            <y>109.5</y>
          </position>
        </camInternal>
        <camSocket>
          <uuid>0233C4AF-5414-2BEA-E41A-C506EB574C46</uuid>
          <socketid>cf5de7f5-12b5-4720-92e5-qqqqqqqqqq</socketid>
          <colour>26163</colour>
          <shape>SocketShape</shape>
          <position>
            <x>475.675</x>
            <y>109.5</y>
          </position>
        </camSocket>
        <entity>
          <dmId>989cc040-45f4-11de-8a39-0800200c9a66</dmId>
        </entity>
        <entity>
          <dmId>af56f6f0-4600-11de-8a39-0800200c9a66</dmId>
        </entity>
      </cam>
    </body>
  </model>
```

```

    </entity>
    <entity>
        <dmId>44c31200-4601-11de-8a39-0800200c9a66</dmId>
    </entity>
    <entity>
        <dmId>a5d15f80-4600-11de-8a39-0800200c9a66</dmId>
    </entity>
</camSocket>
</crt>
<crt type="crt">
    <camCrtUuid>A9F70CA4-1010-87AE-D2BE-C5071FEDD659</camCrtUuid>
    <uuid>cf5de7f5-12b5-4720-92e5-736cac59985b</uuid>
    <shape>Diamond</shape>
    <colour>#0000FF</colour>
    <position>
        <x>280</x>
        <y>169</y>
    </position>
    <camSocket>
        <uuid>5D85FA82-B2F5-7C80-A031-C5071FEE9809</uuid>
        <socketid>cf5de7f5-12b5-4720-92e5-zzzzzzzzzzzz</socketid>
        <colour>26163</colour>
        <shape>SocketShape</shape>
        <position>
            <x>105.325</x>
            <y>227.5</y>
        </position>
        <entity>
            <dmId>44c31200-4601-11de-8a39-0800200c9a66</dmId>
        </entity>
    </camSocket>
    <camSocket>
        <uuid>F16898C3-C4F8-BD81-5E66-C5071FEE6252</uuid>
        <socketid>cf5de7f5-12b5-4720-92e5-pppppppppppp</socketid>
        <colour>26163</colour>
        <shape>SocketShape</shape>
        <position>
            <x>104.325</x>
            <y>411.5</y>
        </position>
        <entity>
            <dmId>e2506f00-4600-11de-8a39-0800200c9a66</dmId>
        </entity>

```



```

    </camSocket>
  </crt>
  <crt type="crt">
    <camCrtUuid>90B2D9EA-1344-3CA0-3F33-C5072C702806</camCrtUuid>
    <uuid>cf5de7f5-12b5-4720-92e5-736cac59985b</uuid>
    <shape>Diamond</shape>
    <colour>#0000FF</colour>
    <position>
      <x>263</x>
      <y>343</y>
    </position>
    <camSocket>
      <uuid>3D06AE6B-183C-BA4C-8DEE-C5072C71A6E5</uuid>
      <socketid>cf5de7f5-12b5-4720-92e5-zzzzzzzzzzzz</socketid>
      <colour>26163</colour>
      <shape>SocketShape</shape>
      <position>
        <x>251.325</x>
        <y>224.5</y>
      </position>
      <entity>
        <dmId>a5d15f80-4600-11de-8a39-0800200c9a66</dmId>
      </entity>
    </camSocket>
    <camSocket>
      <uuid>6EEB1AD4-7BDB-9F8B-F35F-C5072C71C856</uuid>
      <socketid>cf5de7f5-12b5-4720-92e5-pppppppppppp</socketid>
      <colour>26163</colour>
      <shape>SocketShape</shape>
      <position>
        <x>250.625</x>
        <y>415</y>
      </position>
      <entity>
        <dmId>af56f6f0-4600-11de-8a39-0800200c9a66</dmId>
      </entity>
    </camSocket>
  </crt>
  <crt type="crt">
    <camCrtUuid>BA474EAE-5FBF-E193-10C2-C50770E82365</camCrtUuid>
    <uuid>cf5de7f5-12b5-4720-92e5-736cac59985b</uuid>
    <shape>Diamond</shape>
    <colour>#0000FF</colour>

```

```

<position>
  <x>389</x>
  <y>297</y>
</position>
<camSocket>
  <uuid>28283A29-2768-B4C5-2FAC-C50770E9447F</uuid>
  <socketid>cf5de7f5-12b5-4720-92e5-zzzzzzzzzzzz</socketid>
  <colour>0</colour>
  <shape>SocketShape</shape>
  <position>
    <x>417.125</x>
    <y>329.5</y>
  </position>
  <entity>
    <dmId>44c31200-4601-11de-8a39-0800200c9a66</dmId>
  </entity>
</camSocket>
<camSocket>
  <uuid>4D9D26F2-8880-2700-FE1B-C50770E9AC99</uuid>
  <socketid>cf5de7f5-12b5-4720-92e5-pppppppppppp</socketid>
  <colour>16776960</colour>
  <shape>SocketShape</shape>
  <position>
    <x>420.325</x>
    <y>451.5</y>
  </position>
  <entity>
    <dmId>cbb6e940-4600-11de-8a39-0800200c9a66</dmId>
  </entity>
</camSocket>
</crt>
<crt type="crt">
  <camCrtUuid>626405B2-9DA5-21FF-954A-C50799DBE7F4</camCrtUuid>
  <uuid>cf5de7f5-12b5-4720-92e5-736cac59985b</uuid>
  <shape>Diamond</shape>
  <colour>#0000FF</colour>
  <position>
    <x>539</x>
    <y>242</y>
  </position>
  <camSocket>
    <uuid>EE799E90-4EC4-65EC-533F-C50799DB699A</uuid>
    <socketid>cf5de7f5-12b5-4720-92e5-zzzzzzzzzzzz</socketid>

```

```

    <colour>10027008</colour>
    <shape>SocketShape</shape>
    <position>
      <x>567.325</x>
      <y>199.5</y>
    </position>
    <entity>
      <dmId>44c31200-4601-11de-8a39-0800200c9a66</dmId>
    </entity>
  </camSocket>
  <camSocket>
    <uuid>2F0368AC-3992-0621-D3AB-C50799DB032B</uuid>
    <socketid>cf5de7f5-12b5-4720-92e5-pppppppppppp</socketid>
    <colour>3407616</colour>
    <shape>SocketShape</shape>
    <position>
      <x>570.275</x>
      <y>393.5</y>
    </position>
    <entity>
      <dmId>d45f8e30-4600-11de-8a39-0800200c9a66</dmId>
    </entity>
  </camSocket>
</crt>
<crtModel>
  <model>
    <header>
      <title>start</title>
      <description>Start CRT</description>
      <creationtime>1253890206022</creationtime>
      <updatetime/>
      <modeluuid>aaaaaaaa-12b5-4720-92e5-736cac59985b</modeluuid>
      <authoruuid/>
      <authorisation>readwrite</authorisation>
      <modeltype>crt</modeltype>
    </header>
    <body>
      <crt>
        <comment/>
        <crtdialect>CRT</crtdialect>
        <visualrepresentation>
          <colour>0x00FFCC</colour>
          <shape>diamond</shape>

```

```

    </visualrepresentation>
    <adaptationbehaviour>
      <galcode/>
      <usermodel/>
    </adaptationbehaviour>
    <constraints>
      <loopsallowed>false</loopsallowed>
    </constraints>
    <crtsockets>
      <socket type="start_link">
        <uuid>cf5de7f5-12b5-4720-92e5-qqqqqqqqqq</uuid>
        <name>start_socket</name>
        <mincardinality>1</mincardinality>
        <maxcardinality>*</maxcardinality>
      </socket>
    </crtsockets>
    <associateddmrelations/>
  </crt>
</body>
</model>
<model>
  <header>
    <title>prereq-test-mh</title>
    <description>prereq-test-mh</description>
    <creationtime>1253890007331</creationtime>
    <updatetime/>
    <modeluuid>cf5de7f5-12b5-4720-92e5-736cac59985b</modeluuid>
    <authoruuid/>
    <authorisation>readwrite</authorisation>
    <modeltype>crt</modeltype>
  </header>
  <body>
    <crt>
      <comment/>
      <crtdialect>CRT</crtdialect>
      <visualrepresentation>
        <colour>#000000</colour>
        <shape>diamond</shape>
      </visualrepresentation>
      <adaptationbehaviour>
        <galcode/>
        <usermodel/>
      </adaptationbehaviour>
    </crt>
  </body>
</model>

```

```

    <constraints>
      <loopsallowed>false</loopsallowed>
    </constraints>
    <crtsockets>
      <socket type="source">
        <uuid>cf5de7f5-12b5-4720-92e5-zzzzzzzzzzz</uuid>
        <name>beginner</name>
        <mincardinality>1</mincardinality>
        <maxcardinality>*</maxcardinality>
      </socket>
      <socket type="target">
        <uuid>cf5de7f5-12b5-4720-92e5-pppppppppppp</uuid>
        <name>advanced</name>
        <mincardinality>1</mincardinality>
        <maxcardinality>*</maxcardinality>
      </socket>
    </crtsockets>
    <associateddmrelations/>
  </crt>
</body>
</model>
</crtModel>
<domainModel>
  <model>
    <header>
      <modeluuid>660e8400-e29b-41d4-a716-446655440000</modeluuid>
      <modeltype>dm</modeltype>
      <authoruuid>null</authoruuid>
      <authorisation>readwrite</authorisation>
      <creationtime>1242904243000</creationtime>
      <updatetime>1242904243000</updatetime>
      <title>sun-example-dm</title>
      <description>sun-example-dm</description>
    </header>
    <body>
      <dm>
        <vdex>
          <term>
            <termIdentifier>989cc040-45f4-11de-8a39-
0800200c9a66</termIdentifier>
            <caption>
              <langstring language="it">Solar System</langstring>
            </caption>
          </term>
        </vdex>
      </dm>
    </body>
  </model>
</domainModel>

```

```

        <description>
            <langstring language="en">blabla</langstring>
        </description>
        <mediaDescriptor>

