Light-weight Ontologies for Scrutable User Modelling

Andrew Wai Kwong Lum

This thesis submitted to the University of Sydney in fulfillment of the requirements for the degree of Doctor of Philosophy

May 2007
Abstract

This thesis is concerned with the ways light-weight ontologies can support scrutability for large user models and the user modelling process. It explores the role that light-weight ontologies can play, and how they can be exploited, for the purpose of creating and maintaining large, scrutable user models consisting of hundreds of components. We address problems in four key areas: ontology creation, metadata annotation, creation and maintenance of large user models, and user model visualisation, with a goal to provide a simple and adaptable approach that maintains scrutability. Each of these key areas presents a number of challenges that we address.

Our solution is the development of a toolkit, LOSUM, which consists of a number of tools to support the user modelling process. It incorporates light-weight ontologies to fulfill a number of roles: aiding in metadata creation, providing structure for large user model visualisation, and as a means to reason across granularities in the user model. In conjunction with this, LOSUM also features a novel visualisation tool, SIV, which performs a dual role of ontology and user model visualisation, supporting the process of ontology creation, metadata annotation, and user model visualisation.

We evaluated our approach at each stage with small user studies, and conducted a large scale integrative evaluation of these approaches together in an authentic learning context with 114 students, of whom 77 had exposure to their learner models through SIV. The results showed that students could use the interface and understand the process of user model construction. The flexibility and adaptability of the toolkit has also been demonstrated in its deployment in several other application areas.
Acknowledgements

Firstly my supervisor Judy Kay, whose comments, criticisms, inspiration and encouragement have propelled me this far.

This thesis would not be possible without the funding from Hewlett Packard, and also their generosity for giving me the chance to visit their labs. In particular I would like to thank the people I have worked with from Hewlett Packard: Gavin Brebner, Bernard Burg, and Claudio Bartolini. I would also like to thank the Semantic Web Research Group at Bristol for their invaluable feedback and time.

To all the people I have collaborated with at the University of Sydney, especially Trent Apted, Daren Ler, Bruce Li, William Niu, Sam Holden and James Uther. Collaboration spawns endeavor, and working with all of you during this research has not only lead to new ideas and tools, but also improved the quality of previous ones.

The people in the user modelling and AIED community, who have given invaluable feedback from the time I started to the time I submitted. In particular I would like to thank Susan Bull, Juan Diego Zapata-Rivera, Jill Kliger Saliba, Dominik Heckmann, Christo Dichev, Harold Boley, Virendra Bhavsar, Anurag Singh and Ricardo Mazza.

Last but not least, my family and friends, who have been with me during this time, both through the hard times and the good.
Publications

The following is a selected list of refereed publications of work relating to this thesis.

**Book Chapters**


**Edited Proceedings**


**Full Papers at Major International Conferences**


**Short Papers at Major International Conferences**

Workshop and Other Papers


# Table of Contents

Abstract .................................................................................................................................................. III

Acknowledgements ................................................................................................................................. V

Publications ............................................................................................................................................. VII

Table of Contents ................................................................................................................................... IX

List of Figures .......................................................................................................................................... XV

List of Tables .......................................................................................................................................... XXI

Introduction .............................................................................................................................................. 1

1.1 Motivation ......................................................................................................................................... 1

1.1.1 A Course in User Interfaces ......................................................................................................... 1

1.1.2 Shopping for Cameras .................................................................................................................... 4

1.2 Motivation ......................................................................................................................................... 5

1.3 Identification of Key Problems .......................................................................................................... 9

1.4 Light-weight Ontologies for Scrutable User Modelling (LOSUM) .................................................. 16

1.5 Contributions .................................................................................................................................... 20

Background ............................................................................................................................................ 23

2.1 Ontologies and User Modelling ......................................................................................................... 23

2.1.1 Ontologies and the Semantic Web ................................................................................................. 24

2.1.2 Ontologies to Structure Data ......................................................................................................... 26

2.1.3 Ontological Reasoning .................................................................................................................... 28

2.1.4 Ontology Learning .......................................................................................................................... 29

2.1.5 Ontologies and Metadata ............................................................................................................. 32

2.1.6 Metadata Annotation ..................................................................................................................... 33

2.2 Scrutable User Modelling .................................................................................................................. 37

2.2.1 Learner Modelling Systems ............................................................................................................ 38

2.2.2 Visualisation of User Models .......................................................................................................... 41

2.2.3 Visualisation of Groups .................................................................................................................. 48

2.3 Summary .......................................................................................................................................... 54

Theoretical Framework and Approach ..................................................................................................... 55

3.1 Theoretical Overview of LOSUM ....................................................................................................... 56
Applications of LOSUM

9.1 SITS-VLUM – A Visualisation-based Teaching Agent for SITS ................................. 185
  9.1.1 SITS-VLUM Interface ................................................................. 186
9.2 SITS-VLUM Implementation.................................................................................. 189
  9.2.1 Insights ..................................................................................... 190
9.3 Metasaur ................................................................................................................. 190
9.4 Assess ..................................................................................................................... 191
  9.4.1 Self-assessment Tasks ............................................................................... 192
9.5 Advanced Technologies for Learning Course .......................................................... 196
  9.5.1 MeCureo online interface ........................................................................... 196
9.5.2 MyMuseum ............................................................................................... 197
  9.5.3 Insights ......................................................................................... 200
9.6 TARO ................................................................................................................ 200
  9.6.1 Insights ......................................................................................... 203
9.7 Summary .............................................................................................................. 203

Conclusions .................................................................................................................. 205
10.1 Contributions ........................................................................................................ 206
10.2 Future Directions and Limitations ........................................................................ 208
  10.2.1 Domain ......................................................................................... 208
  10.2.2 Ontology ....................................................................................... 208
  10.2.3 User Modelling Server ......................................................................... 209
  10.2.4 User Model Visualisation ........................................................................ 210
  10.2.5 Evaluation ...................................................................................... 211
10.3 Summary .............................................................................................................. 212

Exploration of Local Definition Linkage in MeCureo..................................................... 213
  A.1 Concept Name Only...................................................................................... 214
  A.2 Concept and Definition ............................................................................... 215
  A.3 Concept, Definition and Related .................................................................... 217

Analysis of Web Log Data ......................................................................................... 219
  B.1 Data ............................................................................................................ 219
    B.1.1 GOMS Online Lecture Data (2003-2005) ......................................... 220
    B.1.2 Cognitive Walkthrough Online Lecture Data (2003-2005) ................. 223
B.1.3 Summary of Online Lecture Access Averages (2003-2005) ........................................226
B.2 Analysis ......................................................................................................................227

UI-SIV Evaluation Instructions ..........................................................................................229
C.1 Task 0: Introduction .................................................................................................230
C.2 Task 1: Colour ..........................................................................................................231
C.3 Task 2: Font Size ......................................................................................................232
C.4 Task 3: Searching .....................................................................................................233
C.5 Task 4: Evidence ......................................................................................................234
C.6 Task 5: Selecting a concept .....................................................................................235
C.7 Task 6: Comparing yourself to the average ............................................................236
C.8 Task 7: The Class Average .......................................................................................237
C.9 Task 8: Inferring about knowledge .........................................................................238
C.10 Task 9: Further comments ....................................................................................239
C.11 Stuvac Free Usage .................................................................................................240

UI-SIV Evaluation Raw Results and Processing ...............................................................241
D.1 Processing of Task Results .......................................................................................241
D.2 Interaction Scores and Exam Marks .........................................................................243
D.3 Interaction levels and exam marks (stuvac participants) ..........................................244
D.4 Summaries of Participant Interactions ....................................................................245
D.5 Interactions during stuvac .......................................................................................255
D.6 UI-SIV and UIDP website log analysis during stuvac .............................................257
   D.6.1 Log Analysis: S01 .............................................................................................257
   D.6.2 Log Analysis: S02 .............................................................................................259
   D.6.3 Log Analysis: P03 .............................................................................................261
   D.6.4 Log Analysis: P14 .............................................................................................262
   D.6.5 Log Analysis: P24 .............................................................................................263

References .........................................................................................................................265
List of Figures

Figure 1.1: A screenshot of one of the learning topics, showing a particular slide on Cognitive Walkthrough and controls to listen to the audio. ................................................................. 2
Figure 1.2: Jane’s user model showing a hierarchy of concepts and the scores associated with them... 3
Figure 1.3: Jane’s user model showing a hierarchy of concepts and the scores associated with them, showing her user model after reviewing learning material ....................................................... 3
Figure 1.4: An illustration of Bob’s user model (middle) created by mapping his user data (bottom left) to an ontology (top left), the information he has chosen to make public (top right) and the things that can be inferred about him from the public information (bottom right) ........................................... 8
Figure 1.5: An illustration of Jane’s learner model (middle) created by mapping her usage of the learning concept (bottom left) to an ontology of the course concepts (top left) where there is no evidence for the high level concepts. Inference allows Jane to get an overview of her knowledge in these concepts (middle right) ............................................................................................................. 9
Figure 1.6: Problems and goals associated with scrutable user modelling based on light-weight ontologies ........................................................................................................................................ 10
Figure 1.7: Interaction of the core components in LOSUM ................................................................................................................................. 17
Figure 2.1: Screenshot of the LOMGen user interface for adding metadata to learning objects ........ 36
Figure 2.2: Screenshot of Mr. Collins, showing a text-based negotiation system between the learner and the system ...................................................................................................................................... 38
Figure 2.3: Screenshot of STyLE-OLM, showing the collaboratively constructed graph and dialogue. ........................................................................................................................................ 39
Figure 2.4: Screenshot of SQL-Tutor, showing the learner knowledge as skillometers .................. 40
Figure 2.5: Screenshot of VIUM displaying a user model from a movie recommendation service. Lawrence of Arabia is currently selected. The system is fairly certain the user will enjoy this movie. 44
Figure 2.6: Screenshot of PeerGlass showing six articles as totem poles on the left hand side, and the model plane showing the observed interests on the right hand side ........................................................................ 45
Figure 2.7: Screenshot of the QV tool. The components of the user model are shown, with the nodes expanded for the SAM text editor ........................................................................................................ 47
Figure 2.8: Screenshot of ViSMod, showing part of a Bayesian student model. The system belief of the student knowledge level is represented by the colour and size of the nodes. .................................48

Figure 2.9: Screenshot of the CourseVis Discussion Graph showing social interaction between learners..............................................................50

Figure 2.10: Screenshot of the CourseVis Cognitive Matrix that shows a coloured grid representing student’s mastery of course topics. ...........................................................................51

Figure 2.11: Screenshot of a CourseVis Behaviour Graph, showing multiple sets of data on the vertical axis mapped against the date on the horizontal axis for an individual learner...........................................52

Figure 2.12: A PeopleGarden visualisation showing numerous PeopleFlowers at different stages of bloom, indicating their participation in the social space.................................................................53

Figure 3.1: Overview of LOSUM showing the system components discussed in this chapter highlighted in dark blue. ..................................................................................................................56

Figure 3.2: Overview of a learning topic. .................................................................................................................................59

Figure 3.3: A screenshot from the User Interface Design and Programming course website showing one of the audio slides from the User Centered Design lecture. .........................................................60

Figure 3.4: A transcript of the audio narrative from the slide shown in Figure 3.3.........................61

Figure 3.5: Screenshot of the SIV interface. The visualisation is at the left, with the concepts listed vertically. The concept user interface critique is in focus and has the largest font .........................67

Figure 3.6: The concept cognitive modelling is in focus, and shown at Term Expansion level 1, 2 and 5 (as indicated by the Term Expansion radio buttons on the top of the visualisation). As the expansion level increases, the number of visible related concepts is also shown..........................................................68

Figure 3.7: The concept storyboard is currently in focus (left screenshot). The user clicks the Select/Deselect button at the top of the visualisation, which moves the concept to the far left margin (middle screenshot). When the user clicks on other concepts, in this case GOMS, the selected concepts are still noticeable (right screenshot)........................................................................................................................................71

Figure 3.8: Searching for a term (left screenshot) produces a list of relevant results. Here the user searches for the term colour. The results become the focus (right screenshot) and the user can select the one that is relevant to their query. We can see the results colour theory and colour in the right screenshot; all other concepts are made smaller ........................................................72

Figure 4.1: System overview showing the SIV system component discussed in this chapter highlighted in dark blue. ................................................................................................................75
Figure 4.2: Pictorial representation of the way VIUM parses a single data file for the user model (top), whereas SIV reads in an ontology file and subsequently overlays it with additional user model data (bottom). ................................................................. 78

Figure 4.3: Sample of original RDF format used by VIUM. Of note is the combined domain and user knowledge. ................................................................. 79

Figure 4.4: Sample of ontology OWL format for the same concept shown in Figure 4.3 for SIV. Of note is the ontology only contains the domain data. User model data is serialized to a separate file... 80

Figure 4.5: Snippets from two separate user model overlay XML files, Mark (line A3) and Reliability (line B3), for the concept *Fuel use in cells*. The corresponding domain ontology snippet for this concept is shown in Figure 4.4. ........................................................................................................ 80

Figure 4.6: SIV showing the filtering of relationships on the visualisation. The menu allows us to choose which relationships to view (middle screenshot), and the results are filtered on the right screenshot. ........................................................................................................ 85

Figure 4.7: A selection of different control elements for SIV. ............................................. 86

Figure 5.1: System overview showing the core components of LOSUM.............................................. 91

Figure 5.2: Steps identified by Maedche and Staab (2001) for the ontology learning process........ 94

Figure 5.3: FOLDOC screenshot showing the definition for the term *declarative language*.......... 95

Figure 5.4: An excerpt from the Mecureo keywords configuration file. ................................. 97

Figure 5.5: Screenshot of the Mecureo Ontology Querier showing a query on the concept *declarative language*. Each concept is a node in the sub-graph, their weights are shown along side the concept name associated with that node........................................................................................................ 99

Figure 5.6: The contents of the node list frame from Figure 5.5, showing an expansion of the concept *declarative language* at threshold 0.75. ........................................................................................................ 100

Figure 5.7: Mecureo DOT output for the concept *declarative language* at distances 0.5 and 0.75... 102

Figure 5.8: The Mecureo concept and relationship definitions in OWL. ................................. 103

Figure 5.9: The OWL representation for the term *imperative language* showing relationships to other terms........................................................................................................ 104

Figure 5.10: SIV screenshot visualising the Mecureo generated ontology for *declarative language*. 104

Figure 5.11: Scenario for evaluating light-weight ontologies.................................................. 108

Figure 5.12: Screenshot of interface participants worked with. The concept Microsoft Word is currently selected. ........................................................................................................ 109
Figure 5.13: An entry for the concept *exploration* (declared on line 1). The second line is the URL identifier for the concept, followed by the definition and the related (existing) concepts in the dictionary and which categories this concept belongs to, respectively. ................................................. 114

Figure 5.14: The distribution of terms in our ontology grouped by the number of peers (fan-out) they have. There are a significant number of terms having a low peerage or an extremely high peerage. 116

Figure 5.15: A close-up of Figure 5.14, with cut-off at 20 peers ......................................................... 117

Figure 5.16: Janus Term editor, for editing the glossary definitions. The concepts are categorized on the left by the number of peers as generated by Mecureo ......................................................... 118

Figure 6.1: System overview showing the core components of LOSUM ......................................................... 122

Figure 6.2: Screenshot of the Metasaur interface. The SIV visualisation is on the left-hand side, allowing users to access the ontology as they examine the learning content on the right-hand side.. 125

Figure 6.3: User wishes to add metadata concepts to the slide labeled Empirical techniques. She does a search for the word *observation*; the resulting matches are shown on the visualisation ............................................. 126

Figure 6.5: User selects the concept observational study on the visualisation and clicks on Add Metadata Element to add the concept to the slide. ................................................................. 127

Figure 6.5: Close-up of the Existing Metadata panel after the concept observational study has been added to the slide metadata. ......................................................................................... 127

Figure 6.6: The user has observational study in focus, and sees the related concept ethnography (above left). The user clicks on ethnography to bring it into focus (above right). ............................ 128

Figure 6.7: Overview of the prototype Metasaur architecture ................................................................. 129

Figure 6.8: The Usability Glossary from the Usability First website showing the definition for the concept cognitive walkthrough ................................................................. 130

Figure 6.9: Instruction for participants to annotate a series of slide with metadata ............................................. 132

Figure 6.10: Slide 4 in the experiment .......................................................... 134

Figure 6.11: Slide 5 in the experiment .......................................................... 134

Figure 6.12: Overview of the improved Metasaur architecture ................................................................. 137

Figure 6.13: The revised Metasaur interface, with an added section for users to add their own dictionary definitions on the bottom right. The user has filled in a new local definition of the word exploration in the pop-up box ................................................................. 138

Figure 7.1: System overview showing the core components of LOSUM ......................................................... 142

XVIII
Figure 7.2: A student model with fine-grain evidence for learner knowledge of concepts in the HCI domain. ................................................................. 143

Figure 7.3: Overview of the onto-increment approach showing direct evidence in green, and onto-increment boost in blue. ........................................................................................................ 148

Figure 7.4: Personis block diagram showing flow of resolvers to produce final user model overlay. 151

Figure 7.5: The SIV interface used for the evaluation of the created user models. ................................. 152

Figure 7.6: The task description for the evaluation given to the participants. ........................................ 153

Figure 8.1: The SIV interface with tutorial frame. ................................................................. 161

Figure 8.2 Bar chart of the distribution of the students over the range of exam marks. Students in exam mark distribution have been divided into categories based on their access period. ............ 164

Figure 8.3: Graph of Exam Mark versus Interaction Level. .............................................................. 167

Figure 9.1: The SITS-VLUM interface, with the visualisation on the left and a document on the right. ................................................................................................................................. 186

Figure 9.2: SITS-VLUM with document SGI: priority_queue selected. Concepts relating to this document are shown on the right. ............................................................................................... 187

Figure 9.3: SITS-VLUM with concept DefaultConstructable selected. Documents relating to this concept are shown on the left ........................................................................................................ 188

Figure 9.4: Excerpt of a document type from the SITS-VLUM RDF serialisation. We can see on line 3 the designation of the component as a url .................................................................................. 189

Figure 9.5: Excerpt of a concept type from the SITS-VLUM RDF serialisation. We can see on line 3 the designation of the component as a concept ............................................................................... 189

Figure 9.6: A problem in Assess showing the steps required for self-evaluation................................. 192

Figure 9.7: Self-assessment criteria for the problem shown in Figure 9.6 ............................................. 193

Figure 9.8: An example solution from Assess for a task called count elements (upper) and the assessment criteria for the problem (lower) .................................................................................. 194

Figure 9.9: The User Profile page in Assess showing SIV on the left hand side and a list of the concepts on the right and side ........................................................................................................ 195

Figure 9.10: Screenshot of Mecureo online interface where students can upload their own dictionaries. ............................................................................................................................. 196

Figure 9.11: Screenshot of Mecureo online interface showing a variety of output formats that were available. .......................................................................................................................... 197
Figure 9.12: MyMuseum interface for a page on Hector (upper) and showing the adaptations (lower).

Figure 9.13: TARO architecture showing the addition of SKOS input and output to the ontology generation process and the visualisation in SIV.

Figure 9.14: Fragment of SAGO input serialized in SKOS for the concept Write permission.

Figure 9.15: Fragment of SAGO output for the input shown in Figure 9.14. Note the added relationships and the information about them.

Figure 9.16: TARO interface. SIV can be seen on the left, and the augmented definition with information about the relationships on the right.

Figure 10.1: Overview of the major system components and interactions between them.

Figure C.1: Screenshot of Task 0: Introduction.

Figure C.2: Screenshot of Task 1: Colour.

Figure C.3: Screenshot for Task 2: Font Size.

Figure C.4: Screenshot for Task 3: Searching.

Figure C.5: Screenshot for Task 4: Evidence.

Figure C.6: Screenshot for Task 5: Selecting a concept.

Figure C.7: Screenshot for Task 6: Comparing yourself to the average.

Figure C.8: Screenshot for Task 7: The class Average.