<mediaLocator>http://host/solarSystem.jpg</mediaLocator>
        <interpretationNote>
            <langstring language="en">some note on the
media</langstring>
        </interpretationNote>
        </mediaDescriptor>
        <metadata/>
    </term>
    <term>
        <termIdentifier>a5d15f80-4600-11de-8a39-
0800200c9a66</termIdentifier>
        <caption>
            <langstring language="it">Star</langstring>
        </caption>
        <description>
            <langstring language="en">blabla</langstring>
        </description>
        <mediaDescriptor>
            <mediaLocator>http://host/star.pdf</mediaLocator>
            <interpretationNote>
                <langstring language="en">some note on the
media</langstring>
            </interpretationNote>
        </mediaDescriptor>
        <mediaDescriptor>

<mediaLocator>http://en.wikipedia.org/wiki/star</mediaLocator>
        <interpretationNote>
            <langstring language="en">some note on the
media</langstring>
        </interpretationNote>
        </mediaDescriptor>
        <metadata/>
        <index>0</index>
    </term>
    <term>
        <termIdentifier>af56f6f0-4600-11de-8a39-
0800200c9a66</termIdentifier>

```

```

        <caption>
            <langstring language="it">Sun</langstring>
        </caption>
        <description>
            <langstring language="en">blabla</langstring>
        </description>
        <mediaDescriptor>
            <mediaLocator>http://host/sun.x3d</mediaLocator>
            <interpretationNote>
                <langstring language="en">some note on the
media</langstring>
            </interpretationNote>
        </mediaDescriptor>
        <mediaDescriptor>

<mediaLocator>http://en.wikipedia.org/wiki/sun</mediaLocator>
        <interpretationNote>
            <langstring language="en">some note on the
media</langstring>
        </interpretationNote>
    </mediaDescriptor>
    <metadata/>
    <index>1</index>
</term>
<term>
    <termIdentifier>cbb6e940-4600-11de-8a39-
0800200c9a66</termIdentifier>
    <caption>
        <langstring language="it">Mars</langstring>
    </caption>
    <description>
        <langstring language="en">blabla</langstring>
    </description>
    <mediaDescriptor>

<mediaLocator>http://en.wikipedia.org/wiki/mars</mediaLocator>
        <interpretationNote>
            <langstring language="en">some note on the
media</langstring>
        </interpretationNote>
    </mediaDescriptor>
    <dmId>cbb6e940-4600-11de-8a39-0800200c9a66</dmId>
    <entity>

```

```

        <dmId>cbb6e940-4600-11de-8a39-0800200c9a66</dmId>
    </entity>
</term>
<term>
    <termIdentifier>44c31200-4601-11de-8a39-
0800200c9a66</termIdentifier>
    <caption>
        <langstring language="it">Planet</langstring>
    </caption>
    <description>
        <langstring language="en">blabla</langstring>
    </description>
    <mediaDescriptor>

<mediaLocator>http://en.wikipedia.org/wiki/Planet</mediaLocator>
    <interpretationNote>
        <langstring language="en">some note on the
media</langstring>
    </interpretationNote>
</mediaDescriptor>
<metadata/>
<dmId>44c31200-4601-11de-8a39-0800200c9a66</dmId>
<entity>
    <dmId>44c31200-4601-11de-8a39-0800200c9a66</dmId>
</entity>
</term>
<term>
    <termIdentifier>d45f8e30-4600-11de-8a39-
0800200c9a66</termIdentifier>
    <caption>
        <langstring language="it">Earth</langstring>
    </caption>
    <description>
        <langstring language="en">blabla</langstring>
    </description>
    <mediaDescriptor>

<mediaLocator>http://en.wikipedia.org/wiki/earth</mediaLocator>
    <interpretationNote>
        <langstring language="en">some note on the
media</langstring>
    </interpretationNote>
</mediaDescriptor>

```



```

        <metadata/>
        <dmId>d45f8e30-4600-11de-8a39-0800200c9a66</dmId>
        <entity>
            <dmId>d45f8e30-4600-11de-8a39-0800200c9a66</dmId>
        </entity>
    </term>
    <term>
        <termIdentifier>e2506f00-4600-11de-8a39-
0800200c9a66</termIdentifier>
        <caption>
            <langstring language="it">Jupiter</langstring>
        </caption>
        <description>
            <langstring language="en">blabla</langstring>
        </description>
        <mediaDescriptor>

<mediaLocator>http://en.wikipedia.org/wiki/jupiter</mediaLocator>
        <interpretationNote>
            <langstring language="en">some note on the
media</langstring>
        </interpretationNote>
    </mediaDescriptor>
    <metadata/>
    <dmId>e2506f00-4600-11de-8a39-0800200c9a66</dmId>
    <entity>
        <dmId>e2506f00-4600-11de-8a39-0800200c9a66</dmId>
    </entity>
</term>
<relationship>
    <sourceTerm>af56f6f0-4600-11de-8a39-
0800200c9a66</sourceTerm>
    <targetTerm>a5d15f80-4600-11de-8a39-
0800200c9a66</targetTerm>
    <relationshipType
source="http://www.grapple.org/relations.xml">is_a</relationshipType>
    <metadata/>
</relationship>
<relationship>
    <sourceTerm>e2506f00-4600-11de-8a39-
0800200c9a66</sourceTerm>
    <targetTerm>44c31200-4601-11de-8a39-
0800200c9a66</targetTerm>

```

```

        <relationshipType
source="http://www.grapple.org/relations.xml">is_a</relationshipType>
        <metadata/>
    </relationship>
    <relationship>
        <sourceTerm>d45f8e30-4600-11de-8a39-
0800200c9a66</sourceTerm>
        <targetTerm>44c31200-4601-11de-8a39-
0800200c9a66</targetTerm>
        <relationshipType
source="http://www.grapple.org/relations.xml">is_a</relationshipType>
        <metadata/>
    </relationship>
    <relationship>
        <sourceTerm>cbb6e940-4600-11de-8a39-
0800200c9a66</sourceTerm>
        <targetTerm>44c31200-4601-11de-8a39-
0800200c9a66</targetTerm>
        <relationshipType
source="http://www.grapple.org">is_a</relationshipType>
    </relationship>
    <relationship>
        <sourceTerm>989cc040-45f4-11de-8a39-
0800200c9a66</sourceTerm>
        <targetTerm>af56f6f0-4600-11de-8a39-
0800200c9a66</targetTerm>
        <relationshipType
source="http://www.grapple.org/relations.xml">main_star</relationshipType>
        <metadata/>
    </relationship>
    <relationship>
        <sourceTerm>af56f6f0-4600-11de-8a39-
0800200c9a66</sourceTerm>
        <targetTerm>a5d15f80-4600-11de-8a39-
0800200c9a66</targetTerm>
        <relationshipType
source="http://www.grapple.org/relations.xml">is_a</relationshipType>
        <metadata/>
    </relationship>
    <relationship>
        <sourceTerm>989cc040-45f4-11de-8a39-
0800200c9a66</sourceTerm>

```

```

        <targetTerm>af56f6f0-4600-11de-8a39-
0800200c9a66</targetTerm>
        <relationshipType
source="http://www.grapple.org/relations.xml">is_composed_of</relationship
Type>
        <metadata/>
        </relationship>
        <relationship>
        <sourceTerm>989cc040-45f4-11de-8a39-
0800200c9a66</sourceTerm>
        <targetTerm>cbb6e940-4600-11de-8a39-
0800200c9a66</targetTerm>
        <relationshipType
source="http://www.grapple.org/relations.xml">is_composed_of</relationship
Type>
        <metadata/>
        </relationship>
        <relationship>
        <sourceTerm>989cc040-45f4-11de-8a39-
0800200c9a66</sourceTerm>
        <targetTerm>44c31200-4601-11de-8a39-
0800200c9a66</targetTerm>
        <relationshipType
source="http://www.grapple.org/relations.xml">is_composed_of</relationship
Type>
        <metadata/>
        </relationship>
        <relationship>
        <sourceTerm>989cc040-45f4-11de-8a39-
0800200c9a66</sourceTerm>
        <targetTerm>d45f8e30-4600-11de-8a39-
0800200c9a66</targetTerm>
        <relationshipType
source="http://www.grapple.org/relations.xml">is_composed_of</relationship
Type>
        <metadata/>
        </relationship>
        <relationship>
        <sourceTerm>989cc040-45f4-11de-8a39-
0800200c9a66</sourceTerm>
        <targetTerm>e2506f00-4600-11de-8a39-
0800200c9a66</targetTerm>

```

```

        <relationshipType
source="http://www.grapple.org/relations.xml">is_composed_of</relationship
Type>
        <metadata/>
    </relationship>
</vdex>
</dm>
</body>
</model>
</domainModel>
</camInternal>
</cam>
</body>
</model>

```

Appendix IV. XML Schema of CAM External Language

This section contains the XML schema specification of the external CAM XML format. CAMs which are exported by the CAM are valid against this schema specification. Note that the external file format is a simplification of the internal file format and that the GRAPPLE components are designed to ignore the unneeded information in the internal format.

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.example.org/CAM-external-oct2009"
  xmlns:tns="http://www.example.org/CAM-external-oct2009"
  elementFormDefault="qualified"
  xmlns:Q1="http://www.imsglobal.org/xsd/imsvdex_vlp0"
  xmlns:Q2="http://grapple-project.org/GAT/" xmlns:Q3="http://www.grapple-
project.org">

  <import schemaLocation="dm-vdex-extensions.xsd"
    namespace="http://www.grapple-project.org"></import>

  <import schemaLocation="CRTOct2009.xsd" namespace="http://grapple-
project.org/GAT/"></import>

  <import schemaLocation="vdex.xsd"
    namespace="http://www.imsglobal.org/xsd/imsvdex_vlp0"></import>

  <element name="model" type="tns:modelType"></element>

  <complexType name="modelType">
    <sequence maxOccurs="1" minOccurs="1">
      <element name="header" type="tns:headerType" maxOccurs="1"
minOccurs="1"></element>
      <element name="body" type="tns:camBodyType" maxOccurs="1"
minOccurs="1"></element>
    </sequence>
  </complexType>

  <complexType name="headerType">
    <sequence>
      <element name="modeluuid" type="tns:UUIDtype" maxOccurs="1"
minOccurs="1">
        </element>
    </sequence>
  </complexType>
```

```

        <element name="modeltype" type="string" maxOccurs="1"
            minOccurs="1">
        </element>
        <element name="authoruuid" type="tns:UUIDtype" maxOccurs="1"
minOccurs="1"></element>
        <element name="authorisation" type="tns:authorisationType"
maxOccurs="1" minOccurs="1"></element>
        <element name="creationtime" type="dateTime" maxOccurs="1"
minOccurs="1"></element>
        <element name="updatetime" type="dateTime" maxOccurs="1"
minOccurs="1"></element>
        <element name="title" type="string" maxOccurs="1"
minOccurs="1"></element>
        <element name="description" type="string" maxOccurs="1"
minOccurs="1"></element>
    </sequence>
</complexType>

<complexType name="camBodyType">
    <sequence>
        <element name="cam" type="tns:camType" maxOccurs="1"
minOccurs="1"></element>
    </sequence>
</complexType>

<simpleType name="authorisationType">
    <restriction base="string">
        <enumeration value="read"></enumeration>
        <enumeration value="write"></enumeration>
        <enumeration value="readwrite"></enumeration>
    </restriction>
</simpleType>

<simpleType name="UUIDtype">
    <restriction base="string">
        <pattern
            value="[a-z0-9]{8}\-[a-z0-9]{4}\-[a-z0-9]{4}\-[a-z0-
9]{4}\-[a-z0-9]{12}">
        </pattern>
    </restriction>
</simpleType>

```

```

<complexType name="camType">
  <sequence>
    <element name="camInternal"
type="tns:camInternalType"></element>
  </sequence>
</complexType>

<complexType name="camInternalType">
  <sequence>
    <element name="domainModel" type="tns:dmModelType"
maxOccurs="1"
      minOccurs="1">
    </element>
    <element name="crtModel" type="tns:crtModelType"
maxOccurs="1"
      minOccurs="1">
    </element>
    <element name="crt" type="string" maxOccurs="unbounded"
minOccurs="0"></element>
  </sequence>
</complexType>

<complexType name="positionType">
  <sequence>
    <element name="x" type="int"></element>
    <element name="y" type="int"></element>
  </sequence>
</complexType>

<complexType name="camSocketType">
  <sequence>
    <element name="caption" type="string" maxOccurs="1"
      minOccurs="0">
    </element>
    <element name="uuid" type="tns:UUIDtype" maxOccurs="1"
      minOccurs="1">
    </element>
    <element name="socketId" type="tns:UUIDtype" maxOccurs="1"
      minOccurs="1">
    </element>
  </sequence>

```

```

        <element name="entity " type="tns:entityType"
maxOccurs="unbounded" minOccurs="0"></element>
    </sequence>
</complexType>

<complexType name="entityType">
    <sequence>
        <element name="dmID" type="tns:UUIDtype" maxOccurs="1"
minOccurs="0">
            </element>
        <element name="label" type="string" maxOccurs="unbounded"
minOccurs="0">
            </element>
        <element name="relationshipType " type="string" maxOccurs="1"
minOccurs="0"></element>
    </sequence>
</complexType>

<complexType name="dmModelType">
    <sequence>
        <element name="model" type="tns:dmModelType2"
maxOccurs="unbounded" minOccurs="0"></element>
    </sequence>
</complexType>

<complexType name="dmModelType2">
    <sequence>
        <element name="header" type="tns:headerType"></element>
        <element name="body" type="tns:dmType"></element>
    </sequence>
</complexType>

<complexType name="dmType">
    <sequence>
        <element name="vdex" type="Q1:vdexType" maxOccurs="1"
minOccurs="1"></element>
    </sequence>
</complexType>

<complexType name="crtModelType">
    <sequence>
        <element name="model" type="tns:crtModelType2"
maxOccurs="unbounded" minOccurs="0"></element>

```



```

        </sequence>
    </complexType>

    <complexType name="crtModelType2">
        <sequence>
            <element name="header" type="tns:headerType" maxOccurs="1"
                minOccurs="1">
                </element>
            <element name="body" type="tns:crtType" maxOccurs="1"
minOccurs="1"><complexType></complexType></element>
        </sequence>
    </complexType>

    <complexType name="crtType">
        <sequence>
            <element name="crtDialect" type="Q2:crtDialectType"
                maxOccurs="1" minOccurs="1">
                </element>
            <element name="comment" type="Q3:contentType" maxOccurs="1"
                minOccurs="1">
                </element>
            <element name="crtSockets" type="string"></element>
            <element name="adaptationbehaviour" type="string"></element>
            <element name="constraints" type="string"></element>
            <element name="associateddmrelations"
type="string"></element>
        </sequence>
    </complexType>
</schema>

```

Appendix V. Information for Integrating GAT

Component Into the GAT Shell

For the integration with the shell the following issues need to be observed:

1. If you run your module stand-alone, copy the assets to your source path.
2. Please make your component implement ClosableVBox
3. The events (see 3) need to be handled by the individual components

In the source path ../shared/main/flex there are the following folders:

V.I. *assets*

If you run your module stand-alone, copy the assets to your source path.

File	Component	Use
Font	all	Used by CSS in main
hitskin.png	all	Used by main, floating window library
selectionCursor.gif	all	Selection cursor
settings.xml	all	used for identifying the WSDL for web services
shapes.xml	CRT CAM	indicates which shapes for CRTs and sockets are available

V.II. *net*

net contains the library for floating windows, as used by main. Please make your components implement ClosableVBox as in the cam component (cam.mxml on top):

```
<goozo:ClosableVBox xmlns:mx="http://www.adobe.com/2006/mxml"
  xmlns:shared="org.grapple_project.authoring.ui.*"
  xmlns:ravis="org.un.cava.birdeye.ravis.graphLayout.visual.*"
  xmlns:grapplevisualgraph="org.grapple_project.authoring.*"
  xmlns:goozo="net.goozo.mx.dockable.*" ...
```

To determine what happens just before the window is close, override *closeTab*, see

```
override public function closeTab():Boolean in cam.mxml
```

V.III. org.grapple_authoring_tool.authoring.events

The events are the main method of communication between the shell and the individual components. The following events have to be handled by the components, see cam.mxml for examples.

V.III.I. onLoadModel

See

```
protected function load( event :LoadModelEvent):void
```

in cam.mxml. The model to be loaded is given. It's guaranteed to be of the correct type (the DM component will only be asked to open a DM)

V.III.II. onSaveModel

The author clicked save, or selected a file for save as. See protected function

```
save( event :SaveAsModelEvent=null) : void
```

in cam.mxml. The method uses

```
toolbar.saveModel
```

to save the model (this function deals with the WS calls).

V.III.III. onNewModel

This event is fired, when the authors selects File → New see

```
protected function newModel( event:NewModelEvent) : void
```

in cam.mxml. This function will only be called on a newly opened window of a DM/CRT/CAM component, so you can assume that this instance was not already holding any models.

V.III.IV. onMenuItemClick

The author clicked an item on the menu bar. This event fires for all buttons on the menu bar, with a switch the relevant actions can be coupled to buttons. See protected

```
function menuItemClicked( event :MenuItemClickEvent) : void
```

in cam.mxml

V.III.V. onUpdateCurrentModel

With this method the toolbar notifies a dm/crt/CAM component window that it wants to know the model currently opened in the component in its current state.

`toolbar.currentModel` should be set. See

```
protected function updateModel( event:UpdateCurrentModelEvent) : void
```

in `cam.mxml`.

V.III.VI. onSetToolbar

This event gives the component a handle to the toolbar. It will be called at start-up and at every change of the active window. See

```
public function setToolBar(event:SetToolbarEvent ):void
```

in `cam.mxml` . The availability status of buttons in the toolbar should be set, see

```
public function updateToolbar():void
```

in `cam.mxml`

V.IV. *org.grapple_authoring_tool.authoring.ui*

This package contains the toolbar and widgets used by it. Some of these can be useful independently of the toolbar. For example `OpenFileDialog` is used in `cam.mxml` to allow the author to select which crt he wants to insert, see

```
protected function addCrt(evt:ContextMenuEvent):void
```

in `cam.mxml`.

The most important class for the GAT components is `Toolbar.mxml`. It is uses throughout `cam.mxml`, see the functions

```
updateSelection, setToolBar, getToolbar, updateToolbar, onAdded, load, save,  
newModel, menuItemClicked, updateModel, addCrt, confirmDeleteConcepts.
```

The toolbar has the following public variables and functions:

Name	Use
<code>static const MENU_FILE</code>	Name (of type <code>String</code>) of menu item as passed in <code>onMenuItemClick</code>
<code>static const MENU_FILE_NEW</code>	Name of menu item
<code>static const MENU_FILE_OPEN</code>	Name of menu item

Name	Use
<code>static const MENU_FILE_SAVE</code>	Name of menu item
<code>static const MENU_FILE_SAVE_AS</code>	Name of menu item
<code>static const MENU_FILE_PROPERTIES</code>	Name of menu item
<code>static const MENU_FILE_DELETE</code>	Name of menu item
<code>static const MENU_FILE_DEPLOY</code>	Name of menu item
<code>static const MENU_EDIT</code>	Name of menu item
<code>static const MENU_EDIT_SELECT</code>	Name of menu item
<code>static const MENU_EDIT_SELECT_ALL</code>	Name of menu item
<code>static const MENU_EDIT_CUT</code>	Name of menu item
<code>static const MENU_EDIT_COPY</code>	Name of menu item
<code>static const MENU_EDIT_PASTE</code>	Name of menu item
<code>static const MENU_EDIT_DELETE</code>	Name of menu item
<code>static const MENU_HELP_HELP</code>	Name of menu item
<code>static const MENU_VR</code>	Name of menu item
<code>static const MENU_SIMULATION</code>	Name of menu item
<code>static const MENU_HELP</code>	Name of menu item
<code>static const MENU_HELP_ABOUT</code>	Name of menu item
<code>var enabledMENU_FILE</code>	Var that determines enabledness (Boolean) of button in menu. Changes will only be active after a call to <code>setButtonEnabled</code> . Please call this after all variables have been set as repeated calls may cause a visible refresh delay.
<code>var enabledMENU_FILE_NEW</code>	determines enabledness of button in menu
<code>var enabledMENU_FILE_OPEN</code>	determines enabledness of button in menu
<code>var enabledMENU_FILE_SAVE</code>	determines enabledness of button in menu
<code>var enabledMENU_FILE_SAVE_AS</code>	determines enabledness of button in menu
<code>var enabledMENU_FILE_PROPERTIES</code>	determines enabledness of button in menu