Figure C.9: Screenshot for Task 8: Inferring about knowledge.
List of Tables

Table 1.1: Summary of widely available evidence................................................................. 11
Table 1.2: Summary of problems and challenges................................................................. 16
Table 2.1: Summary of visualisation tools for user models.................................................. 42
Table 3.1: Evidence sources in the UIDP course................................................................. 63
Table 4.1: Comparison of VIUM to SIV................................................................................ 77
Table 4.2: Parser performance for a small (189 concepts) and large (1131 concepts) ontology file................................................................. 87
Table 5.1: Relationship types in Mecureo.............................................................................. 97
Table 5.2: Conversion of relationship strength and type in Mecureo.................................. 101
Table 5.3: Responses to question about the general approach.......................................... 110
Table 5.4: Responses to question about a more extended inference distance in the ontology........ 111
Table 5.5: Responses to question about a more restricted inference distance in the ontology........ 111
Table 5.6: User assessments of the overall idea of inference ........................................... 112
Table 5.7: User assessment of the principal of inference rather than explicit user input......... 113
Table 5.8: Added Concept Linkage...................................................................................... 115
Table 5.9: Distribution of concepts and peers...................................................................... 115
Table 6.1: Summary of participant familiarity with SIV and the UIDP course.................... 131
Table 6.2: Summary of Metasaur Evaluation Logs.............................................................. 133
Table 6.3: Concepts added by users for slide 4................................................................... 135
Table 6.4: Concepts added by users for slide 5................................................................... 135
Table 7.1: Understanding of audio slides based on duration stayed................................ 145
Table 7.2: The inferred values for each concept................................................................. 154
Table 7.3: The information written by participants on the signup sheet........................... 155
Table 8.1: Elements of evaluation related back to hypothesis........................................... 162
Table 8.2: Different groups of participants and their activities ........................................... 164
Table 8.3: Participant usage of UI-SIV during stuvac period, split by whether their level of use was above the threshold of 50 interface actions

Table 8.4: Summary of actions during stuvac

Table 8.5: Summary of task completion for the 37 students who attempted the tutorial

Table 8.6: Summary of participant interactions based on region in Figure 8.3.

Table 8.7: Average of actions required and used to complete tutorial tasks

Table 8.8: Qualitative responses when participants were asked to comment on concepts that the system thought they were performing poorly in

Table 8.9: Qualitative responses when participants asked for additional comments on SIV

Table 10.1: Comparison of the user modelling servers Personis and CUMULATE

Table B.1: Processed Log Data for GOMS Lecture (2003)

Table B.2: Processed Log Data for GOMS Lecture (2004)

Table B.3: Processed Log Data for GOMS Lecture (2005)

Table B.4: Processed Log Data for Cognitive Walkthrough Lecture (2003)

Table B.5: Processed Log Data for Cognitive Walkthrough Lecture (2004)

Table B.6: Processed Log Data for Cognitive Walkthrough Lecture (2005)

Table B.7: Summary of Averages (2003)

Table B.8: Summary of Averages (2004)

Table B.9: Summary of Averages (2005)

Table D.1: Summary of SIV Interaction Scores and Exam Marks

Table D.2: Summary of SIV Interaction Scores and Exam Marks (stuvac participants)

Table D.3: Summary of participant interactions during stuvac
Chapter 1

Introduction

This thesis explores the way light-weight ontologies can support scrutability for large user models and the user modelling process. That is, we are interested in approaches to create and interact with user models in a way that exploits light-weight ontologies for the purpose of scrutability.

We start with two motivational scenarios that demonstrate the uses of user models in authentic environments and illustrate why scrutability is an important aspect, and then we discuss why and how light-weight ontologies can be exploited to support user modelling with large user models.

1.1 Motivation

The first scenario describes a university course where much of the learning takes place online, supplemented with weekly face to face contact with instructors. In fact, the course described in this scenario is similar to the context in which we conducted our large scale evaluation, and forms an introduction to it. The second scenario describes a possible situation in a ubiquitous computing environment.

1.1.1 A Course in User Interfaces

This scenario involves a university course on user interface usability and design. The course is taught over one semester with learning material delivered through a website. Students are required to log into the website to access the learning content.

The course is organised into learning topics; each object teaches a number of low level concepts all relating to usability or design. The learning topics contain online lectures: visual (text and images) slides augmented with audio narration. The narration provides a much more in-depth explanation of the information on the slide than the content of the visual material. An example of a slide is shown in Figure 1.1. Essentially, this is similar to a typical use of slides as an adjunct to the verbal presentation of the lecturer in a lecture theatre. Each learning topic has between one and three online lectures.
Each learning topic also has an associated weekly face to face tutorial session that contains activities related to the online lectures. The nature of the website means that students can access any of the learning topics at any time, and can always review past learning objects if they need to. Keen students can also access learning topics ahead of the recommended week, moving ahead of the minimum rate of progress.

The course website gives us a perfect environment in which we can capture data about the students’ progress through the semester through simply observing the log data of accesses made to the web pages of the learning objects. Additional information about student progress comes through the marks accumulated through assessments each week. The assessment marks are graded by a tutor after students attend the tutorial session. Now let us consider the following scenario:

Suppose it is Week 6, half way through the semester. A student in the course, Jane, has unfortunately fallen behind in the tutorial sessions because she failed to attend the recommended learning objects for Week 5. She decides to examine her profile, accessible through the web site.

Her profile can be seen in Figure 1.2. It lists the learning topics in the course in a hierarchical structure and a summary of the concepts taught under each topic. Next to each concept is a score given as a percentage. This score is generated as a combination of the number of times a student has accessed the web page and the marks received from assessments (in this case, marks from the tutorial sessions). In turn, each topic also has a score associated with it aggregated from the scores of
the concepts under it as well as from lower level topics in the hierarchy. For example, the general topic of *Usability* has the subtopics *Empirical Usability* and *Predictive Usability*.

Jane notices that the topic *Predictive Usability* has a low score. Upon further investigation, she sees that it has three subtopics: *Cognitive Walkthrough*, *GOMS Analysis* and *Heuristic Evaluation*. Both *Cognitive Walkthrough* and *Heuristic Evaluation* have low scores, which in turn is causing the low score for *Predictive Usability*. Jane realises that these topics were online lectures she had skipped, the cause for the low score in her user model. She now attends the two online lectures.
When she checks her profile again, it appears as in Figure 1.3; she notices that her score for *Predictive Usability* has risen to because she has spent time viewing the learning objects.

Jane is now happy with the progress she has made so far in the semester. She now decides to try working ahead by attending next week’s recommended learning topics. There is a single topic recommended for next week: *Think Aloud*. She examines her user profile again and sees the topic *Think Aloud* as a sub-topic of *Empirical Usability*. She decides to access this learning topic, as she has recently attended the other topic, *Monitoring*.

This scenario has much in common with blended e-learning, featuring a conventional learner management system. The critical difference is that Jane’s course also has a learner model, and all parts of the blended course are linked to it. So, for example, her “attendance” at online lectures is reflected in the learner model.

Here, a scrutable, open user model provides a way for students to reflect upon their progress and learning during the course. The course described in this scenario will be referred to in the rest of the thesis, and will be described in more detail in Chapter 3.

### 1.1.2 Shopping for Cameras

Crazy Camera Bargains specialises in all sorts of cameras and photography equipment, in particular digital cameras. They provide an innovative service where people with PDAs or Smart Phones can export personal information to the shop via wireless technology and receive personalised recommendations for products. The following scenario describes a customer’s possible interactions with the system:

Bob is planning a trip to Japan and has decided he needs to purchase a digital camera. He is travelling as part of a business trip to attend a conference, so he will not have much time for sightseeing. Bob needs a light compact camera that he can quickly pull out and take photos of interesting sights as well as photos of colleagues at the conference.

Bob accesses his user information on his PDA and decides to share some details that the shop can use to give him personalised recommendations. Bob decides to share his dates of travel, dimensions of a preferred camera, as well as a price range among other things. Bob decides not to share certain personal details such as the destination or purpose of his travel, or his name and contact details.
The shop receives the information from Bob’s PDA as he enters the store and the shop system creates a set of recommendations. Bob is presented with a list of recommended cameras on his PDA as well as their locations in the shop, all within the size and price ranges Bob prefers. In addition, the store has inferred that he is leaving the country soon and also lists a duty free price for all the cameras, as well as some peripheral products he might consider purchasing such as camera bags, spare batteries, and an international power adapter for the battery charger.

Bob selects a camera he likes, as well as a camera bag and a spare battery. He decides not to purchase an adapter as he has one already. Bob goes back to his PDA and exports his contact details from his user information that is then used by the store to generate an invoice and the appropriate warranty information.

Once the transactions are complete, Bob sets all his recently shared information to private again and leaves the store with his new purchase.

This scenario illustrates the potential benefit if users have an easy way to manage their user models in a situation where inferences can be made about them from the information they share out.

1.2 Motivation

We will draw upon the scenarios above to illustrate our motivation for exploring how light-weight ontologies can support scrutability of large user models and user modelling.

The need for scrutability and user control

User adaptive systems employ collected and processed data about a user to perform adaptation. This data contributes to a user model, which is a knowledge source containing explicit assumptions about all relevant aspects of the user, that can be decoupled from the rest of the system (Kobsa and Wahlster 1989).

This thesis is concerned with making the user model scrutable (Kay 1999). It is important that users are able to inspect and understand the user model and have a sense of control over what is done with the information held within. The two scenarios described above illustrate this:

- Jane needs to be able to relate what she sees in her user profile to what is taught in the course. Jane realises that the concepts in the user profile are categorised under the course topics and knows that she can visit the relevant learning objects when there are concepts she does not understand.
• Bob keeps personal information on his PDA. This information includes details of his tasks (such as his trip details) as well as preferences (his specifications for a camera). Bob can inspect this information and control what is shared with the store.

The European Union Data Protection Directive (1995) defines personal data as information relating to a person who can be identified or is identifiable from this information “in particular by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity”. Article 12 also states the user’s right to obtain:

“communication in an intelligible form of the data undergoing processing and of any available information as to their source“

“as appropriate the rectification, erasure or blocking of data, the processing of which does not comply with the provisions of this Directive, in particular because of the incomplete or inaccurate nature of the data”

User models should be considered personal data (Shreck 2001), and under the EU directive, they should be accessible to the user. There is also growing recognition among commercial organisations and companies regarding privacy, user control and knowledge of their personal data. Prabhakar Raghavan, current head of Yahoo! Research¹ states (Farber 2005):

“Users should have control of what data is collected or given up and knowledge of what is done with it”

“Giving every person their clickstream doesn’t make a lot of sense —most don’t want it — but they should have knowledge and control.”

In 2007, Google Inc.² made a press release regarding their data retention policies, stating that beyond possible legal requirements for retention of user identifiable search logs, any data used in personalisation services would be at the control of the user (Fleischer and Wong 2007):

“In the future, it's possible that data retention laws will obligate us to retain logs for longer periods. Of course, you can always choose to have us retain this data for more personalized services like Search History. But that's up to you.”

These are compelling examples of the importance and value of user data. They also highlight the importance of users being able to understand their own data to enhance their own experiences, for example, the abovementioned personalised services provided by Google.

¹ http://research.yahoo.com/
Scrutability becomes especially important when user models are very large, consisting of hundreds or even thousands of components. Inference can be something that becomes very complex due to the number of components involved, creating considerable challenges to maintain the scrutability of the user model. There is a need to explore ways to structure and visualise user models to make the user modelling process in these situations more explicit and understandable.

**Ontologies to structure user models**

An ontology is defined as an explicit specification of a conceptualisation. It is a shared understanding between the system and users of the objects, concepts and their relationships (Gruber 1993).

There is a role for ontologies in providing structure to a user model. In cases where the user model does not have a suitable structure to organise the components in the model, an ontology should be extremely useful, both because it can provide such a graph structure and because that structure should make sense to the user, in terms of the meanings of the concepts modelled.

- Jane’s course has a hierarchical structure over the learning objects, which in turn helps categorise the concepts within them. The user profile that Jane views utilises the structure of the course to display the information in her profile.

- The information that Bob shares is used by the store to generate recommendations. There is a structure to the product catalogue in the store and that structure allows the system to find items related to Bob’s personal information.

This thesis proposes the use of *light-weight ontologies* to structure user models. In the scenarios we can see that both Jane’s user model and the product catalogue for the camera store are both hierarchical in nature, and can be considered light-weight ontologies. We are interested in light-weight ontologies because of their prevalence in many systems, and the ease at which they can be created.

**Ontologies and reasoning about users**

Ontologies also can be used for reasoning. By knowing a subset of facts within the domain, new facts can be produced by inference. We can use inference on user models with an ontological structure to find out new facts about the user.
User models typically consist of components, each a single piece of information or belief about the user. Even though the user model may be very large with hundreds or even thousands of components, it is important that the system maintains scrutability, particularly in reasoning about the data, whether it is by the user modelling system or other users who might have access to data from the user model.

Therefore, it would be extremely useful for users to be able to explore possible inferences that can be made about them by an agent from their partial user model. This is important because even though the partial user model may only contain a subset of data, many inferences can be made about the user based on an underlying ontology.

This is especially true in Bob’s case where he wants to keep certain details of his trip as well as personal contact information private. The store can still provide adequate recommendations even with partial information about Bob by making inferences. Figure 1.4 illustrates this. In this case, Bob has chosen only a subset of components from his user model (top right) to be shared out. The system allows him to inspect possible inferences that can be made about him from this information (bottom right).

Ontological inference is very important in environments where evidence feeds into the user model at different granularities. A typical example of this is in e-learning environments, where course materials typically teach fine-grained concepts that contribute to higher level learning goals. It is important to be able to model both coarse and fine grain concepts in the learner model, so that learners can see their overall learning progress and goals through the coarse grained concepts, and also
determine what elements of work contribute to these high level goals through the fine grained concepts (McCalla and Greer 1994).

In the scenario where a learning system was modelling Jane’s learning of usability and user interface design, the granularity of the learning material plays an important role in the distribution of evidence, as seen in the screenshot of her user profile in Figure 1.2. The more general concept *Usability* has no direct evidence sources (and thus no value associated with it), but the finer-grained concepts all have direct evidence sources and values based on learning evidence. Ontological inference plays a role in reasoning about Jane’s knowledge of the higher level concept *Usability*.

Figure 1.5 illustrates how Jane’s learner model is created by overlaying her usage of the course material to an ontology of the course subject matter, in this case the field of Usability. Her learner model contains some concepts that have accreted evidence directly from an evidence source (coloured in shades of pink) and others that have no evidence because they are coarse-grained (coloured white). Ontological inference allows the system to reason about Jane’s knowledge of these concepts; this is based on her knowledge of the concepts that have direct evidence (the white concepts have become pink on the middle right illustration of her learner model after ontological inference).

1.3 Identification of Key Problems

To support the scrutability of large user models and the user modelling process with an approach based on light-weight ontologies, such as in the scenarios involving Bob and Jane, there are a number
of challenges in different areas that must be addressed. A graphical representation of these challenges is shown in Figure 1.6.

The four boxed components are core to our approach to ontology-supported user modelling. The user (blue box at the top) interacts with a visualisation driven interface to a large user model, as described in the previous section. Problems that must be overcome are associated with each core component via the black lines. Our approaches to address these problems using other core components are shown with the blue lines.

We now define and describe each of these core components in Figure 1.6 along with their associated problems and challenges:

**Issues relating to the user model and user modelling**

The overall goal of this thesis is to address the *scrumtable user model interface problem*. We have highlighted this problem in pink in Figure 1.6. When users interact with their user model, there is a real challenge in making it easily understandable, explainable and accountable. We want to be able to provide an interface that allows users to access and explore their user model as well as the underlying processes for system beliefs. When we are dealing with a large amount of information coming from both the user model and the domain content, the importance of an effective interface to support scrutiny of the user model and user modeling process is paramount. We now briefly discuss

![Figure 1.6: Problems and goals associated with scrutable user modelling based on light-weight ontologies.](image-url)
underlying issues with user modelling that we must overcome in the process of providing such an interface.

One of the most common ways to represent a user model is an overlay model, where components in the user model are mapped to a domain model (VanLehn 1988). The user model is typically an unstructured bag of concepts, each having a value representing user knowledge. In this thesis, we are interested in creating overlay models where light-weight ontologies are used as domain models. This way, the ontology can be exploited for both scrutability of the user model and also reasoning about users.

We now examine the different evidence sources that are widely available in user modeling environments, in particular, learning environments. Our goal is to build scrutinable user models, so it is important that we carefully consider the characteristics of the different evidence sources available for reasoning about the user. Ideally, even after possibly complex processing of the evidence, a user should be able to understand how the evidence source contributes to their user model. We now refine this notion of evidence, identifying three important properties which play a critical role in user modeling. They are all particularly important because they interact with potential roles for ontologies. Table 1.1 shows a (non-exhaustive) list of evidence typically available in a blended e-learning environment. The table summarises some important properties of each of the evidence sources.

The granularity column indicates the grain size that is typical of the learner model evidence from that source. For example, in the scenario with Jane accessing online lectures, the evidence is fine grained because a single lecture slide, with its associated audio, typically teaches a very specific set of fine-grained concepts. Similarly, a single multiple choice question tends to be very fine grained (e.g. a question on single digit *addition*). However, a combined grade for several multiple choice questions on a variety of different concepts relating to a particular topic is coarse-grained (e.g. a multiple choice quiz on single digit arithmetic involving *addition* and *subtraction*). We can see that widely available sources of learner modelling evidence in blended e-learning contexts have a range of granularity.

Table 1.1: Summary of widely available evidence.

<table>
<thead>
<tr>
<th>Examples of sources</th>
<th>Granularity</th>
<th>Purity</th>
<th>Reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logs of online lecture “attendance”</td>
<td>Fine</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Final course mark</td>
<td>Coarse</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Multiple choice question on a single concept</td>
<td>Fine</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Grade for weekly class quiz involving several multiple choice questions</td>
<td>Coarse</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>Semester long group project</td>
<td>Coarse</td>
<td>Low</td>
<td>Medium</td>
</tr>
</tbody>
</table>
The second column shows the purity of the evidence, meaning whether the single piece of evidence includes different aspects. For example, a student answer to a sophisticated and complex task may require several skills and an incorrect answer may be caused by problems in any of these and so, the purity of the evidence is low and they are multi-faceted. An example of this from Table 1.1 is a semester long group project that asks students to design a website and build a prototype from a set of requirements. Here, the students must demonstrate knowledge of both screen design and requirements analysis concepts, as well as web programming skills. Marks from such a project would generate evidence across several different components in the user model that represent different conceptual aspects of the project.

There is also a relationship between purity and grain size, in that very fine grained evidence will tend to have high purity. It is easier to build reliable learner models from very pure evidence. In a system that can capture marks for individual multiple choice questions (which would be fine-grained and pure), a mapping can be made between each question to a concept in the user model. In contrast, a system that can only capture aggregated marks from a multiple choice quiz (which can be seen as coarse grained and impure) will have to map a single mark to multiple user model concepts that the quiz examined.

The final column indicates the reliability of each evidence source. For example, evidence based on the time spent listening to online lecture slides is of quite low reliability for many reasons. We certainly cannot tell whether the learner actually listened to the audio. Nor can we tell whether accesses were actually due to a student allowing another to use their account. All the sources of noise in this evidence source make it of low reliability. Similarly, the reliability of marks from a group project is not highly reliable unless the contributions from each member are honestly stated and accurate. By contrast, examination marks are generally highly reliable; with high likelihood they represent the right person and because the grading and recording of examination marks are done especially carefully.

Based on these issues, and also illustrated in Figure 1.6, there are a number of challenges related to user modelling that we need to overcome in this thesis:

- **Definition:** overlay models typically consist of a bag of concepts that represent components in the user model, with these mapping back to the domain model. We need to be able to choose a suitable set of concepts that are representative of the task, context and domain for which we are creating user models. We call this the user model definition problem. In our approach shown in Figure 1.6, we aim to draw upon the concepts used as metadata to define the user model.
• Maintenance: user models contain system beliefs about users generated from evidence collected about users, from different sources, in a domain or context. An important problem is the process taken to update the user model with new evidence. We want this process to be simple enough to be able to explain it easily to users, and also flexible enough to allow customisation to match user preferences or interaction styles. Importantly, we need to consider the issues of purity, reliability and granularity as discussed above. There are two important issues that arise relating to reliability and purity:

• The evidence normalisation problem: define how the available data from the different types of evidence sources contribute to concepts in the user model, addressing the issue of varying types and amounts of evidence from the evidence sources. An impure source of evidence will teach several concepts. Therefore, we must be able to feed evidence into all relevant concepts based on this type of evidence source.

• The evidence combination problem: we must deal with issues of varying the reliability and number of evidence sources when aggregating the evidence across different sources to determine a value in the user model. We must be able to distinguish between reliable and unreliable sources of evidence and combine them accordingly.

• Granularity: the granularity problem relates to the way evidence sources contribute data to the user model and the fact that there may be components in the user model that have little or no evidence, but we would still like to reason about them in a useful fashion. We call this the granularity problem.

Issues relating to ontologies

We categorise ontologies into two types: heavy-weight and light-weight. Heavy-weight ontologies tend to have formal groundings and axioms that can be verified using logic. A light-weight ontology is usually a hierarchy of concepts related by taxonomic is-a relationships. There may also be non-taxonomic relationships. The relationships are not formalised in logic such as the ones in heavy-weight ontologies.

This thesis is primarily concerned with the application of light-weight ontologies for scrutable user modeling. Thus, there are some additional requirements specific to this purpose. These are scrutability and the ability to support a reasoning layer specific to user evidence. There are a number of challenges as indicated in Figure 1.6 that must be addressed for the ontologies to be used in our approach:
• **Construction**: creating ontologies is challenging; we require a source of knowledge of the domain we are trying to model (either through experts or documentation); a process by which we can identify the concepts and relationships that make up the ontology, and a way to represent the ontology. For the purpose of this thesis, we are particularly concerned with scrutability, and need an ontology construction process that can facilitate this need. We call this the *ontology construction problem*.

• **Enhancement**: an ontology, as initially created may not contain all the concepts and relationships necessary for the task intended. It is imperative that ontologies be easily and quickly updatable; this, however, introduces the challenge of being able to enhance the ontology without compromising its integrity. We call this the *restricted ontology problem*.

• **Interface**: we require an effective interface for viewing and exploring the ontology as well as in the aforementioned construction and enhancement processes. We also explore how the interface can be applied to the other core areas of metadata annotation and user modelling. The blue line to this problem from the Visualisation component in Figure 1.6 indicates our goal to utilise visualisation techniques to contribute a solution to this challenge. We call this the *ontology interface problem*.

**Issues relating to metadata**

Metadata is information about information. It provides a structured, machine-readable description of content in unstructured or semi-structured information sources that is machine understandable (Swick 2000). This is usually achieved through the use of annotations to the content from a set of standardised terms (the metadata vocabulary).

As this thesis is concerned with the application of light-weight ontologies for user modelling in real domains, there is a need to support markup of metadata for the domain content. This needs to be done in such a way that the user modeling system can interpret the metadata and relate the content it describes to components in the user model. There are several challenges relating to metadata in this context, as shown in Figure 1.6:

• **Vocabulary**: structured metadata needs a common vocabulary of descriptive terms that can be used to describe domain content. The vocabulary must be verbose enough to describe the content in the domain adequately. We call this the *metadata vocabulary problem*. This problem has been widely recognised in the context of information retrieval (Furnas, Landauer et al. 1987).
Our approach, as seen by the blue line to this challenge in Figure 1.6, is through the use of ontologies, as they readily provide a vocabulary of metadata terms through their explicit specification of concepts that exist in a domain. Ontology enhancement techniques can provide added benefit in directly affecting and improving the terms in the vocabulary when the source ontology is improved with additional terms or clarifications.

- **Interface**: metadata annotation is a time-consuming and tedious task, especially when one has to annotate a large collection of documents. Providing a good interface can help make the task easier and faster. However, there are two challenging issues associated with this. Firstly, the metadata vocabulary can be very large and those people who need to define the metadata will need a way to retrieve relevant terms easily and quickly. Secondly, the content to be annotated may involve multimedia objects such as audio or video; in that case we cannot rely on traditional text processing techniques to identify keywords. We call this the *metadata annotation problem*. We believe visualisation of an ontology can aid users in the metadata annotation process, as seen by the blue line to this problem in Figure 1.6.

**Issues relating to Visualisation**

A visual interactive interface of information that aids and amplifies people’s cognition of data (Card, Mackinlay et al. 1999). For this thesis, the use of a visualisation and visualisation techniques underpin our approach to addressing the interface challenges with respect to the ontology, metadata and user model components, as indicated by the blue lines from the visualisation component in Figure 1.6. In particular, we are concerned with the use of ontology visualisation as a basis for creating scrutable interfaces. There is a problem relating to this that we must address:

- **Fan-out**: Even though ontologies lend themselves naturally to visualisation as a graph structure, there are issues when there is a large variance in the amount of connections into and out of each node in the graph. There is a dual challenge in being able to create ontologies that minimise this variance, and visualisations that are flexible enough to display ontologies that have a large variance in the number of connections for each node. We call this the *fan-out problem*.
Summary of problems and challenges

Table 1.2: Summary of problems and challenges.

<table>
<thead>
<tr>
<th>Scrutable user model interface problem</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User Modelling</strong></td>
</tr>
<tr>
<td>• The user model definition problem</td>
</tr>
<tr>
<td>• The evidence normalisation problem</td>
</tr>
<tr>
<td>• The evidence combination problem</td>
</tr>
<tr>
<td>• The granularity problem</td>
</tr>
<tr>
<td><strong>Ontology</strong></td>
</tr>
<tr>
<td>• The ontology construction problem</td>
</tr>
<tr>
<td>• The restricted ontology problem</td>
</tr>
<tr>
<td>• The ontology interface problem</td>
</tr>
<tr>
<td><strong>Metadata</strong></td>
</tr>
<tr>
<td>• The metadata vocabulary problem</td>
</tr>
<tr>
<td>• The metadata annotation problem</td>
</tr>
<tr>
<td><strong>Ontology and Visualisation</strong></td>
</tr>
<tr>
<td>• The fan-out problem</td>
</tr>
</tbody>
</table>

1.4 Light-weight Ontologies for Scrutable User Modelling (LOSUM)

Our approach and solution presented in this thesis is the *Light-weight Ontology-based Scrutable User Modelling* (LOSUM) toolkit. Together, this not only provides a complete system, but also a collection of tools that utilise light-weight ontologies for the purpose of scrutable user modelling. In particular, we apply the user model visualisation tool, *Scrutable Inference Viewer* (SIV), for a number of important roles in our approach, as suggested in Figure 1.6.

Figure 1.7 shows a broad overview of our approach featuring four core data components in our system in light blue. The directed arrows between these show the information flow. The actual application components utilised in LOSUM are shown in dark blue boxes on the outer edges. We describe these below and how LOSUM aims to address the problems indicated in Figure 1.6.

**User Model Visualisation**

An important goal of LOSUM is to be able to provide an interface for users to scrutinise user models that have large number of components in environments such as those described in the scenarios for Bob and Jane. We believe this is important, as user models are complex data structures that should be able to be explained or interpreted by users of the system who might not have a deep understanding of the modelling process.
In LOSUM we address the *scrumtable user model interface problem* with an interface called the **Scrumtable Inference Viewer** (SIV). We chose a visualisation approach to aid users in scrutinizing and understanding the information in these large user models. It provides scrutability by exploiting the scrutatable nature of an underlying light-weight ontology to both structure the user model components, and also as a way to perform ontological reasoning.

SIV is incorporated into both the ontology and metadata components of LOSUM which we describe below. It is this flexibility of SIV that makes it a core component of LOSUM and also our approach to addressing many of the aforementioned problems.

**Ontology**

The ontology plays a central role in LOSUM, and is shown on the bottom left of Figure 1.7; in particular, it gives an underlying structure to the user model visualisation and also the ability to perform inference on the user model. At the same time, it forms the basis for a metadata vocabulary. However, as shown in Figure 1.6, there are a number of issues and problems that deserve attention.

The first is the *ontology construction problem*. Although it would make sense to utilise a ready made ontology, such as a product hierarchy described in the Camera Shopping scenario, this may not be available for the domain being modelled. LOSUM addresses this by including a tool called **Mecureo** that can construct light-weight ontologies for user modelling from existing glossary and dictionary sources (Apted and Kay 2004). Mecureo has been designed to create scrutatable ontologies, in the sense the ontological reasoning can be explained by references back to the source dictionary.
The restricted ontology problem is another issue that must be addressed. We need a way to enhance ontologies constructed from glossary sources, to take into account terms used in definitions but not defined in the dictionary or glossary. In the User Interface course scenario, such terms might be the concepts that are core to the domain, but are at a level too fundamental to be included in a specialised glossary. An example of this might be the concept novice user, which might not be included in a specialised glossary on usability evaluation techniques. Such concepts are often fundamental to a teaching context, and thus it is crucial that they are identified and defined early on in the process.

LOSUM tackles the restricted ontology problem during the metadata annotation stage. An interface has been added to the metadata annotation system to allow users to define new metadata terms and definitions for them, thus treating them as part of a new user defined glossary. This user defined glossary is then processed with the original glossary or dictionary in the same fashion by passing it through Mecureo to generate a new ontology incorporating the new terms.

The fan-out problem is addressed in a similar way to the restricted ontology problem. There are two main problems here. The first is in the case of concepts that have little or no relationships to other concepts in the ontology. In this case, since the ontologies are generated through automated techniques, the most straightforward approach is to enhance the definitions for the concepts that have a low number of relationships to other concepts in the ontology. The second problem arises when concepts have a lot of relationships. LOSUM provides a solution on the interface level where relationships can be filter out on the display.

The ontology interface problem is addressed through the use of SIV as a way to visualise the ontology. It is important to provide an effective visualisation of the ontology, to easily identify any problems with the automatically-generated relationships early on. SIV provides both an efficient and broad overview of the ontology, and also allows for exploration and scrutiny of the concepts and relationships. Therefore SIV plays an important role in both the ontology creation process, and also in managing the fan-out issues described above.

Content and Metadata

We require the content to be annotated with metadata. Metadata annotation is a time-consuming and tedious task, especially when the content being annotated is multimedia.

The content and metadata components of the system are indicated on the top left boxes in Figure 1.7. We can see the outward arrows from these components to the user model: metadata allows systems to map user interests or knowledge from their user models to the content itself (User Model Definition arrow), thus providing a way to easily accrete evidence about user behaviour to specific components in the user model (Evidence arrow). The inwards arrow in Figure 1.7 shows that ontologies can play a
role in the definition of metadata because they not only provide a common understanding of the domain, but also a common vocabulary that can be used in the annotation process.

In the context of the thesis work, the User Interface Design and Programming (UIDP) course taught here at the University of Sydney provides learning content that require annotation. LOSUM addresses the metadata annotation problem through a novel interface that incorporates the SIV visualisation tool to allow users to explore the ontology in the annotation process. This tool is called Metasaur. Ontology visualisation during the annotation process can help user cognition in the discovery of metadata concepts that might be appropriate for the markup of a particular piece of content. Metasaur also incorporates the interface to add new concepts to the domain glossary and address the restricted ontology problem as mentioned above.

User Model

User models are stored in a user modelling system that provides a way for evidence to be accumulated and stored, and also a framework for the interpretation of this evidence to form system beliefs. In addition, the user modelling system provides user model data to client applications and tools. LOSUM incorporates the Personis user modelling server (Kay, Kummerfeld et al. 2002), with a set of custom resolvers to process the evidence. Personis stores user models, and provides functionality to accrete and resolve evidence that is stored with each model.

The user model maintenance problem involving the maintenance of user models based on usage of existing systems is a challenging task. We explore a process to construct individual and group user models from web log and assessment data. We want the whole process to be simple enough to explain it easily to users, and also easily customised to match their preferences or learning styles. This process takes into account the reliability, purity and granularity of the different evidence sources. There are two important issues that arise, with the approaches described implanted as resolvers in Personis:

- The evidence normalisation problem: we deal with impure and varying amounts of evidence from a single source by introducing a relative measure. Evidence pouring into the user model is compared and normalised against a standard user model. This ensures that data is comparable between different user models where there is a varying amount of evidence generated for each user.

- The evidence combination problem: we must deal with the issues of varying reliability and number of evidence sources when aggregating the evidence across different sources to determine a value in the user model. We introduce a scrutable weighting formula for each evidence source so they can be combined in a way that takes into account their reliability;
The granularity problem is addressed through the use of ontological reasoning to infer about components or concepts in the user model that do not have direct sources of evidence. An important issue is that reasoning in light-weight ontologies cannot use the same methods as formal ontologies: either the ontology must be repaired, or non-standard reasoning methods have to be applied (Huang, Harmelen et al. 2005). In LOSUM, we have developed a method for light-weight ontological inference called the onto-increment approach that takes into account ontological relationships with a particular emphasis on scrutability of the reasoning process.

1.5 Contributions

The contributions this thesis provides to the field of user modelling are:

- LOSUM, an implementation of a toolkit that utilises light-weight ontologies to support the user modelling process and in particular, large user models that comprise of hundreds of components. Chapters 3 to 7 describe the elements of LOSUM.

- SIV, a visualisation tool that shows how light-weight ontologies can be used to structure and visualise a large user model, addressing the scrutably user model interface problem and also the ontology interface problem. Chapter 3 describes the user view of SIV in the context of a learning domain; Chapter 4 discusses the implementation of SIV and the flexibility of its design to allow it to operate in a number of different roles.

- Metasaur, an interface for annotation of content, illustrating our approach to the metadata annotation problem through the inclusion of SIV as an ontology visualisation tool to aid users in discovering metadata terms, and also as an interface to address the restricted ontology problem in a practical way.

- The use of light-weight ontologies to address a number of problems encountered in the user modelling process and the scrutability of large user models, as shown in Figure 1.6, and demonstrated in the LOSUM toolkit mentioned above. These are:
  
  - the issue of common vocabularies for metadata terms and user model components in the user model definition problem;
  
  - the granularity problem in which we utilise the ontology to support reasoning across granularities in the user model;
  
  - the interface issues associated with the metadata annotation problem and the scrutably user model interface problem.
• Scrutable approaches to processing evidence based on comparison to a relative standard as a way to address the *evidence normalisation problem*, and a scrutable method to combine evidence across different evidence sources to address the *evidence combination problem* (Chapter 7).

• An evaluation of actual use and usability in an authentic learning context of a course with 114 students, of whom 77 had exposure to their learner models through a configuration of LOSUM called UI-SIV (Chapter 8).

• A demonstration of the flexibility and power of LOSUM as a toolkit for constructing user models from light-weight ontologies by its deployment in several applications (Chapter 9).
Chapter 2

Background

This thesis explores the ways that light-weight ontologies can support scrutable user modelling. This chapter presents a review of research relating to this thesis, and how they address the problems identified in Chapter 1. The review is divided into two main areas:

- Ontologies in user modelling: discussing light-weight ontologies and their role in helping define metadata and user models.

- Scrutable User Modelling: examine different user modelling systems, and then specifically user model visualisation tools.

The core components from the system are indicated in italics. Many of the systems and approaches described here address one or more of these core areas; we indicate how they relate to the work done in this thesis.

2.1 Ontologies and User Modelling

This thesis is concerned with the use of ontologies, in particular, light-weight ontologies in user modelling. We are interested in ways to create ontologies easily, and have them readily usable to both structure user models and also play a role in defining or describing domain content as well as reasoning about the user.

Although ontologies have their roots in the field of artificial intelligence, they have also attracted the attention from other fields such as knowledge management, information retrieval and electronic commerce (Fensel 2001). Ontologies have also been applied to the field of user modelling, which itself overlaps many of the aforementioned fields. In these systems, ontologies provide a crucial link between the domain content, user models and adaptation. For example, in information retrieval,
ontologies can aid in the provision of more personalised results based on individual interests and knowledge based on information in a user model (Luca and Nurnberger 2004).

This section begins with an examination of the role of ontologies in the Semantic Web, and the technologies to support them. We then examine methods to construct ontologies using ontology learning techniques, ontological reasoning and follow up with a discussion on the relationship between ontologies and metadata. During our discussion we will highlight relevant systems or applications that relate to user modelling, and in particular we focus on the use of light-weight ontologies. However, we will discuss heavier-weight ontologies as appropriate due to the ease at which they can be simplified into light-weight ontologies, such as the approach taken in (Tallon 2005) where relationship types in the heavy-weight ontology are simplified.

2.1.1 Ontologies and the Semantic Web

The Semantic Web (Berners-Lee, Hendler et al. 2001) is a vision that aims to imbue machine understandable meaning into content on the World Wide Web, where software agents can roam web content and perform useful tasks for humans. For this to happen, structured collections of information and sets of inference rules must be available for computers to use for automated reasoning. One effect of the Semantic Web vision is that it has promoted the use of ontologies and new standards to support automated reasoning and interoperability between web applications and agents.

The Semantic Web has implications for the work done in this thesis. Primarily, there are a number of standards and languages that have emerged that can support light-weight ontologies. Along with this are tools and applications that can potentially be exploited in our exploration of ways to utilise light-weight ontologies in scrutatable user modelling.

**OWL**

The Ontology Web Language, OWL (McGuinness and Harmelen 2003) is a recommendation put forward by the W3C as an ontology language for the web. OWL extends, and can be viewed as an application of, RDF and RDF Schema by adding a layer of formal semantics. RDF Schema allows us to express classes and properties, as well as hierarchical relationships. OWL increases the expressiveness by allowing things such as property constraints, equivalence, and quantifiers to be defined.

There are three versions of OWL that are of varying complexities:
- **OWL Lite** is designed to support tasks that require a classification hierarchy or taxonomy. It only supports a subset of the OWL language constructs, and is designed to allow for simpler tool support.

- **OWL DL** supports all the OWL language constructs but places certain restrictions on how they can be used. This is to guarantee computational *completeness* and *decidability*, meaning that all OWL DL conclusions are not only computable but also will finish in a finite time. Every legal OWL Lite ontology is also an OWL DL ontology.

- **OWL Full** allows for complete expression of the OWL language. There are no constraints on constructs (for example, classes can be instances of other classes). Unlike OWL Lite and DL, it does not guarantee computational completeness or decidability. Every legal OWL DL ontology is also an OWL Full ontology.

In practice, OWL Lite or OWL DL are usually employed based on the task and purpose, as implementations of reasoning support for OWL Full currently do not exist. For the purpose of this thesis, we are most interested in OWL Lite as a language to model light-weight ontologies. Although the language provides supports for constructs that go beyond that of simpler conceptual and semantic networks, the availability of editors such as Protégé (Noy, Sintek et al. 2001) and toolkits such as Jena (Carroll, Dickinson et al. 2003) make it appealing.

**Topic Maps**

The XML Topic Map (XTM) standard (Pepper and Moore 2001) provides another method for knowledge structures to be serialised and exchanged on the web, similar to both OWL and SKOS. It is a way of serializing the ISO/IEC Topic Map specification (Biezunski, Bryan et al. 1999) which comprises of three main concepts:

- **Topics**: the concepts or subjects of the domain being modelled.

- **Associations**: relationships between the topics in the domain.

- **Occurrences**: individual instances of a topic.

Topic Maps have been compared to a structure similar to an index in the back of a book: the index lists *topics*, and each topic lists page numbers which are *occurrences*. This metaphor gives an indication of the way Topic Maps can be used to organise and retrieve information, by finding the desired topics in the Topic Map and then examining their occurrences. The additional semantic information provided through *association* relationships mean that we can find related topics with ease as well.
Topic Maps are relevant to this thesis in two ways. First, they provide both a semantic overview of a domain, and are, for all intents and purposes, a light-weight domain ontology. Secondly, by being able to specify occurrences directly in the Topic Map structure, they provide metadata for the content they reference.

SKOS

The Simple Knowledge Organisation System (SKOS) is a standard for the specification of knowledge organisation systems such as thesauri and taxonomies (Miles and Brickley 2005). SKOS centers on the idea of having a collection of defined concepts (called a concept scheme) with relationships providing semantics to further define and link them. SKOS has four pairs of directed relationship types. These are:

- *BroaderGeneric* and *NarrowerGeneric*: subsumption relationships.
- *BroaderInstantive* and *NarrowerInstantive*: instance-of relationships.
- *BroaderPartitive* and *NarrowerPartitive*: hierarchical part-of relationships.
- *RelatedHasPart* and *RelatedPartOf*: associative part-of relationships.

SKOS has been used in several systems to represent domain ontologies, usually in conjunction with heavy-weight ontologies represented in OWL. Two examples of this use of SKOS are the Semantic Blogging (Cayzer 2004), and Semantic Web Portal (Reynolds, Shabajee et al. 2004) projects.

In the field of e-learning, SKOS has been employed in a student modelling architecture (Winter, Brooks et al. 2005) as the representation for course topic ontologies. SKOS is used to model course ontologies for different courses, and it was chosen over LOM (Hodgins 2002) which was better suited for capturing the relationships between material learning objects rather than pedagogical relationships between course units.