Name	Use
<code>var enabledMENU_FILE_DELETE</code>	determines enabledness of button in menu
<code>var enabledMENU_FILE_DEPLOY</code>	determines enabledness of button in menu
<code>var enabledMENU_EDIT</code>	determines enabledness of button in menu
<code>var enabledMENU_EDIT_SELECT</code>	determines enabledness of button in menu
<code>var enabledMENU_EDIT_SELECT_ALL</code>	determines enabledness of button in menu
<code>var enabledMENU_EDIT_CUT</code>	determines enabledness of button in menu
<code>var enabledMENU_EDIT_COPY</code>	determines enabledness of button in menu
<code>var enabledMENU_EDIT_PASTE</code>	determines enabledness of button in menu
<code>var enabledMENU_EDIT_DELETE</code>	determines enabledness of button in menu
<code>var enabledMENU_VR</code>	determines enabledness of button in menu
<code>var enabledMENU_SIMULATION</code>	determines enabledness of button in menu
<code>var enabledMENU_HELP</code>	determines enabledness of button in menu
<code>var enabledMENU_HELP_HELP</code>	determines enabledness of button in menu
<code>var enabledMENU_HELP_ABOUT</code>	determines enabledness of button in menu
<code>var currentModel:XML=null;</code>	The current model of the current active window, components should set this variable when they receive an <code>updateModelEvent</code>
<code>var container:Container;</code>	Variable for use with popups. See <code>protected function close()</code> in <code>basicPopUp.mxml</code>
<code>function reset():void</code>	Rest the exaltedness of all buttons on the toolbar
<code>function setButtonEnabled():void</code>	Apply changes to the exaltedness of buttons Please call this after all variables have been set as repeated calls may cause a visible refresh delay.

Name	Use
<code>function saveModel(model:XML, message:String="")</code>	Save the model provided in <i>model</i> using the web services and display the message <i>message</i> upon success.
<code>function addPopUp(popup:BasicPopup):void</code>	Add the popup of type BasicPopup on top of the view stack, temporarily disabling all underlying windows.

V.V. org.grapple_authoring_tool.authoring.utils

This package contains the following classes:

V.V.I. DrawingUtils

This class cannot be instantiated and contains the following public static functions that can be used throughout the GAT (importing of the class is still needed):

```
public static function dottedRect(graphics:flash.display.Graphics,  
    fromX:Number, fromY:Number, toX:Number, toY:Number, gap:int=5,  
    dot:int=5):void.
```

Draws a rectangle with dotted or dashed line.

```
public static function dottedLine(graphics:flash.display.Graphics,  
    fromX:Number, fromY:Number, toX:Number, toY:Number, gap:int=10,  
    dash:int=5):void.
```

Draws a dotted or dashed line, also used by `dottedRect`

```
public static function stringToColour(colourString:String):uint.
```

Convert a string to uint. The function converts html style formats (for example #000000) to uint style formats (for example 0x000000)

V.V.II. KeyUtils

Used by the menu for the keyboard shortcuts

V.VI. org.grapple_authoring_tool.authoring.CopyPaste

Class for the copy-paste functionality. The variable `model` and `items` are available globally.

V.VI.I. Concepts

For copy-pasting concepts, the model variable should contain a list of model(s) starting with the `<model>` tag. Often this will be a list of length 1 but items could come from different DM's. The items array should contain a string array with the *uuids* of the items.

V.VI.II. CRTs

For CRTs the items array is not used, the model XMLList is used in the same way as for the Concepts.

V.VII. org.grapple_authoring_tool.authoring.GrappleVgraph

An extension of the VisualGraph in the RaVis library that allows disabling traverse on double click and more importantly, selection of nodes and edges. Notable properties and functions are:

```
traverseOnDoubleClick,      selectedNodes      (see      updateToolbar),
selectedEdges, selectionMode and selectAll (see menuItemClicked).
```

V.VIII. org.grapple_authoring_tool.authoring.crtshapes

These are the actual shapes a CRT can take, to add additional shapes, implement a shape along these lines and add to shapes.xml.

V.IX. org.grapple_authoring_tool.authoring.socketshapes

These are the actual shapes a socket can take, to add additional shapes, implement a shape along these lines and add to shapes.xml

V.X. org.un

Include the ravis library

Appendix VI. GAT Qualitative Online Survey

In this appendix we represent the data gathered from the open questions, at the evaluation of the GAT. Section VI.I lists the questions asked and section VI.II shows the results gathered.

VI.I. Questions asked

The questionnaire consisted of the following three questions.

- UFQQ1: What did you like best about the system and authoring tool?
- UFQQ2: What did you like least about the system and authoring tool?
- UFQQ3: What should be improved and how?

VI.II. Full data collected

Below the data collected is presented. For each questions we show the responses given by the external respondents followed by the responses given by respondents involved in the GRAPPLE project. For both groups of respondents a table lists the answer category, the frequency and the statements that the respondents made.

UFQQ1: What did you like best about the system and authoring tool?

Respondents external to GRAPPLE

Answer Category	Frequency	Answer Statements
usability/interface	5	*The interface was nice and very easy to use. *graphical display *the user interface; the navigation bar *the layout of the user interface *I like the interface layout, very straightforward.
help	2	*the user help *the user manual, it's very helpful

Respondents external to GRAPPLE continued

Answer Category	Frequency	Answer Statements
usability	1	*usability is ok if you have been trained to use the system
versatility	1	*granularity and versatility of the system
possibility of reintegration of components for different purpose	1	*it seems that the various components can be reintegrated to do a slightly different thing (for example educational game)

Respondents involved in other parts of the GRAPPLE project

Answer Category	Frequency	Answer Statements
visual representation - DM	5	*the visually nice representation *the visual character - especially of the DM component, as it is very similar to creating an ordinary concept Model * nice graphical representation *I think DM editing is quite ok. *the visual character - especially of the DM component, as it is very similar to creating an ordinary concept Model
interface - tabs	2	*the interface use tabs *the tabs to jump between the individual tools
functionality	2	*it works :-) (sometimes) *Drag and drop functionality, copy and paste functionality
interface - graphical design	1	*the system is clearly arranged; the user interface and graphical design
use via web	1	*Create everything via the Web

Respondents involved in other parts of the GRAPPLE project continued

Answer Category	Frequency	Answer Statements
existence of examples	1	*existence of other (example) applications

UFQQ2: What did you like least about the system and authoring tool?

Respondents external to GRAPPLE

Answer Category	Frequency	Answer Statements
complexity	2	*pages can get very complex *Lots of fields to fill in!
possible problems with XHTML	1	*generating XHTML with the GRAPPLE specific tasks might be error prone. Either short-hand syntax or a graphical DIE with code hints would need to be used.
no testing capabilities	1	*for quality assurance the page needs to be tested somehow but I did not see any debugging capabilities for that. So I would need to see somehow what variables are used in the page, and have a possibility to manually enter those
association of resources in DM	1	*As far as I noticed, the concepts (ontology) directly Models to a page of information. I did not see the possibility to associate "fly-by" topics, that is, some topics that are done in various other pages and will contribute to the knowledge of an item - not specifically on "its" page.
no complete example	1	*If there is a complete example to use this system then it will rather be easy as compare to this tutorial.
menu - tabs	1	*Two level tab menu

Respondents external to GRAPPLE continued

Answer Category	Frequency	Answer Statements
CRT concept	1	*the concept of the CRT is difficult to understand, even with the user manual.
acronyms	1	*Too many acronyms, difficult to follow...such as CAM, CRT, GAL....

Respondents involved in other parts of the GRAPPLE project

Answer Category	Frequency	Answer Statements
CRT concept	3	*CRT editing is too challenging for ordinary users *CRT component is too complex, not usable by non-computer scientist, unclear how adaptation behaviour is defined, unclear what all the fields/attributes to be defined mean *No idea how to use UM part in CRT
multiplicity of tabs	3	*several tabs of the same tool are there, which is confusing *Tabs cannot be closed, just opened *There are three tabs. Only one seems active; What are the relationships between the tabs (at the two levels)
unexpected reactions	2	*CAM GUI reacts surprisingly sometimes *when trying to mark concepts with the mouse the window view is shifted instead; 'double-click to edit' does not always work
authoring approach as a whole	2	*I have my doubts about this authoring approach *I do not understand what it is for and how it works

Respondents involved in other parts of the GRAPPLE project continued

Answer Category	Frequency	Answer Statements
DM: add relationship	2	<p>*in DM component when right-clicking on a concept and selecting 'add relationship' the respective concept should automatically be the start node</p> <p>*cumbersome to relate nodes in DM</p>
CAM: filling sockets	2	<p>*it is very annoying that for filling the sockets with concepts the DM has to be chosen again and again for each concept</p> <p>*Cumbersome to attach CRT sockets to DM concepts</p>
complexity of tools	2	<p>*the use of the tools is difficult if you do not have any experiences; you have to read the help/the instruction very accurately to understand the tools and the usage of the tools; many buttons to fill in</p> <p>*no idea what every option or possibility in tools mean, how to use them, or what will be their effect in the application</p>
adaptation unclear	2	<p>*CRT component - unclear how adaptation behaviour is defined</p> <p>*no idea how resources can be made adaptive; People will not understand what GAL means. What about renaming that tab to "Adaptation rules" instead?</p>
no testing capability	2	<p>*No feedback when deploying application in CAM</p> <p>*unclear whether the course created would work - no possibility to check whether the envisaged course structure is realised</p>
DM: not full concept name visible	1	<p>*I do not understand what it is for and how it works</p>

Respondents involved in other parts of the GRAPPLE project continued

Answer Category	Frequency	Answer Statements
DM: no zooming	1	*no possibility to enlarge the DM window, no possibility to zoom in/out
CAM: problems with saving and retrieving	1	*after saving the CAM and opening it later again the sockets showed the same concepts a hundred times
no tutorial	1	*no training session
CAM does not use DM graph	1	*Would expect to connect CRTs to DM graph in CAM
selecting resources	1	*How can I choose which resources should be used in application?
fussiness of actions	1	*usability: have to do a lot of boring actions - instead of drag/drop, copy/past and so forth.
DM relations as CRTs unclear	1	*Not sure what will happen with for example domain "is_a" relationship in the final application

UFQQ3: What should be improved and how?