2.1.2 Ontologies to Structure Data

In this section we describe two different systems that utilise the aforementioned semantic web technologies to structure data. Both are directly relevant to the work done in this thesis: GUMO is an upper level ontology for user modelling – it can be used as a starting point for ontology-based user models. TM4l utilises topic maps as a way to explore learning content – here the ontology is used as a direct navigation tool to facilitate the discovery of learning resources for students.

---

3 Both part of the Semantic Web Advanced Development project at http://www.w3.org/2001/sw/Europe/
GUMO

An example of the application of OWL in user modelling is the General User Model Ontology, GUMO (Heckmann, Schwartz et al. 2005). It forms an upper level ontology for user models to facilitate exchange of user model data between different adaptive systems. GUMO focuses on typical dimensions about users that are modelled in adaptive systems, such as knowledge or beliefs. It leaves more general domain models to other ontologies such as SUMO.

GUMO is defined in OWL, and represents its statements about the user in the form of situational statements composed of two parts: main part information consisting of auxiliary, predicate, range and object attributes, and metadata information consisting of attributes such as creator, evidence and confidence. Each statement is designed to representing some dimension of the user model (Heckmann 2003). These statements are the foundation of the User Modeling Markup Language, UserML (Heckmann and Krueger 2003) for providing a common syntax for the serialisation and exchange of user models.

Recent work utilises GUMO to support ubiquitous user models in the SPECTER system (Kröner, Heckmann et al. 2006). SPECTER has relevance to the work done in this thesis not just through the use of an ontology-based user model, but also through its focus on scrutability. In addition, users are able to not only scrutinise the memories in their user model and fine tune it for a better personalisation experience in the future, but to also share them out. This is analogous to people recounting events they have experienced to family and friends.

TM4L

In the area of e-learning, the TM4L (Dicheva and Dichev 2006) system aims to use Topic Maps to provide more targeted access to learning resources for learners. There are two main components of TM4L: the editor component and the viewer component:

- The editor component emphasises reuse of learning content by providing facilities for authors to build or extend Topic Maps and relate them to the content they create. In addition, the TM4L Topic Maps link back to both the Dublin Core (DCMI Metadata Terms 2005) and LOM (Hodgins 2002) metadata standards.

- The viewer component is designed to allow users to not only access learning resources through the exploration of TM4L Topic Maps, but to also encourage exploration of the semantic structure between different topics. It does this by allowing different views of the Topic Maps: tree view, graph view and text view.
• The use of the Topic Map itself as an educational tool makes TM4L novel. It has been designed so that the different views show information that tries to capture the user’s intent for using that view. For example, the graph view aims to show an ontological overview of the topics and associations, and does not show occurrences of topics.

Recent work on TM4L proposed a notion of context as an extension to the Topic Map standard (Dicheva, Dichev et al. 2006). Users are able to specify contextual information or descriptors as part of their current task or goal to provide more relevant results to their queries. It also aids in the scrutability of the queries themselves. The current research on TM4L continues to investigate the use of contextual descriptors and ways to combine querying and browsing when searching digital collections.

2.1.3 Ontological Reasoning

One of the important issues we identified is the use of ontologies to aid in the reasoning about users, in particular, as a way to approach the granularity problem. Granularity of content is a prominent aspect of web based systems: the majority of content in such systems usually represents lower level concepts in the domain (De Bra and Calvi 1998).

For example, a website that teaches C programming will not usually have a page with metadata tag C Programming; instead there will be pages that describe lower level concepts such as pointer, argument, main function, etc. However, an instructor might still wish to infer about a learner’s overall knowledge of C Programming from the web log evidence of their page visits. Therefore the ontology must support the user model in being able to reason about higher level concepts where there is little or no direct evidence (i.e. the granularity problem). It also must support reasoning about concepts that are not explicitly defined in the metadata set. This is especially important in student modelling systems where higher level concepts represent core learning goals of the student, such as the C Programming course example.

There are many ways we can reason about the domain we have modelled using the ontology. The structure and representation method of the ontology itself plays a large role in the implementation performance, accuracy and gestalt acceptance of the reasoning. In research areas such as information retrieval, it is sufficient to use lexical ontologies to model users in order to expand on search queries (Luca and Nurnberger 2004).

This thesis is concerned with the reasoning strategies that would be appropriate for ontologies used in user modelling. In classic philosophy and artificial intelligence, there are two main methods of reasoning that can be employed, being deductive and inductive. We now briefly describe both types of inference and their use in user modelling.
**Deductive Inference**

Deduction or deductive arguments are defined as ones that, given a set of premises that are true, then the conclusion must also be true (Copi 1968). Using deductive reasoning in ontologies, new facts about the domain can be generated that must be conclusively true because the premises or statements that exist in the ontology are true. This relies on a representation of an ontology that has a high degree of formality, enabling deductive reasoning. It also requires a relatively complete model of the domain that includes higher level, fundamental concepts. Early designs in expert systems used representations such as predicate calculus and description logics that have well defined rules for stating propositions and making deduction.

OWL is a good example of a Description Logic ontology language. User modelling systems that make use of OWL also aim to make use of the reasoning support it provides.

**Inductive Inference**

In contrast to deduction, inductive logic does not provide an assertion of truth to the conclusion. It provides a way to deduce that the conclusion is probably true based on the evidence supporting the premises (Copi 1968). The inherently evidence-based reasoning of user modelling makes inductive logic a natural approach to make inferences about users.

Classical examples use Bayesian probabilities to illustrate inductive logic, and there are many existing systems that use this method to reason about users, for example, (Zapata-Rivera and Greer 2000; Conati, Gertner et al. 2002).

**2.1.4 Ontology Learning**

Ontologies are not simple to construct, this is widely recognised by many approaches and techniques that address the *ontology construction problem*. A common approach is to utilise domain experts or existing ontologies as a starting point. These approaches usually ensure that the resulting ontologies are well constructed ontologies that conform to formal theory. The IEEE Standard Upper Merged Ontology, SUMO, (Niles and Pease 2001) is an example of an ontology that people can use and extend as required⁴. On the opposite end of the spectrum are tools and approaches to construct ontologies automatically from existing sources. It is these types of ontologies that are of interest to us in this thesis.

The field of *ontology learning* aims to find techniques and solutions that allow us to construct ontologies automatically or semi-automatically from existing content utilizing machine learning and

---

⁴ Extensions and implementations of SUMO can be found at [http://www.ontologyportal.org/Pubs.html](http://www.ontologyportal.org/Pubs.html)
natural language analysis techniques (Perez and Mancho 2003). There are several reasons why this is desirable:

- Semantic web based applications require a way to construct domain ontologies cheaply so that the application can be built reasonably quickly (Maedche and Staab 2001).

- Existing content describing the domain is often freely or cheaply available. This content is usually either text-based, such as domain documents, glossaries or dictionaries, or from structured data such as a database or XML schema.

- Having access to an ontologist or domain experts is rare. This poses a serious problem for building or extending formal ontologies such as ones based on Cyc, as the domain modelling process becomes an expensive endeavour.

Four evaluation approaches to automatically constructed ontologies are presented in (Sabou 2005). These are: extraction performance by comparing the extracted terms to a manually crafted lexicon of the terms; evaluation by a domain expert or group; domain coverage by comparison to a manually constructed Gold Standard; and a task based evaluation to see how appropriate the ontology is for particular tasks. The last three approaches have the goal of evaluating the ontology in a qualitative manner. A broader overview of analogous approaches applied to ontologies in general is presented and discussed in (Brank, Grobelnik et al. 2005).

There are many systems that utilise automatically constructed ontologies and ontology learning techniques. The majority of these are based on the learning of ontologies from text such as domain documentation. A comprehensive survey of ontology learning methods and tools is available (Perez and Mancho 2003), though in this section we highlight a number of systems that have approaches amenable to this thesis.

**Energy Data Collection System**

The Energy Data Collection (EDC) system (Ambite, Arens et al. 2002; Hovy 2003) is an example of a system that utilises not only light-weight ontologies, but also ontology learning techniques. The EDC system aims to allow users to search across multiple energy-related data sources. This system accesses data from several different databases by utilising different domain ontologies that are mapped to an overarching ontology. The system is structured as follows:

- The overarching ontology used is SENSUS (Knight and Luk 1994), consisting of 70,000 nodes. SENSUS itself is an extension of WordNet (Miller, Beckwith et al. 1993). WordNet is
a semantic network that identifies lexicographical relationships between words, and is a common reference or starting point among many ontology learning approaches.

- The smaller domain ontologies are automatically generated from specialised glossaries by the GlossIt ontology learning system (Klavans, Davis et al. 2002).

- The documents in each domain are identified and extracted. These are mapped to the terms in the automatically constructed domain ontologies.

The system is novel in that it combines an existing upper level ontology with automatically constructed domain ontologies using light-weight ontological relationships. They are termed “generally associated with” (GAW) links, and are designed to be semantically vague. Although the GAW links cannot provide automated inference on the ontology, they do allow users to readily form queries to search across several domains using terms that are most familiar to them in their own domain of expertise or knowledge.

**ConceptNet**

ConceptNet (Liu and Singh 2004), part of the MIT OpenMinds\(^5\) project, is an example of an automatically created ontology that has been used in a number of different applications (Lieberman, Liu et al. 2005). Rather than concentrating on linguistic relationships such as those found in WordNet, ConceptNet provides relationships to represent commonsense knowledge about concepts.

ConceptNet has been created by parsing a set of common sense phrases (assertions) gathered from thousands of contributors via the Web. It consists of a connected graph of over 300,000 nodes, built from roughly 1.6 million assertions. There are 20 relationship types grouped into 8 thematic areas such as affective relationships (e.g. MotivationOf), functional relationships (e.g. UsedFor) and relationships for dealing with things (e.g. PartOf).

An extension to the OpenMinds project at MIT is StoryNet for capturing case-based common sense reasoning by computers. ComicKit (Williams, Barry et al. 2005) is one such tool that allow people to interactively build computer understandable stories via a comic book like interface with ConceptNet powering the available expressions, objects, and actions on the menus.

\(^5\) http://commonsense.media.mit.edu/
2.1.5 Ontologies and Metadata

There is a need to be able to mark up domain content in adaptive systems. Jeffery points out that metadata is of great utility in providing dynamic optimisation, flexibility and integration of distributed heterogeneous information, as well as providing benefit to the end-user cognition (Jeffery 1998).

The metadata vocabulary problem is concerned with the discovery of a suitable set of terms that can be used for metadata annotation. Ontologies provide one solution to the metadata vocabulary problem. There is a close link between ontologies and metadata. One of the goals of ontologies is to promote semantic interoperability between different systems as they form a common understanding of a domain or conceptualisation and also provide a controlled metadata vocabulary. That is, the available terms for describing content exist in a pre-defined set, or in this case, a pre-defined ontology or semantic structure. Metadata can also be part of the semantic structure through explicit relationships. For example, instances in OWL or occurrences in Topic Maps are ways to relate resources (such as a web page) to concepts in an ontology through an explicit relationship.

The work in this thesis utilises a vocabulary provided by the concepts in light-weight ontologies. We are concerned with ways to easily and quickly annotate content with metadata. This section explores how metadata is used and created.

The Relationship between Metadata and User Modelling

Metadata provides a mechanism for adaptive systems to relate the domain content to user models in an explicit manner. For example, an online bookstore has a page for each book, with metadata information such as author, publisher, type of book, subject matter and year of publication. The system can infer that a user might like a particular genre based on the subject matter of the books they have browsed, which provides evidence stored in their user model. The following are examples of user modelling systems that rely on metadata as a link between the content in the domain and the user model itself.

The AHAM reference model for adaptive hypermedia systems (De Bra, Aroyo et al. 2004) has three major components: the user model, the domain model, and the adaptation model. Each web page in the hypermedia system contains metadata keywords that map to concepts in the domain model and the user model. When a user requests a page, the system checks concepts related to the concepts that are associated with the page, as well as the value of the concepts in the user model. Adaptation is then performed based on rules that utilize this information, and the user model is possibly updated.

(Kearney, Anand et al. 2005) describe a system that aims to gain deeper insight into users’ behaviour and improve recommendations by including a domain ontology in conjunction with the visit history to
produce a better interest profile. The system takes advantage of the inherent metadata associated with content on an online movie retailer website to form mappings between content on unique URLs to the domain ontology, such as movie title, director and actor.

Recent work in museum tour guides have also utilized domain models as a basis for recommending exhibits, paths and tailored information to visitors. The system, Ec(h)o (Hatala and Wakkary 2005), utilises a domain ontology to form a mapping between the physical exhibits, the themes of an exhibition, delivered audio content and a user model featuring a set of interests based on the concepts in the ontology.

2.1.6 Metadata Annotation

The task of annotating existing documents and creating metadata is challenging and non-trivial - it is hard to be thorough and consistent. In addition, the task is both demanding and boring, especially in systems with many existing documents and a large metadata term vocabulary (Thornely 1999).

The task of classifying information on the web is unlike traditional information classification problems (Shirky 2005). This is primarily due to the large number of documents, the lack of centralized coordination, and a population of users that are mainly novice or casual.

In (Mathes 2004), three sources of metadata creation are described:

- Professionally created: where classification schemes are designed by dedicated cataloguers or domain experts. This is often expensive and timely to produce, and unwieldy for a fast changing environment such as the web where new content is created at a rapid pace. At the same time, the eventual users of the system are not involved in the metadata creation process.

- Author created: where the author of content is the one who defines the metadata for it. Although cheaper and more scalable to produce than professionally created metadata, it suffers from the same problem that the eventual users are not directly involved in the creation of the metadata.

- User created: the users of the system are the ones involved in defining the metadata for the content. This may be implicit (where the metadata is created in the background as users use the system, such as through stereotype user modeling approaches) or explicit (users actively take part to define the metadata terms through manual interaction).

An additional source of metadata creation is machine created metadata. Utilising techniques for document classification, it is feasible to automatically generate metadata annotations for documents. One problem with an automated method is the quality of the metadata. Human intervention or a semi-
automated process may be required to ensure the quality of annotations. This is similar to the drawbacks of ontology learning techniques for creating ontologies. At the same time, with the use of multimedia learning objects, automated approaches which are usually more suited to text become more difficult to employ with the existence of sound and visual content.

The most feasible approach for creating metadata, in the context of e-learning, is author created metadata coming from a controlled vocabulary. In a learning domain where users do not have sufficient mastery of the material, it is not appropriate to utilise user-created metadata.

Although our own approach utilises author created metadata, the remainder of this section provides a discussion of different metadata authoring systems that use various techniques as we can apply the research in this thesis to domains outside of e-learning.

**del.icio.us**

A recent trend, in many websites designed for users to organize or share information, has been the ability for users to add their own metadata tags to resources. Allowing users to categorise the content they access through tagging, and the ability to aggregate tags provides a pragmatic solution to the classification of web content. This metadata is not only designed to categorise their own resources, but also shared for others to see.

The social bookmarking site, del.icio.us⁶ is an example of an online service employing user created metadata utilising the tagging and sharing paradigm. We focus our discussion on del.icio.us in this section as it is typical of these sorts of online services. Other prominent sites that feature this paradigm are Flickr⁷ and You Tube⁸, for the sharing of photos and videos respectively.

In del.icio.us, each user has their own account to keep track of their bookmarks. Users can input their bookmarks directly on the website and provide corresponding tags and descriptions, or via browser extensions that allow them to bookmark and tag web pages as they are accessing them.

Bookmarks are shared; so for a particular bookmark, users can easily see how many other people have the same bookmark and what tags or descriptions they added. Users can also see the full bookmark list for other users, and the system also presents aggregated information such as the most popular or most recent tags.

The benefits of user-created metadata, in the form of tagging, are the immediate savings in time and cost, as the task is distributed. When a new piece of content (in the case of del.icio.us, a bookmark) is

---

⁶ http://del.icio.us/
⁷ http://www.flickr.com/
⁸ http://www.youtube.com/
added, the effort to add tags is minimised by the fact that the user can add free text. Other users can also tag the same object with their own tags, which may be different from the set of tags used by the original user who added the content. This leads to evolving metadata, which can adapt to changing user vocabularies, directly minimizing the cost of maintaining metadata in such systems.

The use of free text does lead to some drawbacks of the system that would not occur in a controlled metadata vocabulary. Firstly, there is a lack of acronym and abbreviation control. For example, someone who uses the acronym sums to mean scrutable user models to describe a document is very different to someone who tags a mathematical document about sums for summation.

Another problem is that without conventions for synonym or multi-word terms, we end up with a lot of redundancy in the vocabulary. For example, a page about user modelling could be tagged usermodelling or user_modelling by two different users. Similarly, the tags modelling and modeling could both be used depending on the spelling conventions used.

However, even considering these drawbacks, the ease of use and implications from social networking indicate that these systems have made significant inroads into solving the metadata vocabulary and metadata annotation problems in contexts where there is a large user base.

**Annotea**

The Annotea system (Kahan, Koivunen et al. 2001) is built upon semantic web standards and technologies to allow users to collaboratively add metadata to web content. Similar to de.licio.us, metadata is stored on a server, and can be easily shared between users. Users can add descriptions, and also reply to descriptions left by other users.

Annotea is novel in that users can control what metadata they want to see. This is because different communities of interest may have different metadata for the same content; the system allows users to choose the metadata most appropriate for their own interests or knowledge.

**SemTag**

The SemTag (Dill, Eiron et al. 2003) application does semantic annotation of documents, designed for large corpora (for example, existing documents on the Web). SemTag stores the semantic annotations on a server separate from the original document as it does not have permission to add annotations to those files. They have asked arbitrary users to check and approve large numbers of semantic links constructed as a means of evaluation.
AeroDAML

AeroDAML (Kogut and Holms 2001) is a tool that automatically marks up documents in the DAML+OIL ontology language. The amount of automation can be varied to suit the level of user interaction. Technical users are more likely to use a semi-automated approach to annotating the metadata, where non-technical users might prefer an automatic approach. AeroDAML uses the WordNet upper level noun hierarchy as the ontology.

LOMGen

LOMGen (Boley, Bhavsar et al. 2005) is designed for the semi-automatic generation of metadata for learning objects. It combines word frequency counts over the learning object with a lookup to an existing dictionary to suggest keyphrases to describe the learning object. The dictionary used is FOLDOC, the Free Online Dictionary of Computing (Howe 1998). A graphical interface allows users to interact with the metadata to select the most appropriate terms for the learning object after the automatic keyphrase generation step.

Figure 2.1 shows a screenshot of the LOMGen interface. Suggested keyphrases are shown on the left-

![Figure 2.1: Screenshot of the LOMGen user interface for adding metadata to learning objects.](image)
hand side. The user selects suitable keyphrases to describe the learning object by clicking the checkboxes. We can see the selected keyphrases OWL, semantic web, RDF, Internet, World-Wide Web, and resource description framework. Synonyms of the keyphrase can be entered on the right-hand side text input boxes. In this case, the user has entered Web Ontology Language as a synonym of the selected keyphrase OWL. Users can also optionally add terms manually via the text input on the bottom left. The bottom right drop down box enables users to select a top level category for the learning object.

LOMGen is similar to the approach we take in this thesis of using an existing dictionary or glossary to provide an initial metadata vocabulary for the learning objects. However, our approach involves the creation and use of a light-weight ontology first, with additional user defined metadata terms incorporated back into the source glossary and subsequently the ontology to enhance scrutability.9

2.2 Scrutable User Modelling

A scrutable user modeling system is one where users can not only examine the data in their user models, but also the processes that use that data for personalization. As argued by Kay (1999), there are a number of reasons why scrutability is desirable for user modelling systems:

- In the spirit of supporting privacy legislation for electronic mediums (Kobsa 2002), it is important that systems allow users to have an active role in the way their personal data is used. One way to achieve this is to let users directly interact with and inspect their user model, possibly correcting or removing data they feel is incorrect or inappropriate.

- A scrutable user model enforces programmer accountability in the assumptions that can be made about the user. It encourages more careful considerations in personalization choices made by the system, as well as the information collected about the user.

- It allows users to check the correctness and validation of their model. Because individual users change over time (for example, changing tastes in movie preferences), it is important to allow users the ability to correct any inconsistencies they see in their model. For example, in (Self 1991), a method is described for the system to diagnose student models, and allow for correction of the model. Modal logics are introduced as an approach to achieve this.

- Users can gain a better understanding of how the system behaves, thus enhancing the synergy between them. This is also related to the last point, where users who can directly see their user models have less reason to experiment with the system to determine what happens. Doing so could lead to incorrect data in their user model.

9 All images in this chapter are available in high resolution at http://www.cs.usyd.edu.au/~alum/thesis/02/
The idea of having open, scrutable user models of the learners in intelligent tutoring systems aid in reflection and provide additional educational benefits to both the instructor and the student (Bull and Pain 1995; Kay 2001). Because the ontology provides a foundational link between the user model and the domain itself, it aids the user’s comprehension of their user model and the domain itself.

2.2.1 Learner Modelling Systems

Our review of user modelling systems is focused in the area of learner modelling, as this is the domain in which we conduct our evaluations. The systems described in this section are all examples of learning systems that feature an open learner model\(^\text{10}\). That is, learners can examine and explore their learner models in a direct manner, providing the aforementioned benefits of scrutability.

Mr. Collins

Mr. Collins, for COLLaboratively maintained, INSpectable learner model (Bull, Brna et al. 1995), is a system designed for teaching languages and features an open learner model. In Mr. Collins, the learner model is presented through structured text.

Mr. Collins is designed to encourage learners to challenge their model, the aim being to be able to collaboratively construct and also correct the model and result in greater modelling accuracy.

A negotiation process between the system and the learner occur when there are discrepancies between their beliefs about the learner model. Learners communicate with the system through menu choices. Through this back and forth dialogue, the learner model is collaboratively constructed and changed.

---

\(^{10}\) Many more open learner modelling systems are featured at http://www.eee.bham.ac.uk/bull/lemore/examples.html
We can see an example of this in Figure 2.2, where the student is challenging the system’s belief about the learner’s knowledge of the rule positive main clauses. The student has asked the system to justify its belief that the student’s knowledge of positive main clause is ‘unsure’, and the system has responded with evidence from the student’s previous responses when answering questions related to this rule.

An evaluation of Mr. Collins was conducted with nine adult students learning Portuguese, all with no previous knowledge of the language. The study showed that students not only inspected and suggested changes, but also appreciated the fact that the system would challenge them when they disagreed with the beliefs in the learner model.

**STyLE-OLM**

STyLE-OLM (Dimitrova 2003) is an interactive open learner system that emphasizes diagnostic dialogue between itself and the learner. STyLE-OLM uses conceptual graphs to structure the learner model. One reason for this is the advantage of having an inherent and comprehensible visual representation of the learner model. This can be contrast with a system such as Mr. Collins (Bull, Brna et al. 1995) which features a purely text-based interaction method.

In a diagnostic process, the system and learner collaboratively construct conceptual graphs, with a structured text dialogue to enable the system to encourage learner involvement in the process. We can see a screenshot of this in Figure 2.3. As the conceptual graph is constructed, the learner is challenged with questions about the domain. In this case, the system is asking the user about the operation of capital_market.
STyLE-OLM was evaluated in a domain for learning technical terminology in a foreign language – the teaching of financial English to non-English students. The study showed that students could easily interact with the system. At first they focused more on the text dialog at the bottom of the interface, by increased their focus on the graphical representation once they were more familiar with the system. The study also showed that the open learner model aided in fostering reflection.

**SQL-Tutor**

The SQL-Tutor (Mitrovic and Martin 2002) is a constraint-based system (Olsson 1994) that aims to provide an open learner model to students learning about SQL. Students work through problems and write their own solutions. The system compares student solutions to ideal solutions, which are modelled as a set of constraints.

The actual open learner model for SQL-Tutor is very simple compared to systems like STyLE-OLM (Dimitrova 2003). Students are unable to directly modify the data in their learner model, instead having to perform problems. A screenshot of the learner model is shown in Figure 2.4. A graphical representation is employed in the form of skillometers, showing proportions of correct (in green) and incorrect (in red) understanding. A corresponding numerical total is shown along-side the skillometers. The system would also recommend possible next problems for the students to work on.

SQL-Tutor was evaluated with a study where students were divided into four groups: less and more able (based on a pre-test of their knowledge of SQL), with and without the use of the open learner model. The study showed that students could easily interact with the system. At first they focused more on the text dialog at the bottom of the interface, by increased their focus on the graphical representation once they were more familiar with the system. The study also showed that the open learner model aided in fostering reflection.

![SQL-Tutor](image)

**Figure 2.4:** Screenshot of SQL-Tutor, showing the learner knowledge as skillometers.
model. Findings from the study showed that less able students benefited more from the open learner model and performed better than those with comparable pre-test scores without the open learner model. For the more able students, they were also inclined to complete more questions and would challenge the system suggestion on possible next problems to work on.

2.2.2 Visualisation of User Models

To support scrutability and address the *scrutable user model interface problem*, visualizations may be employed to aid in the task, especially when the user model contains a large number of components. This section presents a general overview of information visualization, followed by a review of some existing user modeling systems supporting scrutability through visualization. Specifically, we are interested in user model visualisation tools that can be applied to different backend systems.

Graphical data representations have been used throughout history to help solve problems or improve cognition (Tufte 1983). (Card, Mackinlay et al. 1999), define the term visualisation as “the use of computer-supported, interactive, visual representations of data to amplify cognition”. *Information visualisation* is the application of visualisation to non-physical data such as document collections, financial data, and abstract conceptions. This presents new challenges when compared to *scientific visualisation* with its mathematical models and structures, which often has obvious spatial mappings between the data and the visualisation. Information visualisation is a large field with many aspects relevant to this thesis. In this section, we concentrate on the visualisation of user models.

The definition presented above helps define requirements of a visualisation of user models and ontologies. Firstly, it has to be computer-based, a trivial requirement in this case. Secondly it has to be interactive – users should be able to manipulate and explore the data. Lastly, it has to aid the user by amplifying cognition, that is, to help them in their thinking about the data and domain. Seven elements are presented by Schneiderman as desirable abilities of features of an information visualisation tool (Shneiderman 1996):

- **Overview**: Gain a broad overview of the data such as with the use of different zoom factors.
- **Zoom**: Be able to zoom or focus in on items of interest, while being able to maintain orientation in the data.
- **Filter**: Accentuate items of interest; filter out uninteresting items through dynamic queries.
- **Details-on-demand**: Get details about an individual item or group of items, especially after zoom and filtering.
Table 2.1: Summary of visualisation tools for user models.

<table>
<thead>
<tr>
<th></th>
<th>VIUM</th>
<th>QV</th>
<th>VisMod</th>
<th>PeerGlass</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Overview</strong></td>
<td>Entire UM shown.</td>
<td>Entire UM shown but collapsed.</td>
<td>Partial UM shown.</td>
<td>Entire UM shown.</td>
</tr>
<tr>
<td><strong>Zoom</strong></td>
<td>Can focus and on individual concept.</td>
<td>Can’t zoom.</td>
<td>Nodes, levels or labels can be hidden.</td>
<td>Can’t zoom.</td>
</tr>
<tr>
<td><strong>Filter</strong></td>
<td>Non-relevant items hidden via font size. Can search.</td>
<td>Can collapse nodes.</td>
<td>Choose which sub-trees to display.</td>
<td>Can filter by attribute.</td>
</tr>
<tr>
<td><strong>Details-on-demand</strong></td>
<td>Yes, shown on right hand side contents.</td>
<td>Yes, shown in node labels.</td>
<td>Yes, shown in new window.</td>
<td>Yes, can view the article.</td>
</tr>
<tr>
<td><strong>Relate</strong></td>
<td>Yes.</td>
<td>Yes.</td>
<td>Yes.</td>
<td>Yes.</td>
</tr>
<tr>
<td><strong>History</strong></td>
<td>Yes, back button provided.</td>
<td>No.</td>
<td>Event history can be saved or loaded.</td>
<td>No.</td>
</tr>
<tr>
<td><strong>Extract</strong></td>
<td>No.</td>
<td>No.</td>
<td>No.</td>
<td>No.</td>
</tr>
</tbody>
</table>

- **Relate**: See relationships among items, and see related items, possibly in conjunction with details-on-demand.

- **History**: A history of actions is preserved to enhance usability with backtracking, replay and progressive refinement.

- **Extract**: Be able to extract items or queries of interest to the user. A further step would be to allow users to export and save the data outside of the visualisation.

The user model ontology is a network structure in terms of Shneiderman’s data types; however, the user is primarily interested in the relationship links from a particular node (in other words, a tree with a focus node as the root); so it would make sense to apply guidelines from the tree data type.

It is important that visualisation tools be empirically evaluated, so that the strengths and weaknesses are identified and improvements made. An overview of empirical evaluation techniques for visualisation is provided in (Chen and Yu 2000). Two important measures stand out – accuracy measures (such as the number of correct answers, precision and error rate) and efficiency measures (for example average task completion time).
We now describe visualisation tools for user models. These are summarised in Table 2.1 with respect to Schneiderman’s visualisation tasks.

**VIUM: Visualisation of Large User Models**

Visualisation of Large User Models (VIUM) is a program written to display large user models in web-based systems (Uther 2001). It was inspired primarily by the Automatist Storytelling System (Murtagh 1996), a narrative engine that produced dynamic visualisations from keyword-annotated content. In a similar vein, VIUM produces a dynamic visualisation of a user model that contains hundreds of components, with each component being scrutable by linking back to the domain content.

VIUM is implemented as a Java Applet, which occupies a frame on one side of the browser window. It utilises perspective distortion to allow users to navigate the user model. Users can see the whole user model at once, getting a big picture of the structure of the model through the use of font size, spacing and brightness to see components that are related to each other in the underlying graph.

VIUM also facilitates the viewing of multiple datasets. The visualisation enables a user to see their user model in absolute terms or in terms of another standard. It also supports visualisation of the user model in comparative terms. For example, in the learning domain, students can see how their user model compares to that of the teacher’s expectations or the whole class.

Two classes of user models were examined during development:

- **Movie Preferences**: each concept is a movie, and these are related by actors and directors. The data was mined from the Internet Movie Database (IMDB)\(^{11}\). Scores were based on user ratings, and reliability on the number of people who have rated the movie.

- **The University of Sydney Graduate Medical Program\(^{12}\)**: concepts are all the topics that students should have mastered when they finish their degree. Scores are based on online assessment marks, and reliability is based on percentage of marks answered correctly in a learning topic.

VIUM user model components have a score and certainty. The score represents a heuristic value for the domain. The confidence value shows the system’s confidence that the score is accurate. VIUM uses colour to provide an indication of a component’s score through a slider that adjusts the viewing standard. Any components with a score less than the standard will have a red hue. Components with scores greater than the standard will have a green hue. The further a score is away from the standard, the more saturated the colour will become. In Figure 2.5, we can see that *Lawrence of Arabia* is

---

\(^{11}\) http://www.imdb.com/

\(^{12}\) http://www.gmp.usyd.edu.au/
currently selected; other movies related to it include *The Salamander, Land of the Pharaohs, and The World in His Arms.*

Clicking on a component with the mouse will select it and put it into focus, and the display will change so that the selected component will then have the largest font. Other components will have increased sizes the more relevant they are to the selected one. Components that are not relevant are dimmed and shown in a small font.

At this stage, VIUM is the only mature work that provides a visualisation of user models that contain a large number of components. Some other visualisations that are of interest to this thesis are briefly discussed below.

**PeerGlass**

PeerGlass (Klijger 1995) is a visualization of user models to support user trust in personalized services. In this case the service is the delivery of articles in the form of an electronic newspaper. The visualization system in PeerGlass has of two components:
A Totem pole shows which feature of a particular news article make it appropriate for the user. There are four attributes for each article that are displayed on four sides of the totem pole: source (where the article was from), topic (general interest area, such as sport), type (what the article is about, such as press release) and nature (tone of the article and subject matter). Each attribute has its own associated colour. Multiple totem poles, representing different articles the system thinks the user is interested in, are presented to the user at the same time.

Model planes show an overview of the system beliefs about the user’s interests on a more general level. The system beliefs are reinforced as users read more articles featuring the user’s interests. Users can also explicitly state their interests. At the time of evaluation, PeerGlass had four planes implemented:

- Explicit Interests.
- Interests by observation.
- Sources [of the articles] preferred.
- Communities or grouping of interests the user prefers.

On the *Interests by observation* and the *Communities* plane, font size (representing closeness or depth) is used to represent a user’s interest on the attribute’s score. All the planes retain the same colours associated with the attribute types on the totem pole.

When the user is viewing the newspaper, only the *totem pole* for the current article is shown. When the user is viewing their user model, then the *model plans* are shown in conjunction with the set of *totem poles* for the current articles.

PeerGlass was evaluated by graduate students from the MIT Media Lab and a number of alternatives or enhancements to the visualisation were proposed as a result of the study\(^\text{13}\).

**QV**

The QV (for quick overview) tool is an overview interface for user model in the um user modelling toolkit (Kay 1999). It was evaluated as part of a system to aid students learning how to use the sam text editor (Pike 1987). Learner models were constructed through monitoring of student usage of sam over a period of 18 months, and further evidence was captured through student interaction with their models. Results from the evaluation showed that for some students, interactions with the model helped in their learning of the sam text editor.

A screenshot of QV showing a learner model of knowledge about the sam text editor is shown in Figure 2.7. The domain model itself may be a directed graph, but through repeating nodes, the domain can be easily displayed as a tree hierarchy. The root node lies on the left of the screen, and subsequent nodes can be expanded or collapsed. Branches that are not relevant to the user’s current level of knowledge are initially collapsed to reduce cognitive load when they use the interface to scrutinise their user model. In this case, we can see the hierarchy containing concepts such *sam* itself, other editors (*emacs*, *vi*) and programming languages. The leaf nodes on the upper half of the display show different commands in sam, such as *quit_k*, *undo_k*, and *load_new_k* (the “k” designating a keyboard command).

Different shapes are used to represent beliefs (diamonds), knowledge (squares), non-leaf nodes (circle) and other components (crosses). The fill colours of the shapes represent component values. For knowledge components, the filled shape is true, and unfilled shape is false. For beliefs, the opposite is

\(^{13}\) Proposed interfaces can be found at http://xenia.media.mit.edu/~kliger/poster.html
true, as beliefs model (possibly) sub-optimal knowledge or misconceptions. This results in a user with perfect knowledge having all components coloured black.

**ViSMod**

ViSMod is a visualization to help students (and teachers) understand the way they are modeled in a teaching system (Zapata-Rivera and Greer 2000). The user model is represented as a Bayesian belief network (BBN), and ViSMod builds upon the earlier work of another tool, VisNet (Zapata-Rivera, Neufeld et al. 1999), for explaining Bayesian belief networks in a straightforward and intuitive manner.

In VisNet, the Bayesian belief network is displayed along with the size, proximity and colour of nodes representing strength of relationship, marginal probability and probability propagation. Animation provides a way to dynamically show the probability propagation in the system. ViSMod allows
learners to select nodes of interest on the BBN to visualize. Each concept has a score representing the system’s belief of the student knowledge of that concept. The size and colour of the node represent the user’s knowledge in a particular concept. In ViSM, proximity was not used.

A screenshot can be seen in Figure 2.8. We can see a portion of a BBN in the domain of Java programming. At the bottom is a toolbar that presents a number of features to students, including the ability to load a BBN file representing an initial student model structure, modify the visualisation techniques, change the animation type, query the nodes and view evidence, as well as moving the visualisation to see portions of the model not shown on the screen.

Additional interfaces allow users to explicitly set their knowledge levels for concepts if they disagree with the one displayed in the system. Students are prompted to provide an explanation of why they think the concept value is wrong or different.

### 2.2.3 Visualisation of Groups

Although the emphasis of this thesis is primarily to construct and provide a way to visualise individual user models, the ability to provide a visualisation of a group of users is also of importance.
In many situations, individual users are members of environments that rely on interaction or awareness of other individuals. From the previously described tools, only VIUM allowed individuals to access a group visualisation and provide a comparison view of their own user model components compared to those of their peers.

At the current time, there has been no work to our knowledge that specifically addresses the visualising of group user models; that is, tools that go beyond the visualisation of raw user evidence by providing higher level system beliefs about groups of users. Therefore, in this section we concentrate on some of the current tools available that provide group visualisations.

We define a group visualisation as one that allows a user to see themselves in the context of other users in the environment they are participating in (quite often this is an interactive social space, such as an online discussion forum). The freedom to allow users to access a group visualisation can aid users in the scrutability of their own user models, as they gain insight into their own actions and interactions by comparing it to the group.

**CourseVis**

CourseVis (Mazza and Dimitrova 2005) is designed for instructors rather than students to track performance and mastery of course material for web-based distance learning. It is relevant to this thesis as it deals with large amounts of web log data, which it aims to present through novel visualisations that contribute to the interpretability of the data.

Although not directly designed for student usage, CourseVis nonetheless comprises of an interesting set of visualisations that not only allow instructors to evaluate the progress of the learners, but to also allow them to reflect on the way their course is conducted. CourseVis also addresses similar issues of evidence processing and the need for a metadata vocabulary and the annotation of learning objects as we do in this thesis. We discuss these below.

There are three visualisations that are core to CourseVis:

- **Discussion Graph**: Provides a visualisation of the social aspects of the learners shown as a three dimensional plot as users participate in a discussion board. The axes show the *date*, the *originator* of a thread, and the *topic* the thread is discussing. Each thread is mapped onto the plot as a sphere. Size and colour of the spheres represent the number of follow-up replies to the thread.

  Figure 2.9 shows a screenshot of a discussion graph from two different angles. Looking at the label on the vertical axis, we can see the top graph shows users, with the threads they made at different points during the course mapped to the date axis on the horizontal. The bottom graph
50 shows the same data rotated, so now we have the topics on the right hand side with the same threads now seen from this angle so the instructor can map them across to corresponding topics.

- **Cognitive Matrix**: This visualisation shows the cognitive aspects of the learners through a mapping between the concepts in the course to the mastery of the concepts by the students. Instructors can easily see an overall view of the performance of the students, and identify concepts they are having problems with.

Figure 2.10 shows a screenshot of a cognitive matrix for a course. We can see a list of concepts along the vertical axis, and a list of students along the horizontal axis. The colour of a cell on the grid represents the mastery level of a concept by a student.
• **Behaviour Graph**: Unlike the discussion graph and the cognitive matrix, the behaviour graph shows a model for an individual learner rather than a group. Single axis composition is used to show multiple evidence sources (such as hits to the website, interaction to the discussion board, and assessment submissions) on a two dimensional plot against the dates of the course.

Figure 2.11 shows a behaviour graph with data about an individual student. Different representations are used depending on the evidence source. For example, we can see that the list of accessed course concepts on the top half of the graph are represented by colour squares like a scatter plot, and the aggregated hits to the website are shown under it as a blue bar chart.

Domain concepts are utilised in all of the above visualisations. CourseVis requires instructors to define suitable concepts to describe the course via a separate interface. These concepts need to be associated with the learning material manually. One of the goals of CourseVis was to be easy to use for course instructors, so the choice of using a straight list of concepts to describe the domain was
more appropriate than a process that would require knowledge of ontologies and ontological engineering.

CourseVis had a three stage evaluation with real course instructors (Mazza 2004). The stages of evaluation consisted of an initial focus group, an empirical study with tasks that compared CourseVis against WebCT\textsuperscript{14}, and follow-up interviews with participants from the empirical study. The overall consensus was that CourseVis provided a much faster and easier way for instructors to interpret data about the learners and instructors considered it a useful tool in the teaching of distance courses.

**PeopleGarden**

PeopleGarden (Xiong and Donath 1999) is a visualisation tool that aims to capture interactions between large groups of users in online environments such as forums, message boards and chat rooms. The metaphor of flowers blooming in a garden is used to represent participation in the social space. The visualisation is composed of two core ideas:

\textsuperscript{14} http://www.webct.com/
- **A PeopleFlower**: Each user is modelled as a flower. The height of the flower indicates the length of time the user has been a member of the social space; each petal represents an interaction with its saturation indicating the age of the interaction. More recent interaction petals have more saturation and brilliance than older ones. The colours of the petals can also be used to represent specific types of interactions.

- **The PeopleGarden**: A collection of PeopleFlowers, with the idea that a healthy social space with a lot of interaction and participants will naturally have numerous flowers of different heights, each having many vibrant petals.
The visualisation is effective with the use of a simple and intuitive metaphor that not only allows users to see their own interaction history, but also the holistic view of the space they are participating in. It is also novel in that the group model visualisation simply consists of multiple individual user model visualisations (i.e. the PeopleFlowers).

Figure 2.12 shows a PeopleGarden with many different PeopleFlowers, representing the interactions of participants in a message board. Here, magenta petals represent an initial post, whereas blue petals represent a response. There are several users with flowers consisting of many petals, indicating a high level of interaction in the social space. There are also PeopleFlowers at different heights, showing that the message board has had a consistent number of new participants joining the space since it started.