Respondents external to GRAPPLE

Answer Category	Frequency	Answer Statements
provide example	3	<p>*Example should be given in such a way that any body who uses this tutorial understands how we can implement in real learning environment.</p> <p>*it will be helpful if a demonstration adaptive material is given first and then the steps to create the material is given behind</p> <p>*That would be great if you could provide a couple of samples, so new users can easily import it, and have a play with it quickly</p>

Respondents external to GRAPPLE continued

Answer Category	Frequency	Answer Statements
allow user contributions/annotations	1	*Learning in an informal process most likely involves other peoples as well. a clear support for that (user contributed annotations to pages) would be important, I think.
create possibilities for collaboration	1	*Building up knowledge in a team might be an interesting topic as well. In that case, one would have a "user data model" and a "collective data model" - the user data model would have the individual knowledge status, the collective data model the model of the group. that could lead to interesting learning contexts, ranging from possibility to display "someone in your group already knows that, you try to ask him for more information", or to more sophisticated scenarios, where a competitive learning style could be used.
add tooltips	1	*Tooltips could be added to buttons, windows, tabs, and so on, to aid the user in navigation and use of the system
documentation	1	*Documentation

Respondents involved in other parts of the GRAPPLE project

Answer Category	Frequency	Answer Statements
improve help	3	<p>*help should be improved, provide information on the handling of the tools, tool-specific help</p> <p>*I wish for the help an avatar who helps me (like an avatar in Microsoft Word where I can enter my special question and I get one answer); → interactive help; there could be a simulation for the right use of the tools (there is a simulation button, but it does not work); or a training session with many different examples for the use</p> <p>*Give hints of what to do, when this is not obvious</p>
provide predefined examples	2	<p>*there should be definitely a basic set of predefined CRTs</p> <p>*Give simple example templates that can be used</p>
add testing capability	2	<p>*unclear whether the course created would work - no possibility to check whether the envisaged course structure is realised</p> <p>*Feedback for deploying an application</p>
coherency/consistency	2	<p>*More coherent presentation</p> <p>*consistency</p>
enhance graphical representation of CAM	2	<p>*why not use DM graph in CAM?; CAM in general would be nicer as a view over DM and set of CRTs anyhow</p> <p>* it should be possible to 'join' graphical representations of CRTs - have $A \rightarrow B \rightarrow C$ rather than $A \rightarrow B$ and $B \rightarrow C$</p>

Respondents involved in other parts of the GRAPPLE project continued

Answer Category	Frequency	Answer Statements
improve usability of operations	2	*usability *operations which are performed very often should be made easier or shortcut; the insertion of objects (CRTs like prerequisites) should be consistent in the editing panel (near the mouse button where the user expects it to be)
replace acronyms	1	*the DM, CRT, CAM acronyms are not intuitive and should be replaced by more suitable terms
add zoom function	1	*zoom function in DM and CAM
enhance adding of concepts in sockets of CRTs	1	*enhance filling of sockets with concepts in CAM
simplify CRT	1	*simplify CRT component
labelling DM concept directly in DM graph	1	*DM: possibility of naming concepts within visual present (under the dot)
enhance adding of relationships in DM	1	*DM: relation between concepts (right button click or just while editing concept) → not that concept is selected in the drop list when you want to specify relation with it
enhance functionality for opening files	1	*CAM (and others): I want to open CAM file, why is it not the default file type then
enhance deletion of socket	1	*CAM (and others): I want to open CAM file, why is it not the default file type then

Appendix VII. GAT Quntitative Online Survey

In this appendix we represent the data gathered from the closed questions, at the evaluation of the GAT. Section VI.I lists the questions asked and section VI.II shows the results gathered.

VII.I. Questions asked

The questionnaire consisted of the following twelve questions. Note that the first 10 questions are the standard SUS [19] questions. The possible answers to these closed questions range from 1 to 5 on a Likert scale, with 1 representing *Strongly disagree*, 5 representing *Strongly agree*.

- SUS1. I think that I would like to use this system frequently.
- SUS2. I found the system unnecessarily complex.
- SUS3. I thought the system was easy to use.
- SUS4. I think that I would need the support of a technical person to be able to use this system.
- SUS5. I found the various functions in this system were well integrated.
- SUS6. I thought there was too much inconsistency in this system.
- SUS7. I would imagine that most people would learn to use this system very quickly.
- SUS8. I found the system very cumbersome to use.
- SUS9. I felt very confident using the system.
- SUS10. I needed to learn a lot of things before I could get going with this system.
- UA1. I would like to use this system in the future.
- UA2. I would recommend this system to my colleagues.

VII.II. Full data collected

Below the data collected is presented. First we present the data gathered form external respondents, then the data gathered from the internal respondents. For both groups we start

with a table that gives an overview of the respondents' background, followed by a table that outlines their questionnaire answers and scores.

VII.II.I. Internal respondents

Respondents' background

Respondent nr	Gender	Age	Profession	Use learning environments 1=never 2= less than once a week 3= several times per week 4=daily	Experience learning environments 1= none 2 = several months 3= about 1 year 4= years	Authoring experience 1=none 2=some 3= advanced 4=expert
1	M	34	Software Engineer	1	1	4
2	-	-	-	-	-	-
3	F	30	Researcher	3	4	2
4	M	26	Student	1	1	1
5	M	28	Student	2	3	1
6	M	23	Researcher	2	3	2
7	M	27	PhD student	3	4	3
8	F	27	Software developer	1	4	1
9	-	-	-	-	-	-
10	F	26	PhD student	2	3	2
11	M	26	PhD student	3	4	2

Respondents' questionnaire results

Nr.	SUS 1	SUS 2	SUS 3	SUS 4	SUS 5	SUS 6	SUS 7	SUS 8	SUS 9	SUS 10	UA1	UA2	SUS-score	UA-score
1	4	3	3	2	3	1	3	3	3	1	5	3	62.5	4.00
2	2	3	3	3	3	3	2	3	3	3	3	3	47.5	3.00
3	2	4	2	2	3	2	4	2	1	1	2	2	47.5	2.00
4	4	3	3	4	4	3	3	2	3	5	3	3	52.5	3.00
5	4	2	4	4	4	2	2	4	4	4	3	4	57.5	3.50
6	3	2	3	3	4	2	4	3	3	5	2	2	50	2.00
7	3	2	4	3	4	4	4	2	3	3	3	4	60	3.50
8	3	3	3	1	3	1	3	3	2	2	5	5	62.5	5.00
9	4	2	3	4	4	1	1	3	4	3	5	4	62.5	4.50
10	3	5	2	5	5	1	3	4	1	5	3	3	30	3.00
11	3	2	2	1	3	1	2	3	2	4	4	3	55	3.50

VII.II.II. External respondents

Respondents' background

Respondent nr	Gender	Age	Profession	Use learning environments 1=never 2= less than once a week 3= several times per week 4=daily	Experience learning environments 1= none 2 = several months 3= about 1 year 4= years	Authoring experience 1=none 2=some 3= advanced 4=expert
12	M	46	Researcher	3	4	4
13	M	40	Teacher and Researcher	4	4	4
14	M	40	Researcher (incidentally teacher)	3	4	2
15	F	29	Postgraduate researcher	1	4	1
16	F	27	Student	1	1	1
17		-	-	-	-	-
18		-	-	-	-	-
19	M	31	Computer scientist	3	4	2
20	F	30	TEL (PostDoc)	1	4	4
11	M	27	Research professional with LMS/ALE background	1	2	2

Respondents' questionnaire results

Nr.	SUS 1	SUS 2	SUS 3	SUS 4	SUS 5	SUS 6	SUS 7	SUS 8	SUS 9	SUS 10	UA1	UA2	SUS-score	UA-score
12	3	4	2	2	3	4	2	4	2	4	3	3	35	3.00
13	2	3	2	1	4	1	2	3	4	5	4	2	52.5	3.00
14	1	3	1	5	1	3	2	4	1	4	3	1	32.5	2.00
15	2	4	2	5	3	2	2	4	2	3	3	3	32.5	3.00
16	4	2	2	3	4	2	2	3	2	4	4	4	50	4.00
17	1	5	3	1	3	4	2	3	3	2	1	2	42.5	1.50
18	1	3	4	3	2	2	1	3	4	2	2	3	47.5	2.50
19	2	2	2	4	3	4	2	4	3	2	5	2	40	3.50
20	4	3	3	1	3	2	4	4	3	1	4	4	65	4.00
21	2	3	2	4	3	3	2	3	2	3	4	4	37.5	4.00