2.3 Summary

This chapter has provided background necessary to the thesis by introducing light-weight ontologies to support scrutable user modelling and large user models.

We have examined the ways ontologies in general can be utilised in the user modelling process as ways to reason about users, map the user model to the domain content, and provide a vocabulary for metadata. Existing systems and approaches have been discussed, and we have compared and contrasted these to our own approach.

There are a number of systems that emphasise open learner modelling and the visualisation of user models to enhance aspects of system scrutability. These systems highlight the importance of the scrutable user model interface problem and the benefits such interfaces provide in allowing users to directly interact with the user model.

We discuss the details of LOSUM and our own approach to dealing with the issues associated with the use of light-weight ontologies for scrutable user modelling in the following chapters and build upon the research and insights from this background review.
Chapter 3

Theoretical Framework and Approach

This thesis explores the way light-weight ontologies can support scrutable user modelling. In Chapter 1 we described this problem and motivation for this thesis. In Chapter 2 we described the background and related systems. This chapter is, in essence, an overview to our solution, drawing upon the previous two chapters and unifying them, to introduce the theoretical framework of our approach where light-weight ontologies play a role in our solution of the scrutable user interface problem. We give an overview of the work in terms of the end system and the context in which the development and evaluation have taken place. This is important as it provides a framework for understanding the rest of the thesis.

A critical requirement for practical exploration of the approaches of this thesis was for data that was representative of a realistic and meaningful source of evidence for user modelling. Our choice was to conduct our research in the context of a blended e-learning environment that would provide us with a range of both automatically generated evidence from website access logs and also human generated evidence from tutorial activities marked by course tutors. This context is very important because it is representative of existing and emerging normalised use of e-learning and learning management systems.

The first part of this chapter describes the theoretical overview of our framework in the context of existing systems and literature, and draws on the motivation provided in Chapter 1. We revisit the framework shown in Figure 3.1 and discuss the reasoning behind each element in the system.

The second part of this chapter describes this learning environment and discusses the evidence sources we consider for our user modelling. In Figure 3.1, this corresponds to the top-left dark blue component, the UIDP Website, which provides the content and also the user modelling system with evidence based on user interactions and usage, as shown in Figure 3.1.
This third part of this chapter gives a user view of LOSUM, the complete system we have created, integrating all the aspects shown in Figure 3.1. The core of the user view is the user model visualisation tool, Scrutable Inference Viewer (SIV). This corresponds to the bottom-right dark blue component, SIV, which produces a visualisation for displaying the user model data, and uses the ontology for both structure and also inference as shown in Figure 3.1.

### 3.1 Theoretical Overview of LOSUM

LOSUM has four main parts as shown in Figure 3.1, the Ontology, the Content Delivery System, the User Modelling System and the User Model Visualisation. The separation and distinction of these components is proposed in the Adaptive Hypermedia Application Model, AHAM (De Bra, Houben et al. 1999), which emphasises separate domain models and user models.

In LOSUM, we depart from traditional approaches through the use of a light-weight ontology that is automatically constructed from existing sources as the domain model (and in turn, a structure for the user model). In this section, we draw together the previous two chapters and discuss the theoretical reasons for why we chose to use light-weight, automatically constructed ontologies as the basis for the domain model, and as a structure for the user model. We follow with a discussion of the use of open learner modelling, and the way it has influenced the design decisions of LOSUM.
3.1.1 Automatically Constructed, Light-weight Ontologies

The light-weight ontology is a critical component that underpins the conceptualisation of the domain, the metadata, and the user model. The approaches of using ontologies that are both light-weight and also automatically constructed depart from traditional domain modelling approaches in adaptive systems. In this section we discuss the theoretical motivation for why we chose such an approach.

Light-weight Ontology

The use of an ontological structure to describe data in user modelling and adaptive systems has been wide-spread. A few examples described in the previous chapter include: learning systems, such as STyLE-OLM (Dimitrova 2003) and TM4L (Dicheva and Dichev 2006); movie recommendations (Kearney, Anand et al. 2005); museum tour guide (Hatala and Wakkary 2005). These systems utilise ontologies because of the ability to reason about the domain and the user. In LOSUM, we want to harness the power of ontologies for the same reason.

A departure from these systems in this thesis is the use of light-weight ontologies. In Chapter 1, we defined a light-weight ontology as a hierarchy of concepts related by both taxonomic and non-taxonomic relationships that are not formalised by logic. We chose this route for a number of reasons:

- Formalised ontologies are expensive to construct, especially in terms of both time and effort.
- There are light-weight ontological structures present in existing data sources that can be harnessed.
- The simple nature of the relationships means the ontological structure is not grounded by a particular representation or serialisation.
- We can use fully automated techniques to construct light-weight ontologies which would not be possible if trying to ensure logical consistency or constraints between concepts.

In the context of a teaching environment, where resources to construct ontologies are limited, the choice to utilise a light-weight approach over a heavy-weight approach provides benefits to instructors who can spend time focusing on creating learning content instead. A light-weight ontology constructed from a scruutable source would ideally have little cognitively demand in understanding how concepts are related.

The reuse of an existing scruutable ontology would provide an ideal starting point for LOSUM, as it would immediately overcomes ontology construction problem. Although existing ontological structures such as WordNet (Miller, Beckwith et al. 1993) are available, they are many domains where it is in-appropriate. The primary reason is that often domains may contain many specialised terms that would not be found in sources such as WordNet. An example of this would be a term such
as “Cognitive Walkthrough” for the domain of user interface design, in which the blended e-learning environment that we use for this thesis is based.

Automatically Constructed Ontology

We chose the route of using our own automatically constructed ontology. The field of ontology learning is a practical approach to creating ontologies from existing data sources such as documentation, SQL schemas, databases and dictionary sources, with a number of applications and tools utilizing this method (Perez and Mancho 2003). We chose to use specialised domain glossaries as the source for the ontologies we would construct. This would not only provide explicit concepts (i.e. each term defined in the glossary), but also contain a scrutable source (the definition itself) that can justify the relationship between links. It is also as a source we could easily enhance, addressing the restricted ontology problem – adding new concepts to the ontology would be as simple as creating a new term and definition in the glossary.

There are many approaches that utilise dictionary or thesaurus as sources for automatically constructed ontologies, such as GlossIt (Klavans, Davis et al. 2002) and ArcRank (Jannink and Wiederhold 1999). In LOSUM, we chose the tool Mecureo (Apted and Kay 2004) as a starting point for ontology learning for a number of reasons: it had been designed with scrutability in mind by providing a simple way to query the ontology and extract sub-ontologies; it was readily extensible to interface with other components in the framework; its design was largely consistent with the ontology learning framework proposed by Maedche and Staab (2001) - Chapter 5 describes the process we undertook to ensure that Mecureo fitted fully within the ontology learning framework.

The major issue we need to overcome, and one that is common in such ontologies, is that we need to adopt non-standard reasoning techniques (Huang, Harmelen et al. 2005). In LOSUM, we chose a technique that is based on spreading-activation (Salton and Buckley 1988). This differs from traditional approaches such as those described by Jameson (1996) - the core goal of our technique centers around scrutability and user control. We describe our technique in Chapter 7.

3.1.2 Open Learner Modelling

The use of Semantic Web technologies in educational hypermedia has received considerable interest and attention. Cristea (2004) introduces a number of situations where Semantic Web technologies can benefit such systems. In particular, when students are of different abilities or learning styles, the ability to adapt content to individual student needs is important. Being able to control or understand the adaptations is also important, as the core goal for this thesis is to address the scrutable user model interface problem. Open learner modelling provided an ideal area in which we could evaluate the learner models created with LOSUM.
LOSUM departs the AHAM reference model by creating a system that fostered user models, but does not provide any adaptation (though utilising the models as a basis for adaptation is entirely possible and an avenue for future work). Instead, LOSUM has been developed in the context of a blended e-learning environment that was to allow users to inspect and explore their learner models.

The core of open learner modelling is the provision of an effective interface to the learner model. Section 2.2.1 described several such systems. However, with the exception of VIUM (Uther 2001), there are no current open learner systems that catered for learner models that consist of hundreds of components being displayed simultaneously. This is important – as our user models would be structured by potentially very large ontologies, we want an interface that is scalable to large amounts of data and maintain the desirable features of visualisation systems (Shneiderman 1996). VIUM has been developed with such information visualisation principles and scrutability in mind, and evaluated with user models consisting of over 700 components. In the LOSUM framework, we chose to utilise VIUM as the basis for SIV, and particularly, UI-SIV (the configuration of SIV for the purpose of open learning modelling in the aforementioned blended e-learning environment).

In Section 3.2, we discuss the blended e-learning environment which has provided the context for the research work in this thesis. We follow this up with a user view of UI-SIV in Section 3.3.

3.2 Content Delivery System

The context of our research and evaluations is a course taught at the University of Sydney, a senior level undergraduate course, User Interface Design and Programming (UIDP). This has blended on-campus and online learning, with a full program of online learning content supplemented by live lectures, tutorial/laboratory sessions and a project to build a prototype interface. This is similar to the first scenario in Chapter 1.
Central to the course is the course website. It is from here that students access the learning topics, consisting of tutorials and online lectures. There are 235 slides divided between 19 learning topics that the students are expected to use over the semester. These learning topics address two areas, design and programming. Figure 3.2 shows an overview of one such learning topic. Each learning topic consists of an online lecture component, and a tutorial component. We discuss each of these components below.

### 3.2.1 Online lectures

Online lectures consist of two components: slides with associated audio narrative. In practice, they are designed as a virtual lecture. The audio narrative provides the bulk of the information, with the usual visual slide providing a framework and some elements of the lecture. The audio of the shortest slides lasts just 10 seconds while some of the longest are around 600 seconds.

To learn the main elements of a lecture, students should not only view the slides but also listen to the audio (and make their own notes) in order to gain an understanding of the concepts discussed. They
Now we come to the final part of the specification. These are the functional requirements of usability, these are really important because these are statements about how we are going to decide whether our system is successful or not. These are statements we can evaluate, assess, and say "yep, we did it, we achieved what we were supposed to" or "we didn't". If a client is looking at a system and looking at the specification deciding whether to pay, it is these sorts of requirements that will be used to define whether the system is acceptable.

Each one of these systems will have a rationale, and that will link to the One Sentence Statement, and the documented assumptions about the user. In order to evaluate any one of these, we are going to define some concrete tests for each requirement, so we can have actual users try each of those tests, and then we can see whether the requirements were fulfilled.

Let's look at an example. A computer science researcher is going to be using a bibliographic database. We realised already that if they're going to make use of a bibliographic database, then they are several tasks they need to do, and one that we will identify is that you will need to add new entries to the database. So this requirement says that 90% of first time users can add a new bibliographic entry within 2 minutes. The rationale comes largely from the one sentence statement, but as soon as we did the task analysis, and realised we had to add entries to the bibliographic database, we also realised that new users would potentially do this more slowly than experienced users who had made use of the interface many times before. So what we are saying here is that if we are going to fulfill the requirement of the one sentence statement, that the system be able to be used quickly and easily, we're going to pin that down and say that means you can add a new entry, from a cold start, in 2 minutes. To evaluate that, we're going to have to define some actual bibliographic entries, we're going to have to get some real, new, users and have them sit down, and try to use the interface, to add those entries to the database. Now if 90% of those users can do it within 2 minutes per entry, then we've succeeded.

We also have a matching requirement for experienced users, and this says, for example, they should be able to add a new entry within 1 minute. Again, we set a number of tasks, we'd have to train some users by having them use the system for some time until we deemed them to be experienced users, and then we'd have them add these extra entries, and time them. If 90% of them can do the job in a minute, we've succeeded.

Notice that the 90% is quite typical, it allows for the fact that every group of users has a few people who might take a lot longer, for whatever reason. It's a bit tough to expect an interface to work well for all users, all the time. So it's typical to require 80 or 90% of users to do a particular task within the time set.

There's a third entry here of a requirement that 90% of new users can enter an author search to retrieve the matching items within 30 seconds. Each of the tasks will have a set of their own requirements, just like this. It may be that some tasks won't be relevant to new users. It may be some are only relevant to advanced users, and so on. But all of this will come out of the task analysis, and the one sentence statement.

Figure 3.4: A transcript of the audio narrative from the slide shown in Figure 3.3.
then need to apply this in their tutorial and assignment work. This parallels the way many lecturers use overhead slides for live lectures. Students are expected to attend at times they can choose, but partly dictated by the assessment and recommended reading deadlines for attending each lecture.

**Overview of a typical slide with audio narrative**

Figure 3.3 shows a typical slide from the course for delivering the online lecture content. The interface has a number of major elements:

- Navigation elements to other parts of the course website across the top of the page. Students can visit:
  - **News**, for current news about the course or website.
  - **About**, for more details on the course such as breakdown of marks for assessment.
  - **Project** shows a description of the semester long project.
  - **FAQ** contains frequently asked questions about the course.
  - **Profile** displays their user profile, showing their marks and their visits to the online lectures.

If applicable to their enrolment or role in the course, the following links are also available:

- **SOFT3802** with news and details purely for students enrolled in the advanced stream of the course.
- **Staff** section for tutors to enter marks and read teaching material.

- The left-hand side has fast time-based navigation to all of the learning material including the online lectures and lab activities. These are shown in chronological order grouped by the week of the semester.

- The online lecture slide itself in the middle of the screen consists of three main parts:
  - Slide controls for navigating to the previous and next slide, and controls for playing the audio. In this particular case, the audio lasts for 186 seconds. The audio is delivered in Real Audio format, starts playing by default when the students visit the slide, but they can pause or restart it if they wished. A transcript of the audio narrative is shown in Figure 3.4. We can see that the narrative provides a much more in-depth treatment of the slide topic than just the visual slide.
A visual slide about *Functional requirements for usability*, with bullet points that present the key concepts and some examples.

An “annotation” text input area for students to take their own notes as they listen to the audio and read the slide. These notes are saved in the system and an interface is provided elsewhere on the website for students to print a summary of all the slide content along with their notes.

### 3.2.2 Tutorials

To supplement the material presented in the online lectures, students are scheduled to attend weekly tutorials held in computer laboratories where they apply the knowledge they have gained as the semester progresses. Tutorials comprise of a homework component, and a series of laboratory activities.

The tutorials are marked out of 10 by a tutor. These marks are all mastery-based and often involve group activities. Accordingly, they tend to be an indicator of whether learners actually participated in labs and handed in solid preparation homework. These marks form 30% of the final mark for the course. A final exam at the end of the semester covers the majority of the material taught in the tutorials and online lectures, and makes up the remaining 70% of the course grade.

### 3.2.3 Readily Available Evidence

The learning topics provide potential forms of evidence sources that we can use to gather evidence for the learner model. In Chapter 1 we discussed the types of readily available evidence in blended e-learning environments and their associated granularity, purity and reliability. We now discuss how we

<table>
<thead>
<tr>
<th>Source</th>
<th>Interaction</th>
<th>Evidence</th>
<th>Granularity</th>
<th>Purity</th>
<th>Reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online Lecture</td>
<td>Read slide</td>
<td>Duration of visit from web log data</td>
<td>Fine</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td></td>
<td>Listen to audio</td>
<td>Typed annotations on slides</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Take notes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tutorial</td>
<td>Do homework</td>
<td>Tutorial mark given by tutor</td>
<td>Coarse</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td></td>
<td>Do laboratory activities</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Final Exam</td>
<td>Answer exam questions</td>
<td>Examination mark given by markers</td>
<td>Coarse</td>
<td>Low</td>
<td>High</td>
</tr>
</tbody>
</table>

Table 3.1: Evidence sources in the UIDP course.
apply this to the data available to us from the UIDP course. A deeper discussion of the evidence and how it is utilised to reason about the learner’s knowledge is presented in Chapter 7.

Table 3.1 shows the evidence sources available to us in the course, along with the relevant interactions associated with the source and the nature of the evidence. We chose two sources of evidence to feed into the learner models: the online lectures and the tutorials. From Table 3.1 we can see that they differ considerably in their properties. We also include the final exam as an additional source of evidence in this discussion, though it is not fed directly into the learner models. We discuss below each type of evidence and their associated properties.

**Online Lectures**

The online lectures are typical of the large range of online resources which allow students to study the course content at any computer they wish to use and at any time. Unlike a conventional face-to-face lecture, the student can stop the lecture at any time, replaying an element that they did not understand. Students can return to the lecture or a particular slide in it, as they need, for example, when tackling an assignment, working on a tutorial task or preparing for final examinations.

The logs of student accesses to the online lectures are one form of readily available data in typical widespread e-learning that should be useful for modelling learning activity. As a student plays through each slide and listens to the audio, it is intended that they should learn some of the material taught in that part of the lecture: that is the whole reason for that online lecture’s existence. Therefore, we use the web logs to extract the length of time a student stayed on a particular slide: a student who listened to all of the audio in one sitting should stay on the slide for at least the duration of the audio. We refer to this as audio evidence.

Although we also had the facility for students to type annotations as they listened to the slide, this was an optional feature – many students preferred to take notes by hand or have them recorded in their own text files. Therefore, we chose not to include annotations as a form of evidence for the learner model.

In Table 3.1, we can see that the online lectures have a fine granularity – each slide has a description of very specific concepts which together contribute to the learning of a coarser grain topic. The purity is also very high as a result of this. The nature of the slides is to sequence concepts in a topic over as many slides as necessary; therefore, each slide will have a small number of concepts associated with it. However, the reliability of the audio evidence is low – although we have logs that contain the duration of the student’s visit to a particular slide, it is impossible for us to say if they were paying complete attention to the narrative and topics.
Tutorials

The tutorials formed the backbone of the face-to-face interaction between students and instructors. As it was compulsory for students to attend the tutorials, it was logical to include the tutorial mark as evidence in the learner model. In particular, as marks were given by tutors grading work completed, the high reliability of the evidence was desirable, especially when the other evidence sources provided evidence of much lower reliability. We refer to the tutorial marks as tutorial evidence.

From Table 3.1, we can see that the evidence gathered from tutorial marks are very different in their properties to that of the online lectures described previously. A tutorial consists of a single mark that is of high reliability (as it is given by a human tutor after assessing their work face to face in a laboratory session), but is of coarse granularity (the tutorial is associated with a topic that covers a high-level concept) and low purity (there are several tasks in the tutorial that assess the student’s knowledge and ability in the concepts that make up the topic).

Final Exam

In the learning context, the final exam clearly cannot provide evidence to the progressive learner model as the exam is held at the very end of the course; however, it still provides a valid source of evidence for our analysis of student progress during the semester. Moreover, in cases where a class model has been created for a whole cohort for use in future years, the final mark is a critical part of the cohort model.

We can see in Table 3.1 that the evidence from the final exam is different again from that of the audio and tutorial evidence. It is most similar to that of the tutorials – the overall coarse-grained concepts are assessed in a small number of questions, each question containing a number of parts. The purity of the evidence is quite low - each part will assess the student’s knowledge of several concepts at once, and each question will branch across a number of topics, whereas the tutorials covered only a single topic.

3.2.4 Mapping Evidence to Concepts

All of the learning topics have concepts associated with them. These concepts are taught in the online lectures and the tutorials for that learning topic. The learner models have a component for each concept across all the learning topics; the audio and tutorial evidence feed into these components. The user modelling system will resolve the evidence and present a value to represent the user’s knowledge of the concept.
A scrutable system will allow users to inspect and question these resolved values, which is one of the core goals of this system. We want users to be able to inspect the evidence for the system’s belief about their knowledge of a particular concept, so it is important that the links between components in the user model, the evidence and their originating evidence sources, and the content delivery system are made explicit in the user model interface. A user view and interface overview of the user model visualisation tool SIV is described in the following section.

### 3.3 UI-SIV

Scrutable Interface Viewer, or SIV, is the centerpiece of our research, and is the user model visualisation from the diagram in Figure 3.1. It is our solution to the scrutable user model interface problem and has been built using VlUM (Uther 2001) as a foundation. VlUM was introduced in the previous chapter (see 2.2.2) as a tool for the visualisation of large user models. SIV extends VlUM by utilising ontologies as a way of structuring the user model and as a way to infer across granularities. The need to provide such a structure was the primary motivation for using ontologies to make the visualisation useful.

An important feature of SIV is the different configurations for the interface depending on the task and domain. In this section we describe the user’s view of SIV, as it appears in the context of the UIDP course and our main evaluation in Chapter 8, and cover all of the interface elements in this configuration. We call this particular configuration UI-SIV. We will describe the internal architecture of SIV in Chapter 4. Other interface configurations and domains that SIV has been applied to are presented in Chapter 9.

#### 3.3.1 Focussing and Navigation

Figure 3.5 shows a screenshot of the SIV interface. The left hand side shows the visualisation, and the right hand side shows information about the focus concept. The visualisation has the concept user interface critique in focus: it has the largest font size, spacing and brightness. Learners can navigate through the learner model by clicking on concepts to put them in focus. When a concept is in focus, a spanning tree is generated and related concepts are assigned a font size based on their depth in the spanning tree, with unrelated concepts shown in a dimmed small font. The visualisation changes so that the newly selected concept becomes the focus.

Related concepts are shown in progressively smaller fonts, the idea being that concepts that are less related are smaller. The concept heuristic evaluation (at the top of the display) is one example of a concept related to user interface critique in this screenshot.
The horizontal positioning of the concepts on the visualisation is set at three different levels: fine grained concepts (aligned furthest to the right), coarse grained concepts (aligned at the left but inset from the margin), and selected concepts (aligned against the left margin). Selected concepts are described in more detail in 3.3.6.

It is important to note that the user model in the figure enables the users to see a large model all at once, with several hundred concepts presented. Uther conducted usability studies of VIUM with 100, 300, 500 and 700 concepts displayed at a time. The studies showed that accuracy, speed of actions and navigations of the user model were constant up to 500 concepts and only somewhat slower and more error prone at 700 concepts.

As was the case for VIUM, the colours are used to represent the learner’s knowledge of the concepts. When the model indicates the user knows a concept well, it is coloured green. Otherwise, concepts that the system believes a user does not know well, or the system lacks in supporting evidence, are coloured red.
Figure 3.6: The concept *cognitive modelling* is in focus, and shown at Term Expansion level 1, 2 and 5 (as indicated by the Term Expansion radio buttons on the top of the visualisation). As the expansion level increases, the number of visible related concepts is also shown.

We can see a few red concepts such as *predictive usability* and *usability* in Figure 3.5. In the display, the standard is set to 0.5; so concepts with scores above this value appear in green, and those below in red. The further away from the standard, the more saturated the colour. For example, a brightly saturated green is much better (i.e. higher) than one that is one that is less saturated. The standard is a value that the teacher can decide, and can be raised or lowered to tailor their teaching requirements. For example, if the teacher expects mastery performance, they might set the standard high, say at 80%.

### 3.3.2 Term Expansion

The **Term Expansion** radio buttons allow users to change the number of visible terms on the visualisation. The **Term Expansion** radio button value determines at which point the spanning tree expansion should end, and any concept not in the spanning tree is automatically given the dimmed small font indicating they are not relevant to the user’s current view. The primary reason for allowing users to decide their own cut-off point is to provide control over the number of readable concepts on
the visualisation: in the configuration of UI-SIV used in our evaluations, it defaults to a depth level of 2. The interface has five buttons, though the number of depth cut-off levels required really depends on the peerage of the underlying ontology, as well as the resolution of the user’s browser window.

Figure 3.6 shows SIV at different expansion levels for the concept cognitive modeling, as indicated by the selected radio button option above the visualisation. The left screenshot shows SIV at Term Expansion at level 1 (the selected radio button is the far left option). The middle screenshot shows the same concept at Term Expansion level 2; there are more concepts visible at this level, as we can now see all the related concepts for the concepts visible at Term Expansion level 1. The right screen shows the same concept expanded to Term Expansion level 5, where we can see all related concepts down to 5 levels of expansion.

This provides a good way to get an overview of the whole user model as virtually the whole model is exposed. We can easily see that the user model in this case is indicating an overall good performance by the learner as the model is mostly green. From here the learner might want to focus on the concepts that are red.

### 3.3.3 Display of Evidence

One important difference between VIUM and SIV is that the latter makes use of a user model representation explicitly designed for scrutability. It uses Personis Lite, a user modelling server based on the um scrutable user modelling shell (Kay 1999). The UI-SIV display presents an interface to that representation so that a learner can see not only what values the user model has, but also the system’s reasons for its beliefs.

On the right hand side of Figure 3.5, learners can see the evidence that contributes to the final score for that concept. This is important as a support for reflection since it is here that learners can see the reason that the system concludes what they know. For example, if they have failed to attend the online lectures, they will be able to see this clearly.

UI-SIV has two primary sources of evidence: audio and tutorial. We use metadata to determine what concepts an audio slide or tutorial teaches: accesses to a slide that has been tagged with concepts form audio evidence for those concepts in the user model; tutorial marks in turn become evidence for the corresponding concepts that are tagged for that tutorial. There is also an inferred evidence source that aggregates and boosts the concept score based on the learner’s knowledge of related concepts to the currently selected, focused concept. The metadata tagging of content is described in Chapter 6, and inference is described in Chapter 7.
In the screenshot we can see there are two pieces of audio evidence, and no tutorial evidence. In this case, the user has visited two audio slides that have been tagged with the concept *user interface critique*. This has resulted in evidence being stored in their user model for the concept. The score for this concept is 0.80 based on this evidence.

### 3.3.4 User Model Views

Across the very top of the main part of the interface in Figure 3.5, learners are able to select the learner model view. There are three views:

- **My User Model** colours the concepts, based on values from the evidence collected about the learner. The number of red and green concepts indicates knowledge of concepts below or above a defined standard, in this case, 50%. For example, if the learner had a score for the concept *colour* at 46%, it would be shown in red as it is less than 50%. This is the basic overview of the learner model and the starting point for reflection: if there is a great deal of red, this immediately signals problems. Navigating around this model, the learner can check which areas are their weak points.

- **Me vs. Average** shows the learner’s knowledge compared to the average of the class instead of the absolute 50% standard used in the My User Model view. For example, if the learner had a score for the concept *colour* at 46%, and the average over all the students in the class was 23%, then the concept would be green as the learner’s score was higher than the average. This is also a foundation for reflection since it gives an overview of the individual’s performance compared with their peers. This model is particularly useful earlier in the course, when many of the course concepts have not yet been studied.

- **Average of Class** shows the average score of a concept across the class. In this figure, the colours of the concepts are set against a 50% standard exactly as used in the My User Model view. For example, if the class had a score of 56% for the concept *user interface critique*, then the concept would appear green as the class average is higher than 50%. This is particularly useful for teachers so they can reflect on the progress of the class, identifying areas of difficulty.

Figure 3.5 shows all of these, as they appeared in our main evaluation experiments described in Chapter 8. It is important to emphasise that the underlying SIV framework can provide a range of such absolute, relative, individual and group models, as we discuss in Chapter 4.
We now focus on elements of SIV, shown at the top left of Figure 3.5, above the SIV visualisation. There are three primary interaction buttons at the top of the interface. The first is Search, which allows users to find concepts that are not immediately visible. Figure 3.8 shows the user searching for the concept colour, on the left screenshot, and then in the right screenshot, the results colour theory and colour are highlighted in the visualisation. If there are no relevant results to the user’s query, then all concepts appear small.

3.3.6 Selecting and Deselecting

The Select/Deselect button allows users to tag the currently focused concept: this moves it to the far left margin of the visualisation. Users can tag several concepts of interest to them and so can easily retrieve them at a later stage. Figure 3.7 shows the concept storyboard being selected. When the user focuses on a different concept (in the rightmost screenshot GOMS), the selected concepts are still noticeable because they are at the far left of the learner model, and this helps users to readily see how or if they are related to the current focused concept. In this case, the size of the font for storyboard indicates it is closely related to concepts like cognitive walkthrough and system response.

3.3.7 Inference

The Infer button allows users to see what inferences can be made about a particular concept based on the knowledge of related concepts. Our motivation for providing this functionality is to allow users to get a sense of their performance for concepts that have little or no direct evidence. This is absolutely critical for the types of data we want to exploit for learner modelling: the particular aspects a learner needs to reflect on may have no direct data. With this inference power, based upon the ontology
underlying the learner model, we can make use of the evidence that is available to fill in the gaps at other granularity levels in the learner model. Although primarily designed for inferring about coarse grained concepts, the inference functionality can be used for any of the concepts in the visualisation. We present the algorithms in Chapter 7.

3.4 Summary

This chapter has described our solution to the scrutatable user interface problem. We firstly described the theoretical underpinnings and motivation for our solution in the context of existing research. We have followed this with an introduction and description of the UIDP learning environment and the user interface for UI-SIV. The descriptions in this chapter are, in effect, examples of the SIV and LOSUM system as seen by the students in the UIDP course from which we drew the participants for our evaluation described in Chapter 8.
In particular, we can see on the diagram that SIV relies on both the ontology as a basis for structuring the user model and inference across granularities. Also, the metadata defines what can be modelled about the learner: it is a critical link between the evidence sources and the user model. This chapter provides a concrete introduction to LOSUM in terms of the user’s view of the SIV interface to a scrutable use model. We now describe the underlying machinery that drives this.
Chapter 4

Scrutable Inference Viewer

Because this thesis explores the way light-weight ontologies can support scrutable user modelling, a critical requirement of this is a solution to the scrutable user model interface problem. We do this via an interface to support scrutability of large user models.

In this chapter we move beyond the user view of SIV, the Scrutable Inference Viewer, as presented in the last chapter. We now show how SIV fulfills the role of the user model visualisation in LOSUM as shown in Figure 4.1. This means that it takes the structure for the user model from the ontology so that the visualisation can present the user model effectively.

In our work, we chose to use the tool Visualisation of Large User Models, VIUM (Uther 2001), for a number of reasons. Firstly it fulfills visualisation roles as proposed by Schneiderman (1996) with the

![Diagram](image)

Figure 4.1: System overview showing the SIV system component discussed in this chapter highlighted in dark blue.
exception of Extract. In particular, development on VIUM placed a strong emphasis on overview of
the model with scrutability in mind. It is also the only visualisation and interface that placed
significant design considerations into large user models, consisting of hundreds of components.

One of the serious shortcomings in using VIUM to scrutinise user models is that it requires structure
on the data within the user model. Indeed, the initial motivation to VIUM was to visualise large
learner models in the context of the University of Sydney Graduate Medical Program\(^\text{15}\) with its 526
learning topics. The rigorous evaluation of VIUM was, however, conducted in the context of the
movies domain for two reasons. Firstly, the more widely understood domain meant that the evaluation
could involve a broad population of users. Secondly, and critically important for this thesis, Uther was
not able to create a sensible structure over the medical learning topics, nor was there suitable metadata
on the bank of several thousand test questions. These issues and limitations are discussed in Chapter 3
of Uther’s dissertation.

This thesis aims to make VIUM useful for learning contexts by providing a way to structure the user
model with an ontology. A number of enhancements and modifications have been applied to VIUM;
this new version has been named SIV, as described in the previous chapter. Table 4.1 shows a
comparison between the original VIUM and SIV. The aspects of the visualisation are listed along the
left-hand side. The VIUM interface was described in Chapter 2, and a user view of these features in
SIV was provided in Chapter 3. The core changes and differences between VIUM and SIV are
highlighted in yellow on the table.

The first three aspects (1-3) relate to a change in the graph representation that allows the visualisation
of richer structure associated with ontologies. Functionally, these aspects are the same for VIUM and
SIV (the user view for the graph structure, component values and comparative user model views are
described in Section 3.2). In terms of implementation, the core difference is that VIUM reads all three
from a single serialised source (an RDF file), whereas SIV utilises separate serialised sources. The
advantages of this approach and implementation details are discussed in Section 4.1.

The last aspect (8) is concerned with a visualisation-based approach to addressing the \textit{fan-out
problem}. In Chapter 1, we discussed that the \textit{fan-out problem} occurs when graph-based ontological
structures exhibit a high variance in connecting edges between nodes, causing problems for
visualisations designed with an expected number of connections. SIV addresses one half of the \textit{fan-
out problem}: when the ontology structure has nodes with a very large number of connections. This is
discussed in Section 4.2.

\(^{15}\) http://www.gmp.usyd.edu.au/
This chapter describes SIV in respect to these modifications to VIUM that were necessary for it to exploit an ontology to define a graph structure over the otherwise unstructured collection of user model components. This chapter also discusses other technical enhancements to SIV along with an evaluation of performance using different XML parsers.

4.1 Graph Representation

Essentially making the move to an ontologically structured graph over a user model creates new representation issues in the graph representation with the visualisation tool. In this section we address the limitations of the VIUM graph representation format and the modifications necessary to support an easily extensible and flexible ontology based user model.

The first major issue was the need to separate the ontology from the user model data. The format used by VIUM had both the graph structure and the user model data in a single source file, represented in RDF. One of the goals of SIV was to easily change between different graph structures (or ontologies); however, the coupling of the user model data and graph structure in VIUM required us to modify the underlying representation. These differences in representation are shown in Figure 4.2. SIV can easily...
swap between different overlay or ontology files, leading to greater flexibility of the underlying ontology choice and the number of different views of the user model data.

The introduction of the overlay approach was a major innovation in SIV. During runtime, new overlays, effectively “views” of the user model, can be applied to the visualisation, and the display will change accordingly. This is because we would like to enable users to easily switch between different ways of seeing their user model. Although this aspect was implemented in VIUM, the overlay approach in SIV no longer requires all the user model data to be loaded at startup. This meant that the visualisation could now support an arbitrary number of user model views.

One important insight is that the components of a user model are multifaceted and it is desirable to independently represent an arbitrary number of such facets: for example a student’s knowledge of usability may have several facets such as a score as determined by a teacher, a different score determined by the student self-assessments, the score determined by some other means. Of particular importance is the value of relative, as well as absolute scores. For example, a student may score 80% on usability. If the student aspires to just pass a subject, and the required standard for a pass is 90% on usability, the score has very different significance from a situation where a pass required 50% for usability. This recognizes that absolute values on user model components are not generally meaningful. What is meaningful is the comparison of the component value against a suitable standard, perhaps defined by the user, as in VIUM and as described in Chapter 2.
4.1.1 Implementation of Ontological Structure

In this section, we describe in detail the representation for serialising the user model data in VIUM, and the changes to the serialisation that allows for separate ontology and user model overlay files in SIV.

Figure 4.3: Sample of original RDF format used by VIUM. Of note is the combined domain and user knowledge.

Figure 4.3 shows a sample of the RDF format used by VIUM for the component Fuel use in cells, with the combined user model and domain model data. A user model file would consist of a similar entry in the file for each component (which would number in the hundreds). Shaded in yellow (lines 2-5) are the title and the relationships for this component. Shaded in pink (lines 7-10) and blue (lines 12-15) are user data. The pink area shows the dataset for the user’s average marks (line 8) which has a score of 0.8 (line 9) and a certainty value of 0.4 (line 10). The blue area shows the dataset for curriculum, in the same format as the user data.

This posed a number of limitations. The first was the coupling between the graph structure and the user model data. We could not easily change or update one without also retrieving the other during runtime. Second, as the number of views in the user model increased, there was a linear increase in the physical size of the RDF file. Separation of the graph structure and user model data would mean we could not only easily switch between different structures and datasets, but it would be done on demand, resulting in faster startup times when accessing the user model.
In SIV, the ontology file was parsed first, and a number of overlays parsed and the data incorporated into the final visualisation of the user model at run time. Figure 4.4 shows a portion of an ontology file generated by Mecureo. In these files, there would be a concept element for each term defined in the original glossary source (more details of this are in Chapter 5). The ontology file contains the information to render a graph structure: there are links to other concepts on lines 4 and 5. In addition, there is additional ontological information in the form of link types. These are hasParent and hasSynonym in Figure 4.4.

```xml
1   <Concept rdf:ID=" http://www.gmp.usyd.edu.au/topics/11.html">
2       <dc:Title>=Fuel use in cells</dc:Title>
3       <hasParent rdf:resource="http://www.gmp.usyd.edu.au/topics/125.html"/>
4       <hasSynonym rdf:resource=" http://www.gmp.usyd.edu.au/topics/326.html"/>
5   </Concept>
```

Figure 4.4: Sample of ontology OWL format for the same concept shown in Figure 4.3 for SIV. Of note is the ontology only contains the domain data. User model data is serialized to a separate file.

```xml
A1  <?xml version="1.0" encoding="ISO-8859-1"?>
A2  <resultset>
    ...
A3  <result concept="Fuel use in cells" value="0.8"/>
    ...
A4  </resultset>
```

```xml
B1  <?xml version="1.0" encoding="ISO-8859-1"?>
B2  <resultset>
    ...
B3  <result concept="Fuel use in cells" value="0.4"/>
    ...
B4  </resultset>
```

Figure 4.5: Snippets from two separate user model overlay XML files, Mark (line A3) and Reliability (line B3), for the concept Fuel use in cells. The corresponding domain ontology snippet for this concept is shown in Figure 4.4.

The overlay files themselves simply list the concepts in the user model and a corresponding value and are accessed by passing the system a URL to the appropriate XML file. SIV loads two initial overlay XML files (if they are specified in the starting parameters) – an overlay for the colour (representing the mark) of the concepts, and an overlay for the horizontal positioning (representing the reliability).
The values are interpreted by SIV in the same way VIUM interpreted the mark and reliability values in its original RDF user model representation.

For example, lines 9 and 10 in Figure 4.3 show a mark of 0.4 and a reliability of 0.8 for the concept *Fuel use in cells* in the VIUM RDF serialisation. In SIV, this relates to values in two overlay files, as see this in Figure 4.5, line A3 shows the mark for the concept *Fuel use in cells*, and line B3 shows the reliability.

As mentioned previously, the overlay approach allows for an arbitrary number of user model views. In the example, a separate pair of overlay files can be used to specify the mark and reliability of the *curriculum* dataset from Figure 4.3.

One important change in SIV is that all the data processing for user model views is conducted from the user modelling system supplying the user model data. This is significant, it means that any user modelling system that can serialise its data in the same format (and supply a corresponding ontology) can utilise SIV to visualise the user models it stores.

**Internal representation modifications**

VIUM stored the user model graph as a series of arrays. Although sufficient for the purpose of parsing and storing the original graphs when VIUM was developed, it did not provide enough flexibility for the ontology based graphs. An example of the additional information we needed to represent can be seen in Figure 4.4: lines 3 and 4 show the same relationships in the graph as lines 4 and 5 from Figure 4.3, with the added information for the type of relationship, in this case, *hasParent* and *hasSynonym* respectively.

The array representation of the graph has been replaced by an object based representation. Each node and peer (relationship) in the graph is represented by its own instance at runtime. Information about nodes and peers can be easily added and retrieved.

Although the runtime memory usage is higher with an object based representation, we evaluated the speed trade-off on a Hewlett-Packard TC1100 tablet PC and determined that there was no drop in performance during runtime. Additional information about nodes or arcs can be stored as attributes in the respective class, and new attributes can be easily added.

**Inference Overlays**

Inference is a new functionality added to SIV, the interface was described in 3.2.6, and utilises its own *inference overlay* that is modified at runtime. The inference overlay is handled slightly differently from those just described. It is used to store the inferred values for concepts when the user clicks on
the **Infer** button. When this happens, the system generates the inferred value and stores it in this overlay.

When the user presses the **Infer** button as described in (3.2.6), the resulting inferred value for the concept is stored on the inference overlay. Since the inference overlay sits above the colour overlay, the inferred value for the concept is used instead of the original value by the visualisation for determining the concept colour. This means we preserve the original value for that concept on the colour overlay, and if we want the original value back, we simply clear the corresponding value on the inference overlay.

For example, if the concept *Cognitive Walkthrough* had a value of 0.5 in the colour overlay, but an inferred value of 0.65 in the inference overlay, the visualisation will choose 0.65 as the value to use to determine the colour. If the value of 0.65 is cleared from the inference overlay, then the system will just use the value of 0.5 on the colour overlay.

More details about the actual inference algorithm can be found in Chapter 7.

### 4.2 Relationship fan-out in the ontology

One of the challenges involving the visualisation of ontological structures is the **fan-out problem**. As discussed in Chapter 1, the **fan-out problem** is related to the connectivity of the concepts in the ontology, specifically the incoming and outgoing relationships for each concept.

VIUM was originally designed and tested on tuned graphs that had a set minimum and maximum number of connections for each node (approximately 6 to 10). Our initial tests on visualizing ontologies identified two problems: concepts with few or no relationships to other concepts, and on the other end of the spectrum, a few concepts connected by a very large number of relationships that resulted in very cluttered and difficult to navigate user models when those concepts were in focus.