References

- [1] Abel, F., Heckmann, D., Herder, E., Hidders, J., Houben, G-J., Krause, D., Leonardi, E., van der Sluijs, K., A Framework for Flexible User Profile Mashups, *Workshop on Adaptation and Personalization for Web 2.0 (AP-WEB 2.0)*, 2009.
- [2] ADL Co-Laboratory, The SCORM 2004 Overview, <http://www.adlnet.org/Technologies/scorm/default.aspx>, viewed 07/10/2009.
- [3] AHA! Adaptive Hypermedia For All, <http://aha.win.tue.nl>, viewed 07/10/2009.
- [4] Albert, D., Nussbaumer, A., Steiner, C., Using Visual Guidance and Feedback Based on Competence Structures for Personalising E-Learning Experience, *International Conference on Computers in Education (ICCE)*, 2008.
- [5] Albert, D., Nussbaumer, A., Steiner, C., M., Hendrix, M., Cristea, A., Design and Development of an Authoring Tool for Pedagogical Relationship Types between Concepts, *International Conference on Computers in Education (ICCE)*, 2009.
- [6] Alessi, S., Trollip, S., Multimedia for Learning: Methods and Development, *Allyn and Bacon, third edition*, 2000.
- [7] Anderson J.R., Reiser B.J., The Lisp Tutor, *Byte, vol.10*, 159-75, 1985.
- [8] Anderson, L.W., Krathwohl, D.R., Airasian, P.W., Cruikshank, K.A., Mayer, R.E., Pintrich, P.R., Raths, J., Wittrock, M.C., A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives, *Allyn and Bacon, complete edition*, 2000.
- [9] Avgeriou, P., Vogiatzis, D., Tzanavari, A., Retalis, S., Towards a Pattern Language in Adaptive Web-Based Educational Systems, *Advanced Technology for Learning*, 2004.
- [10] Beagle website, http://beagle-project.org/Main_Page, viewed 07/10/2009.
- [11] Beagle++ website, <http://beagle2.kbs.uni-hannover.de/>, viewed 07/10/2009.
- [12] Bechhofer, S., OWL Reasoning Examples, 2003, <http://owl.man.ac.uk/2003/why/latest/>, viewed 12/10/2009.
- [13] Berlanga, A., Garcia, F.J., Towards Reusable Adaptive Rules, *Adaptive Hypermedia and Collaborative Web-based Systems (AHCW)*, 2004.

- [14] Berners-Lee, T., Hendler, J., Lassila, O., The Semantic Web, *Scientific American Magazine*, 17 May 2001.
- [15] Blackboard LMS, <http://www.blackboard.com/us/index>, viewed 07/10/2009
- [16] Bonwell, C., Eison, J., Active Learning: Creating Excitement in the Classroom, *AEHE-ERIC Higher Education Report 1*, 1991.
- [17] Boticario, J.G., Santos, O.C., A dynamic assistance approach to support the development and modelling of adaptive learning scenarios based on educational standards, *Authoring of Adaptive and Adaptable Educational Hypermedia, A3EH, Workshop at, User Modelling*, 2007.
- [18] Broekstra, J., Kampman, A., SeRQL: An RDF Query and Transformation, *International Semantic Web Conference (ISWC)*, 2004.
- [19] Brooke, J., SUS: a "quick and dirty" usability scale, *Jordan, P.W., Thomas, B., Weerdmeester, B.A., McClelland, A.L., Usability Evaluation in Industry*, Taylor and Francis, 1996.
- [20] Brown, E., Cristea, A.I., Stewart, C., Brailsford, T., Patterns in Authoring of Adaptive Educational Hypermedia: A Taxonomy of Learning Styles, *"Education Technology and Society", Special Issue on Authoring of Adaptive Educational Hypermedia, (Volume 8, Issue 3)*, 2005.
- [21] Brusilovsky, P., Adaptive hypermedia, *User Modeling and User Adapted Interaction, Ten Year Anniversary Issue (Alfred Kobsa, ed.) 11 (1/2)*, 87-110, 2001.
- [22] Brusilovsky, P., Rizzo, R., Accessing Web educational resources from mobile wireless devices: The Knowledge Sea approach, *Mobile HCI workshop*, 2004.
- [23] Brusilovsky, P., Sosnovsky, S., Yudelson, M., Adaptive Hypermedia Services for E-Learning, *Applying Adaptive Hypermedia Techniques to Service Oriented Environments at Adaptive Hypermedia and Adaptive Web-Based Systems (AH)*, 2004.
- [24] Buneman P., Davidson, S., Kosky, A., Theoretical Aspects of Schema Merging, *Extending Database Technology*, 1992.

- [25] Burgos, D., Step by step. How to install CopperCore, how to publish and run a UoL, <http://hdl.handle.net/1820/327>, 2004, viewed 07/10/2009.
- [26] Burgos, D., Koper, R., Practical pedagogical uses of IMS Learning Design's Level B, *SIGOSSEE*, 2005.
- [27] Burgos, D., Tattersall, C., Koper, R., Representing adaptive eLearning strategies in IMS Learning Design, *TENCompetence Conference*, 2006.
- [28] Burgos, D., Tattersall, C., Koper, R., How to represent adaptation in eLearning with IMS Learning Design, *Interactive Learning Environments*, 15(2), 161-170, 2007.
- [29] Cannataro, M., Pugliese, A., XAHM: an XML-based Adaptive Hypermedia Model and its Implementation, *Adaptive Hypertext and Hypermedia (AH) workshop at ACM Conference on Hypertext and Hypermedia*, 2001.
- [30] Ceri, S., Fraternali, P., Bongio, A., Web Modeling Language (WebML): a modeling language for designing Web sites, *Web Engineering, Workshop at International Conference on Software Engineering (ICSE)*, 2000.
- [31] Chen, C.-T., Cheng, Y.C., Hsieh, C.-Y., Towards a Pattern Language Approach to Establishing Personal Authoring Environments in E-Learning., *Web-based Education*, 2007.
- [32] Chirita, P.-A., Costache, S., Nejdl, W., Paiu, R., Beagle++: Semantically Enhanced Searching and Ranking on the Desktop, *European Semantic Web Conference (ESWC)*, 2006.
- [33] Chirita, P.-A., Gavriloaie, R., Ghita, S., Nejdl, W., Paiu, R., Activity-Based Metadata for Semantic Desktop Search, *European Semantic Web Conference (ESWC)*, 2005.
- [34] Coffield, F.C., Moseley, D.V.M., Hall, E., Ecclestone, K., Should We Be Using Learning Styles? What Research Has to Say to Practice, *Learning and Skills Research Centre*, 2004.
- [35] Conlan, O., Hockemeyer, C., Wade, V., Albert, D., Gargan, M., An Architecture for Integrating Adaptive Hypermedia Services with Open Learning Environments,

World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA), 2002.

- [36] Conlan, O., Wade, V., Evaluation of APeLS - An Adaptive eLearning Service Based on the Multi-Model, Metadata-Driven Approach, *Adaptive Hypermedia and Adaptive Web-Based Systems (AH)*, 2004.
- [37] Cristea, A.I., Adaptive Course Creation for All, *International Conference on Information Technology (ITCC)*, 2004.
- [38] Cristea, A.I., Authoring of Adaptive and Adaptable Educational Hypermedia: where are we now and where are we going?, *Web Based Education (WBE)*, 2004.
- [39] Cristea, A.I., Burgos, D., Authoring Adaptive Hypermedia and IMS Learning Design: A Possible Understanding, *Conference on Advanced Learning Technologies (ICALT)*, 2006.
- [40] Cristea, A.I., Cristea, P., Evaluation of Adaptive Hypermedia Authoring Patterns during a Socrates Programme Class, *International Journal Advanced Technology For Learning 1(2)*, 2004.
- [41] Cristea, A.I., De Mooij, A., LAOS: Layered WWW AHS Authoring Model and its corresponding Algebraic Operators, *World Wide Web conference (WWW), Alternate Education track*, 2003.
- [42] Cristea, A.I., De Mooij, A., Adaptive Course Authoring: My Online Teacher, *International Conference on Telecommunications (ICT)*, 2003.
- [43] Cristea, A.I., Calvi, L., The three Layers of Adaptation Granularity, *User Modelling (UM)*, 2003.
- [44] Cristea, A.I., Ghali, F., Joy, M., Social, Personalized Lifelong Learning E-Infrastructures and Technologies for Lifelong Learning (ETLL), *IGI Global*, to appear, 2010.
- [45] Cristea, A.I., Smits, D., Bevan, J., Hendrix, M., LAG 2.0: Refining a reusable Adaptation Language and Improving on its Authoring, *European Conference on Technology Enhanced Learning (ECTEL)*, 2009.
- [46] Cristea, A.I., Smits, D., De Bra, P., Writing MOT, Reading AHA! - converting between an authoring and a delivery system for adaptive educational hypermedia,

Authoring of Adaptive & Adaptable Educational Hypermedia (A3EH), Workshop at Artificial Intelligence in Education (AIED), 2005.

- [47] Cristea, A.I., Stewart, C., Automatic Authoring of Adaptive Educational Hypermedia, *book chapter in Web-Based Intelligent e-Learning Systems: Technologies and Applications*, 2005.
- [48] Cristea, A.I., Verschoor, M., The LAG Grammar for Authoring the Adaptive Web, *International Conference on Information Technology (ITCC)*, 2004.
- [49] Cristea, A.I., Wentzler, A., Heuvelman, E., De Bra, P., Adapting SME Learning Environments for Adaptivity, *International Conference on Advanced Learning Technologies (ICALT)*, 2006.
- [50] Dagger, D., Personalised eLearning Development Environment, *PhD thesis, Trinity College Dublin*, 2006.
- [51] Dagger, D., Conlan, O., Wade, V., An Architecture for Candidacy in Adaptive eLearning Systems to Facilitate the Reuse of Learning Resources, *World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education (E-Learn)*, 2003.
- [52] De Bra, P., Aerts, A., Berden, B., De Lange, B., Rousseau, B., Stanic, T., Smits, D., Stash, N., AHA! The Adaptive Hypermedia Architecture, *ACM Hypertext Conference*, 2003.
- [53] De Bra, P., Ruiter, J.P., AHA! Adaptive Hypermedia for All, *WebNet Conference*, 2001.
- [54] De Bra, P., Smits, D., Stash, N., The Design of AHA!, *ACM Hypertext Conference*, 2006.
- [55] De Bra, P., Calvi, L., AHA! An open Adaptive Hypermedia Architecture, *New Review of Hypermedia and Multimedia*, vol. 4, 115-139, 1998.
- [56] De Bra, P., Houben, G.J., Wu, H., AHAM: A Dexter-based Reference Model for Adaptive Hypermedia, *ACM Conference on Hypertext and Hypermedia*, 1999.
- [57] De Troyer, O., Kleineremann, F., Pellens, B., Ewais, A., Supporting Virtual Reality in an Adaptive Web-based Learning Environment, *European Conference on Technology Enhanced Learning (ECTEL)*, 2009.

- [58] de Vrieze, P., van Bommel, P., van der Weide, T., A Generic Adaptive Model, *Adaptive Hypermedia, (AH)*, 2004.
- [59] Dolog, P., Gavriloaie, R., Nejdl, W., Brase, J., Integrating adaptive hypermedia techniques, *World Wide Web Conference (WWW)*, 2003
- [60] Dong, X., Halevy, A., Malleable Schemas, *Workshop on the Web and Databases (WebDB)*, 2005.
- [61] Dublin Core Metadata Initiative, <http://www.dublincore.org>, viewed 21/09/2009.
- [62] Duval, E., Forte, E., Cardinaels, K., Verhoeven, B., Van Durm, R., Hendrikx, K., Wentland-Forte, M., Ebel, N., Macowicz, M., Warkentyne, K., Haenni, F., The ARIADNE Knowledge Pool System, *Communications of the ACM* 44 (5), pp. 73-78, 2001.
- [63] Edwards, D., M., Hardman, L., Lost In Hyperspace: Cognitive Mapping And Navigation In A Hypertext Environment., *Hypertext: Theory in Practice*, 105-125, 1988.
- [64] Eklund, J., Brusilovsky, P., InterBook: An Adaptive Tutoring System, *UniServe Science News Vol. 12.*, 1999.
- [65] Fegen, N., What is IMS Vocabulary Definition Exchange, 2006, <http://metadata.cetis.ac.uk/guides/WhatIsVDEX.pdf>, viewed 07/10/2009.
- [66] Felder, R.M., Silverman, L.K., Learning and teaching styles in engineering education, *Engineering Education*, 78(7), pp. 674-681, 1988
- [67] Foss, J.G.K., Cristea, A.I., Adaptive Hypermedia Content Authoring using MOT3.0, *Authoring of Adaptive and Adaptable Hypermedia Workshop (A3H), Workshop at European Conference on Technology-Enhanced Learning (ECTEL)*, 2009.
- [68] Freire, M., Rodríguez, P., Comparing Graphs and Trees for Adaptive Hypermedia Authoring, *Authoring of Adaptive and Adaptable Educational Hypermedia (A3EH), Workshop at Artificial Intelligence in Education (AIED)*, 2005.
- [69] Frequently Asked Questions about the University's use of the Pattern Language, University of Oregon's Planning Office

<http://www.uoregon.edu/~uplan/faq/FAQPatternLanguage.html>, viewed 07/10/2009.

- [70] Gardarin, G., Dang-Ngoc, T-T., Mediating the Semantic Web, *Journal on Extraction and Management of Knowledge*, 2004.
- [71] Garzotto F., Cristea A.I., ADAPT: Major design dimensions for educational adaptive hypermedia, *World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA)*, 2004.
- [72] Ghali, F., Cristea, Hendrix, M., Augmenting E-Learning Standards with Adaptation, *Conference on Computers and Advanced Technology in Education (CATE)*, 2008.
- [73] Ghali F., Cristea, A.I., Social Reference Model for Adaptive Web Learning, *International Conference on Web-based Learning (ICWL)*, 2009.
- [74] Ghita, S., Nejdl, W., Paiu, R., Semantically Rich Recommendations in Social Networks for Sharing, Exchanging and Ranking Semantic Context, *International Semantic Web Conference (ISWC)*, 2005.
- [75] Glahn, C., Steiner, C., Verpoorten, D., Docq, F., Mazzola, L., GRAPPLE Deliverable 9.3, First Empirical Evaluation in Higher Education Settings, <http://www.grapple-project.org/deliverables>.
- [76] Google Scholar, <http://scholar.google.com/>, viewed 16/10/2009.
- [77] Gutierrez, S., Authoring of Adaptive Sequencing for IMS-LD, *Authoring of Adaptive and Adaptable Educational Hypermedia (A3EH), Workshop at User Modelling (UM)*, 2007.
- [78] Hachichi, A., Thomas, G., Martin, C., Patarin, S., Folliot, B., A Generic Language for Dynamic Adaptation, *European Conference on Parallel Processing*, 2005.
- [79] Halasz F., Schwartz, M., The Dexter hypertext reference model: Hypermedia, *Communications of the ACM*, 37(2), 30-39, 1994.
- [80] Harrigan, M., Kravčik, M., Steiner, C., Wade, V., What Do Academic Users Really Want from an Adaptive Learning System, *User Modeling, Adaptation, and Personalization (UMAP)*, 2009.

- [81] Harrigan, M., Wade, V., Towards a Conceptual and Service-Based Adaptation Model, *Dynamic and Adaptive Hypertext: Generic Frameworks, Approaches and Techniques, Workshop at ACM Conference on Hypertext and Hypermedia*, 2009.
- [82] Hendrix, M., Nussbaumer, A., Dicerto, M., Oneto, L., Cristea, A.I., GAT: The FP7 GRAPPLE Authoring Tool Set, *European Conference on Technology Enhanced Learning (ECTEL)*, 2009.
- [83] Hendrix, M., Cristea, A.I., Stewart, C., Adaptation languages for learning: the CAM meta-model, *International Conference on Advanced Learning Technologies (ICALT)*, 2009.
- [84] Hendrix, M., De Bra, P., Pechenizkiy, M., Smits, D., Cristea, A., Defining adaptation in a generic multi layer model: CAM: The GRAPPLE Conceptual Adaptation Model, *European Conference on Technology Enhanced Learning (ECTEL)*, 2008.
- [85] Hendrix, M., Cristea, A.I., Burgos, D., Comparative Analysis of Adaptation in Adaptive Educational Hypermedia and IMS-Learning Design, *Technical Report RR443*, 2009.
- [86] Hendrix, M., Cristea, A.I., Reuse Patterns in Adaptation Languages: Creating a meta-level for the LAG adaptation language, *Adaptive Hypermedia and Adaptive web-based systems (AH)*, 2008.
- [87] Hendrix, M., Cristea, A.I., A meta level for LAG Adaptation Language re-use in education, *Adaptable Hypermedia Workshop (A3H), workshop at Adaptive Hypermedia and Adaptive Web-Based Systems (AH)*, 2008.
- [88] Hendrix, M., Cristea, A.I., Meta-Levels of Adaptation in Education, *Computers and Advanced Technology in Education (CATE)*, 2008.
- [89] Hendrix, M., Cristea, A.I., A Spiral Model for Adding Automatic Adaptive Authoring to Adaptive Hypermedia, *Journal of Universal Computer Science (J.UCS)*, volume 14, 2008.
- [90] Hendrix, M., Cristea, A.I., Joy, M., Evaluating the automatic and manual creation process of adaptive lessons, *International Conference on Advanced Learning Technologies (ICALT)*, 2007.

- [91] Hendrix, M., Cristea, A.I., Evaluating Adaptive authoring of Adaptive Hypermedia, *Adaptive and Adaptable Educational Hypermedia (AEH), Workshop at User Modelling (UM)*, 2007.
- [92] Hendrix, M., Cristea, A., A qualitative and quantitative evaluation of Adaptive authoring of Adaptive Hypermedia, *European Conference on Technology Enhanced Learning (ECTEL)*, 2007.
- [93] Hendrix, M., Cristea, A.I., Nejdl, W., Authoring Adaptive Educational Hypermedia on the Semantic Desktop, *International Journal of Learning Technology (IJLT)*, volume 3, 2007.
- [94] Hendrix, M., Cristea, A.I., Nejdl, W., Automatic and Manual Annotation Using Flexible Schemas for Adaptation on the Semantic Desktop, *European Conference on Technology Enhanced Learning (ECTEL)*, 2006.
- [95] Hendrix, M., Cristea, A.I., Nejdl, W., Authoring Adaptive Learning Material on the Semantic Desktop, *Authoring of Adaptive and Adaptable Educational Hypermedia (A3EH), Workshop at Adaptive Hypermedia and Adaptive Web-based Systems*, 2006.
- [96] Hepworth, M., How is knowledge about the consumer of information being applied in the design and delivery of information products and services, *Asia-Pacific Conference on Library & Information Education & Practice (A-LIEP)*, 2006.
- [97] Hillmann, D., Dublin Core User guide, 2005, <http://dublincore.org/documents/usageguide/>, viewed 07/10/2009.
- [98] IEEE, The Learning Object Metadata (LOM) Specification, <http://www.ieeeltsc.org/standards.OM>, Learning Object Metadata, IEEE-LTSC, viewed 21/09/2009.
- [99] IMS, IMS Content Packaging, <http://www.imsglobal.org/content/packaging/>, 2004.
- [100] IMS, IMS Learning Design, <http://www.imsglobal.org/learningdesign/>, 2003.
- [101] IMS, IMS Question and Test Interoperability, <http://www.imsglobal.org/question/>, 2004.

- [102] IMS, IMS-VDEX., <http://www.imsglobal.org/vdex/>, 2004.
- [103] Jones, E.R., Implications of SCORM™ and emerging e-learning standards on engineering education, *ASEE Gulf-Southwest Annual Conference*, 2002.
- [104] Koch, N., Wirsing, M., The Munich Reference Model for adaptive hypermedia applications, *Adaptive Hypermedia and Adaptive Web-Based Systems (AH)*, 2002.
- [105] Koper, R., Tattersall, C., Learning Design: A handbook on modelling and delivering networked education and training, *Springer Verlag*, 2005.
- [106] Kostelník, R., Bieliková, M., Web-Based Environment using Adapted Sequences of Programming Exercises, *Information Systems Implementation and Modelling (ISIM)*, 2003.
- [107] LAMS, <http://www.lamsinternational.com>, viewed 07/10/2009.
- [108] Lave, J., Wenger, E., Situated Learning: Legitimate Peripheral Participation, *Cambridge University Press*, first edition, 1991.
- [109] LN4LD, Units of Learning developed by several authors at Learning Network for Learning Design of The Open University of The Netherlands, <http://dspace.ou.nl>, viewed 07/10/2009 .
- [110] Martin, B., Mitrovic, A., Suraweera, P., Domain Modelling with Ontology: A Case Study, *Authoring of Adaptive and Adaptable Educational Hypermedia, A3EH, Workshop at, User Modelling*, 2007.
- [111] Mazzola, L., Mazza, R., An infrastructure for creating graphical indicators of the learner profile by mashing up different sources, *Advanced Visual Interfaces (AVI)*, 2010.
- [112] McGuinness, van Harmelen, Web Ontology Language (OWL) Overview, <http://www.w3.org/TR/owl-features/>, 2004
- [113] Meccawy, M., Stewart, C., Ashman, H., Adaptive Educational Hypermedia Content Creation: A Web Service based Architecture, *Authoring of Adaptive and Adaptable Educational Hypermedia (A3EH), Workshop at Adaptive Hypermedia and Adaptive Web-Based Systems (AH)*, 2006.
- [114] Merceron, A., Yacef, K., A Web-based tutoring tool with mining facilities to improve learning and teaching, *Artificial Intelligence in Education (AI-Ed)*, 2003.

- [115] Miliband, D., Personalised Learning: Building a New Relationship with Schools, *North of England Education Conference*, 2004.
- [116] Moreno-Ger, P., Martínez-Ortiz, I., Sierra, J.L., Fernández-Manjón, B., Language-Driven Development of Videogames: The <e-Game> Ex-perience, *International Conference in Entertainment Computing (ICEC)* 2006.
- [117] Moore, A., Brailsford, T.J., Stewart, C.D., Personally tailored teaching in WHURLE using conditional transclusion, *ACM conference on Hypertext and Hypermedia*, 2001.
- [118] Moore, A., Stewart, C.D., Zakaria, M.R., Brailsford, T.J., WHURLE - an adaptive remote learning framework, *International conference on Engineering Education (ICEE)*, 2003.
- [119] MOT, <http://prolearn.dcs.warwick.ac.uk/mot.html>, viewed 07/10/2009
- [120] OASIS, Web Services for Remote Portlets (WSRP), *Technical report*, <http://docs.oasis-open.org/wsrp>, viewed 28/01/2010.
- [121] Object Constraint Language Specification, version 2.0, <http://www.omg.org/technology/documents/formal/ocl.htm>, viewed 16/02/2010.
- [122] Ohene-Djan J., A Formal Approach to Personalisable, Adaptive Hyperlink-Based Interaction, *PhD thesis, Department of Computing, Goldsmiths College, University of London*, 2000.
- [123] O’Keeffe, I., Conlan, O., Wade, V., A Unified Approach to Adaptive Hypermedia Personalisation and Adaptive Service Composition, *Adaptive Hypermedia and Adaptive Web-Based Systems (AH)*, 2006.
- [124] O’Keeffe, I., Wade, V., Personalised Web Experiences: Seamless Adaptivity across Web Service Composition and Web Content, *User Modeling, Adaptation, and Personalization (UMAP)*, 2009.
- [125] Oneto, L., Herder, E., Abel, F., Smits, D., Making today's Learning Management Systems adaptive, *Learning Management Systems meet Adaptive Learning Environments, Workshop at European Conference on Technology Enhanced Learning (ECTEL)*, 2009.

- [126] Pacurar, E., Trigano, P., Alupoaie, S., A QTI editor integrated into the netUniversité web portal using IMS LD, *Journal of Interactive Media in Education, Advances in Learning Design*, 2005.
- [127] Paramythis, A., Cristea, A.I., Towards Adaptation Languages for Adaptive Collaborative Learning Support, *Individual and Group Adaptation in Collaborative Learning Environments, Workshop at European Conference on Technology Enhanced Learning (ECTEL)*, 2008.
- [128] Paredes, P., Rodriguez, P., A Mixed Approach to Modelling Learning Styles in Adaptive Educational Hypermedia, *Journal (208) Advanced Technology for Learning, 1(4)*, 2004.
- [129] Resource Description Format, W3C Recommendation, <http://www.w3.org/RDF/>, 2004.
- [130] RDF Primer, W3C Recommendation, 2004.
- [131] Rey-Lopez, M., Diaz-Redondo, R.P., Fernandez-Vilas, A., Pazos-Arias, J.J., Garcia-Duque, J., Gil-Solla, A., Ramos-Cabrera, M., An extension to the ADL SCORM standard to support adaptivity: The t-learning case-study, *Computer Standards & Interfaces, Volume 31, Issue 2, pages 309-318*, 2009.
- [132] Sauermann, L., Bernardi, A., Dengel, A., Overview and outlook on the Semantic Desktop, *Semantic Desktop, Workshop at International Semantic Web Conference (ISWC)*, 2005.
- [133] Schlieder, T., Naumann, F., Approximate tree embedding for querying xml data, Heterogeneous and Distributed Information Retrieval, *ACM SIGIR Workshop on CML and Information Retrieval*, 2002 .
- [134] Scotton, J., Moebs, S., McManis, J., Cristea, A.I., A Case Study on Merging Strategies for Authoring QoE-based Adaptive Hypermedia, *Authoring of Adaptive and Adaptable Hypermedia Workshop (A3H), Workshop at European Conference on Technology-Enhanced Learning (ECTEL)*, 2009.
- [135] Semantic Desktop, <http://Semantic Desktop>, viewed 07/10/2009.
- [136] Stash, N., Cristea, A.I., De Bra, P., Adaptation Languages as vehicles of explicit intelligence in Adaptive Hypermedia, *International Journal on Continuing*

- Engineering Education and Life-Long Learning*, vol. 17, nr 4/5, pp. 319-336, 2007.
- [137] Steiner, C., Nussbaumer, A., Albert, D., Supporting self-regulated personalised learning through Competence-based Knowledge Space Theory, *Policy Futures in Education*, Volume 7 Number 6, 2009.
- [138] Shasha, D., Wang, J., Shan, H., Zhang, K., Atreegrep: Approximate searching in unordered trees, *Scientific and Statistical DB Management*, 2002.
- [139] Strauss, A., Corbin, J., Basics of qualitative research, *Grounded theory procedures and techniques*, Newbury Park: SAGE Publications, 1990.
- [140] SMIL, <http://www.w3.org/AudioVideo/>, viewed 07/10/2009
- [141] Spivey, J.M., The Z Notation, *Prentice-Hall International, Hertfordshire, England*, 1989.
- [142] The Open University., The Sled player, <http://sled.open.ac.uk>, viewed 07/10/2009.
- [143] Underwood, J. et al., Personalised Learning, http://partners.becta.org.uk/upload-dir/downloads/page_documents/research/reports/personalised_learning.odt, viewed 23/11/2009.
- [144] van der Aalst, W., ter Hofstede, A., Kiepuszewski, B., Barros, A., Workflow Patterns, *Distributed and Parallel Databases*, 14(3):5–51, 2003.
- [145] van der Sluijs, K., Hidders, J., Leonardi, E., Houben G.J., Generic Adaptation Language for describing Adaptive Hypermedia, *Workshop on Dynamic and Adaptive Hypertext: Generic Frameworks, Approaches and Techniques*, 2009.
- [146] Van der Vegt, W., CopperAuthor, *Open University of The Netherlands: Heerlen.*, 2005.
- [147] Vialardi, C., Bravo, J., Ortigosa, A., Empowering AEH Authors Using Data Mining Techniques, *Authoring of Adaptive and Adaptable Educational Hypermedia (A3EH), Workshop at User Modelling (UM)*, 2007.
- [148] Vogten, H., Martens, H., CopperCore 3.1, *Open University of The Netherlands: Heerlen.*, <http://www.coppercore.org>, viewed 07/10/2009.

- [149] Vasilyeva, E., Pechenizkiy, M., De Bra, P., Adaptation of Feedback in e-learning System at Individual and Group Level, *Personalisation in E-Learning Environments at Individual and Group Level (PING), Workshop at User Modelling (UM)*, 2007.
- [150] Weibelzahl, S., Jedlitschka, A., Ayari, B., Eliciting Requirements for an Adaptive Decision Support System through Structured User Interviews, *Authoring of Adaptive and Adaptable Educational Hypermedia (A3EH), Workshop at Adaptive Hypermedia and Adaptive Web-Based Systems (AH)*, 2006.
- [151] Wentzler, A., Cristea, A.I., Heuvelman, E., De Bra, P., A Step towards Commercial Adaptive Hypermedia, *Authoring of Adaptive and Adaptable Educational Hypermedia (A3EH), Workshop at Adaptive Hypermedia and Adaptive Web-Based Systems (AH)*, 2006.
- [152] Wordnet, <http://www.cogsci.princeton.edu/wn/>, viewed 07/10/2009.
- [153] Wu, H., A Reference Architecture for Adaptive Hypermedia Applications, *PhD thesis, Eindhoven University of Technology, The Netherlands, ISBN 90-386-0572-2*, 2002.
- [154] Workflow Management Coalition Terminology and Glossary, *Workflow Management Coalition (WfMC)*, 1999.
- [155] Wu, H., De Kort, E., De Bra, P., Design Issues for General-Purpose Adaptive Hypermedia Systems, *ACM Conference on Hypertext and Hypermedia*, 2001.
- [156] Zakaria, M.R., Moore, A., Stewart, C.D., Brailsford, T.J., "Pluggable" user models for adaptive hypermedia in education., *ACM Conference on Hypertext and Hypermedia*, 2003.
- [157] Zukerman, I., Litman, D., Natural language processing and user modeling: synergies and limitations, *User Modeling and User Adapted Interaction 11(1-2)*, 129-158, 2001.