There are two approaches to deal with this problem (in this thesis we utilise both approaches). One approach lays in the ontology construction and maintenance methods, by factoring in the requirements for visualisation and attempting to ensure a certain level of connectivity in the ontology structure. This is primarily to deal with concepts that have little or no relationships to other concepts in the ontology. We address an ontology construction and maintenance-based approach in Chapter 5.

The second approach, and the one described in this section, is through the visualisation interface to provide ontology navigation and exploration features that can manage the amount of concepts and relationships displayed to minimise the effects of highly variable fan-out. This is primarily to deal
with concepts that have a large amount of relationships to other concepts in the ontology. We identified and implemented two visualisation-based solutions:

- **Filtering of relationships**: exploiting the fact that relationships in light-weight ontologies are usually typed (such as in a hierarchy consisting of parent/child relationships), the number of concepts displayed can be limited based on relationship types. Users can adjust the interface to filter specific relationships that they only wanted to see on the visualisation.

- **Modification of layout algorithms**: visualisations can limit the amount of concepts and relationships displayed by modifying the layout algorithms. In the case of graph structures such as ontologies, we can assign thresholds to the level of expansion, and thus the number of displayed concepts, in spanning tree-based approaches (such as in fish-eye style navigations) of the ontology.

We now describe both of these modifications in more detail.

### 4.2.1 Filtering of relationships

We wanted to utilise the fact that ontologies had typing of relationships as a way to reduce the amount of expansion, by only displaying certain types of relationships specified by the user. This provided a simple but effective way to reduce clutter on the interface, and provide further refinement by showing only relationships that users wanted to see. This also opens the opportunity to exploit this in providing alternate views of a user model. For example, in an e-learning based ontology that has pedagogical relationships between the concepts, a learner might only wish to see the **pre-requisite** relationships for a particular concept.

There were three main steps to implementing this enhancement in SIV:

1. An additional attribute was added to the peer object to store its type (as a string). For the ontologies generated by Mecureo, type information was always included with the relationships for each concept when the ontology was serialised to OWL (or SKOS).

2. The interface was enhanced to allow users to choose which relationships they wanted to filter. A drop down menu provides a list of relationship types found in the ontology. This menu can be seen in the middle screenshot from Figure 4.6. When the user accesses this menu and chooses one of the items, the relationship type they pick becomes selected.

3. A modification to layout algorithm allows us to only display concepts related to the currently focused concept of the selected relationship type. All other relationship types cause those
target concepts to be blurred into the background with the irrelevant concepts, thus reducing
the amount of concepts displayed.

An example of the relationship typing integrated to the visualisation can be seen below. In Figure 4.6,
we visualize an ontology of HCI terms (left screenshot). The user clicks on the Filter menu item, and
is shown a list of the relationships in the ontology (middle screenshot). These are the relationship
types defined in the Mecureo ontology construction process. Further discussion of these relationship
types are in Chapter 5. The user selects to view only concepts with relationships of type hasParent,
resulting in the visualisation changing to only show those related concepts (right screenshot).

Contrasting the left and right screenshots we can see that the number of concepts visible is now
greatly reduced and clarity of the concepts linked by the hasParent relationship is enhanced by the
additional screen real-estate for layout of the concepts. This enhancement was first explored in the
SIV variant, SITS-VIUM (Chapter 9).

As mentioned above, the relationships shown are those found in the configuration of Mecureo that we
used in the process to create user models for the evaluation. If a different ontology format were used,
for example, SKOS, than the relationships displayed would be broader, narrower, related and
category.

4.2.2 Modification of layout algorithms

Our user studies (Apted, Kay et al. 2003; Kay and Lum 2004) indicated that some people prefer a less
cluttered screen with less term expansion while others prefer to have many concepts on the screen at
once, with a high term expansion. It may also be that, depending upon the situation, a learner may
want to reflect on a smaller or larger expanse of the model. To accommodate these differences, and
also addressing the fan-out problem, we provide the user with this facility, to alter the amount of term
expansion, as they wish.

VIUM contains an implementation of a spanning tree algorithm on the underlying graph structure that
allocates colour, font size and spacing to the concepts on the visualisation. The graphs in the original
VIUM were generated and tuned to have an average number of 6 peers.

This was done through very simple text matching, and although the relationships were often not
meaningful, it provided a graph that resulted in a pleasant visualisation with minimal clutter. It also
resulted in the spanning tree algorithm being tailored to a massaged graph that had no leaf nodes, and
a constant that affected the vertical spacing for the layout of the text.
In SIV, the spanning tree algorithms have been enhanced to handle leaf nodes and much larger graph structures. The visualisation has been tested with graphs consisting of 1300 nodes with some nodes having up to 100 peers (and some with no peers). Although a graph of this size had a large amount of overlapping text on the visualisation, the tool was able to handle the increase in processing required.

The next modification to the algorithms was the implementation of a cut-off to the spanning tree expansion after a certain depth. This meant that users could limit the concepts displayed to a certain level of “relatedness”. They can choose to see only the directly connected concepts to the one selected, or the directly connected concepts and all of their children, and so on. Beyond a certain point it is impractical to see distinctions between depth levels, however we set a default expansion level of 2, unless specified otherwise by the application.

A control element on the SIV interface has been synchronised with the cut-off limit, allowing users to choose what depth of expansion they want to see on the display. This has provided an additional navigation benefit as another way for users to explore the graph as the display changes dynamically when the depth level is changed. Section 3.2.7 describes the Term Expansion function which implements the control element as a set of radio buttons.

Figure 4.6: SIV showing the filtering of relationships on the visualisation. The menu allows us to choose which relationships to view (middle screenshot), and the results are filtered on the right screenshot.
4.3 Other Technical Enhancements

We now describe some other technical enhancements to VIUM that allow us to effectively visualise ontology-based user models.

**Modular user interface elements**

Work was carried out to abstract the visualisation from the control elements (the menus and sliders). The controls have been made modular, allowing for custom controls depending on the domain and use of the visualisation.

![Figure 4.7: A selection of different control elements for SIV.](image)

Figure 4.7 shows a selection of the different user interface elements we have developed: the first two are the menu and standard slider from VIUM, the next is a combination of the back function from the Action menu implemented as a button and the standard slider as a single element, followed by an early version of the **Term Expansion** element implemented as a slider and the last two are the elements from UI-SIV as described in Chapter 3.

**4.3.2 Performance Issues**

One of the problems encountered during the early development of SIV was the relatively slow load times of the applet. We identified two main bottlenecks that caused this: the number of additional libraries required by SIV that would be downloaded when the applet was initialized in the browser, and the fetching and parsing of the XML files that constituted the user model. We modified the VIUM
graph parsing component to have a composite structure, allowing us to easily change the parsing component for a different implementation.

These different graph implementations allowed us to easily test different XML parsing libraries. The majority of the work done for SIV was using an OWL based representation of the domain. Initially we switched from using the Xerces XML parsing libraries\[^{16}\] to Jena 2\[^{17}\]. However, the slow parse times and the relatively large file sizes made it infeasible to deploy SIV in an environment where users could access it over a potentially limited connection to the internet. We moved to a package called QDXML\[^{18}\] and modified it to fetch and parse files off a URL. We ended up with a very fast parser that had relatively small runtime memory usage and very small file size, resulting in the SIV applet and all associated files (excluding the user model XML files) being under 200k. Table 4.2 shows a summary of the parser performance for the different XML parsers we implemented, tested on a Hewlett-Packard TC1100 tablet PC.

<table>
<thead>
<tr>
<th>Small Ontology (189 concepts, 552 peers)</th>
<th>Large Ontology (1131 concepts, 9964 peers)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Usage</td>
<td>Time</td>
</tr>
<tr>
<td>Xerces</td>
<td>1166352</td>
</tr>
<tr>
<td>Jena 2</td>
<td>10370448</td>
</tr>
<tr>
<td>QDXML</td>
<td>1483776</td>
</tr>
</tbody>
</table>

**4.4 Evaluation of SIV**

Aside from the performance testing described in the previous section, a number of usability evaluations have been carried out during the course of this thesis. We list these below and how they relate to the design of SIV described in this chapter:

- Chapter 5 describes SIV as an interface to address the ontology interface problem. It evaluates the visualisation as a tool to explore a fully automatically generated ontology with overlaid user model data. The evaluation asked users make use of a slider to adjust the view of the user model to show data coloured against different meaningful standards.

- Chapter 6 evaluates SIV incorporated as an ontology navigation tool to address the metadata annotation problem. This evaluation demonstrates the use of SIV using just the ontology file as input with no overlays of user model data.

\[^{16}\] http://xerces.apache.org/
\[^{17}\] http://jena.sourceforge.net/
Chapter 7 evaluates the use of ontological inference in SIV. As described in this chapter, the results of inference are stored in a special type of overlay that builds upon the user model data utilised by SIV.

Chapter 8 presents an evaluation in an authentic learning context, making use of all the enhancements described in this chapter: the ontology and different overlay-based user model views, and the approaches to address the fan-out problem.

Chapter 9 describes a number of different systems that incorporate SIV. These demonstrate the flexibility and robustness of the representations that allow SIV to be applied to different domains. Importantly, they demonstrate the use of different ontology representations and user model overlays.

4.5 Summary

Ontologies help address some of the issues associated with the scrutable user model interface problem. They also create new needs and challenges as described in this chapter. Equally, they create new opportunities for user views of large user models, as in the case of the filtering opportunities.

There are many visualisation tools available, and many of these may be amenable to exploitation of ontologies to structure the data being displayed through similar approaches as we have done in the case of SIV, built from the foundations of VIUM.

In this thesis, we utilise ontologies that have been automatically generated from glossary sources utilising Mecureo. However, the approach described should be generalisable to ontologies created with other techniques. At the same time, the approaches we have taken to enable the structuring of user models based on an underlying ontology should be generalisable to other visualisations.

The contributions provided by the enhancements to SIV are:

- The ability to easily change the ontology as required, thus allowing a different interpretation of the user model data, effectively we can have many user models of the same user in different conceptualisations of the domain.

- The ability to easily change the user model overlay data, especially at runtime, allowing us to cater for an arbitrary number of different user model views. Of importance is that this allows the tool to be utilised in any user modelling system that utilises an overlay approach and supports serialisation of the domain model and user model data in a format readable by SIV.

- Bandwidth efficiency, as different user model views are only downloaded as required.
• Addressing the *fan-out problem* in the case of components that have a high degree of fan-out utilising approaches to filter and limit the number of components displayed.
Chapter 5

Constructing Light-weight Ontologies for User Modelling

Because this thesis explores the way light-weight ontologies can support scrutable user modelling, the ontology plays a pivotal role. Figure 5.1 shows an overview of LOSUM. The ontology is generated by Mecureo (in dark blue on the left) underpins the visualisation of the user model through SIV (in dark blue on the right) by providing an underlying structure that can be exploited for ontological inference and reasoning about users. At the same time, the ontology also forms a vocabulary for the metadata term set that is used to annotate the domain content, in our case, learning objects. Therefore we need to be able to construct these ontologies in an efficient manner and ensure they have the right attributes to support the user modelling and visualisation. We call this the ontology construction problem.

Figure 5.1: System overview showing the core components of LOSUM.
Another issue is the **restricted ontology problem**, where the ontology-based metadata vocabulary does not provide all the concepts a user wishes to use in their annotations. We want an ontology refinement process that allows easy addition of these missing concepts, yet can maintain scrutability of the ontology.

There is also a need for interacting with the ontology because it will ultimately be an underlying structure for a user model, which is also important for the system developer and maintainer during the ontology engineering process. We call this the **ontology interface problem**.

Although ontologies have the potential to be very useful, they are generally difficult and time-consuming to construct manually as well as maintain. This is especially true with heavy-weight ontologies due to the overhead of ensuring that they are formally correct. We believe that in many cases where people want to create user models, having access to a heavy-weight ontological domain model is costly because of these construction and maintenance issues, and a light-weight ontology will often suffice for the task.

This is especially true for course instructors who might seek a specialised ontology for their teaching domain, but lack the engineering skill to create such an ontology. This has been recognised in other e-learning systems, such as CourseVis (Mazza 2004), where there as been an emphasis on providing instructors an easy to use system.

At the time that the research began in 2002, there was a lack of suitable ontologies available for us to use. There was also the need for an ontology we could easily control. We chose to explore ways to automatically construct an ontology from existing documents and, in particular, from existing dictionaries. This offered the premise that we could justify inference and relationships in the ontology by referring to the original dictionary source, again for the purpose of scrutability.

The choice of using automatically constructed light-weight ontologies has significant implications for the nature of our work. The nature of light-weight ontologies means that they usually use few logical relationships (Fluit, Sabou et al. 2003). This means that either we must repair the ontology in order to use standard reasoning techniques, or employ different techniques altogether for inference (Huang, Harmelen et al. 2005). In this thesis we take the latter approach, with scrutability in mind. We discuss ontological inference in our light-weight ontologies in Chapter 7.

This chapter explains how we construct light-weight ontologies suitable for structuring a user model, and presents our solution to the **ontology construction problem**, by incorporating a tool called Mecureo, for the automatic construction of light-weight ontologies from existing glossary sources, into LOSUM. We provide an overview of Mecureo and its role in LOSUM.
To address the ontology interface problem, we need to provide an interface for interacting with the ontology. The fact that these ontologies will feed into a visualisation for learners has been an important factor in the ontology construction process. In Chapter 2, we discussed the importance of visualisations tools to aid open learner modelling. The ontology is a way to structure a user model so that user model visualisation tools can operate more effectively. The user should be able to select a focus concept and at the same time examine closely related concepts. An effective ontology will ensure that the user can see that the concepts are, indeed, related to the focus concept. This selection of concepts to be made visible to the user is an essential part of the visualisation that can potentially assist learners in exploring domains with hundreds of concepts.

SIV is used to address the ontology interface problem. We have added enhancements to provide output in a suitable format for SIV, as discussed in Chapter 4, and conduct a qualitative evaluation on the feasibility of using SIV as an ontology visualisation tool for the ontologies constructed by Mecureo. We discuss an ontology-based approach to address the fan-out problem.

5.1 Mecureo

We chose Mecureo, for Model Expansion and Construction Using Reverse Engineered Ontologies (Apted and Kay 2004), to build the ontologies in this thesis. It automatically constructs ontologies from semi-structured dictionary or glossary sources.

An important motivation for this approach is the ease with which we can apply our work to different domains, as long as there is a suitable dictionary or glossary available. Since we are particularly concerned with teaching systems, there are generally readily available dictionaries and glossaries, including local ones for a particular source.

Another second motivation is that we want to ensure user control and system transparency. By building the ontology from a dictionary, we can always provide users with a human understandable explanation of any part of the ontology by presenting relevant dictionary definitions.

In this chapter we describe Mecureo in the context of a framework consisting of five steps of an ontology learning identified by Maedche and Staab (2001): import and reuse, extraction, pruning, refinement, and application. These are structured as a cycle (as shown in Figure 5.2), where one can continually improve or expand the automatically-created ontology so that it is sufficient for the task or application for which it has been designed. We discuss each of these steps along with the deeper issues of the ontology construction problem, the restricted ontology problem, the ontology interface problem, and the fan-out problem.
Mecureo was developed in the context of FOLDOC\textsuperscript{19}, the Free Online Dictionary of Computing (Howe 1998). The initial stages of import and reuse, extraction, and pruning are described in this context. During the metadata annotation step for the work in this thesis (described in Chapter 6), we provided functionality to allow users to refine the ontology with concepts that were not in the ontology but critical for the task of metadata annotation and learner modelling. The ontology refinement process is, therefore, described in the context of the Usability First glossary of HCI terms\textsuperscript{20}.

The final stage of applying the ontology is presented in the form of two evaluations and an analysis of the suitability of the resulting ontologies for the purpose of user modelling, specifically for the modelling of learner knowledge, and visualisation of the ontology-structured user model using SIV. A further evaluation of the ontology as an aid to the metadata annotation process is described in the following chapter.

Although we have described the ontology engineering process using Mecureo in the context of these two particular glossaries, the approach can be applied to any glossary source to create an ontology for that domain. Demonstrations of this can be found in Chapter 9 where we provided an online version of Mecureo for students to use in their projects in a course on advanced learning technologies.

5.2 Import and Reuse

This initial step involves the identification of existing structures that can be exploited for ontology learning. As our approach is based on the use of glossary and dictionary sources, we can identify explicit structure in those types of documents that can be exploited in the ontology learning process. In particular, glossaries often exhibit a hierarchical categorization structure that provides basic granularity information about the concepts and their relationship to each other. In addition, non-

\textsuperscript{19} http://www.foldoc.org/
\textsuperscript{20} http://www.usabilityfirst.com/glossary/
declarative language

Any relational language or functional language. These kinds of programming language describe relationships between variables in terms of functions or inference rules, and the language executor (interpreter or compiler) applies some fixed algorithm to these relations to produce a result.

Declarative languages contrast with imperative languages which specify explicit manipulation of the computer’s internal state; or procedural languages which specify an explicit sequence of steps to follow.

The most common examples of declarative languages are logic programming languages such as Prolog and functional languages like Haskell.

See also production system.

(2004-05-17)

Try this search on Wikipedia, OneLook, Google

Nearby terms: Decision Support Systems « decision theory « deductive « declarative language » DECmate I » DECnet » decode

Figure 5.3: FOLDOC screenshot showing the definition for the term declarative language.

taxonomic relationships might be explicit in the form of “see also” references to other definitions in the glossary.

In this section we describe the basic structure of an glossary entry in FOLDOC that identify the basic elements that can be used by Mecureo as an initial basis for the automatically-generated ontologies. This can be generalised easily to other glossaries with a similar structure.

The FOLDOC Context

FOLDOC is a glossary of terms used in computing, and was chosen for several reasons, including its breadth of coverage and constant updates, and serialisation into a machine readable format. FOLDOC has over 14000 terms. It has also been used in other systems, such as LOMGen (Boley, Bhavsar et al. 2005), as a starting point for semi-automatic metadata annotation of learning objects.

FOLDOC has been designed with serialisation in mind. As a glossary source and through the ability to provide a machine readable format for its content, it contains well structured definitions for each of the terms. Figure 5.3 shows a screenshot of the FOLDOC website, in this case with the term declarative language loaded. We can see existing structure that we can import and reuse in the Mecureo-generated ontology to form an initial basis for the ontology:
There is a hierarchy for categories. The category name in angle brackets at the start of the definition, in this case, *declarative language* is in the category *language*.

- There are hyperlinks to other terms in FOLDOC, such as *relational language*, *functional language* and *programming language* on the first line of the definition.

- In addition to this, the entry consists of hyperlinks in the definition to other terms in the glossary. We can exploit these links in the following stage of ontology extraction to determine the type of relationship based on the structure of the definition itself.

### 5.3 Extraction

The ontology extraction phase is a core phase in the ontology learning process that builds the majority of the ontology through information in the source documents. The term definition itself is used as the source to identify additional non-explicit relationships. There is structure to the definition that can be exploited for this purpose.

For example, we note in Figure 5.3 that there is an identifiable structure, with the concept (in this case *declarative language*) defined by explaining synonymous and broader concepts first (like *programming language*), then attributes and narrower concepts (such as *inference rules*), followed by antonyms (*imperative languages*), examples (*Prolog*) and finally links to other concepts (*production system*).

Mecureo uses part-of-speech tagging to determine the type of relationships like the ones described in the last point above. In this way, Mecureo builds a light-weight ontology of the domain through the content in the dictionary. We now describe this process in detail.

**Graph Construction Process**

The word and definition tuples in the dictionary are parsed to create a digraph of keywords linked to related keywords that appear in the definition. Grammatical conventions in the definition are used to add typing to the links. Table 5.1 shows a list of the relationship types in the base configuration of Mecureo, these are discussed in further detail in (Apted and Kay 2004). A relationship strength is also associated with the relationship, one of four values are assigned: “weak”, “normal”, “strong” or “very”. This is based on the particular keyword that Mecureo has detected. Keywords are stored in a configurable file; an excerpt of one can be seen in Figure 5.4.

For example, the definition for *declarative language* in Figure 5.3 contains the phrase “Declarative languages *contrast* with imperative languages”. This shows an antonymous relationship, and the parsing process finds the word “contrast” in the keyword configuration file, which equates to a
“strong dissim” relationship strength and type. A relationship is created between the concepts *declarative language* and *imperative language* in the digraph with this relationship information.

An important feature of Mecureo is that it allows a user to scrutinise the relationships between the words in the graph. It does this by linking each term back to the original dictionary definition. The definition should provide an excellent basis for explaining to a user why a link exists. We can highlight the occurrences of the other term in the definition body. From the previous example, given a relationship between terms *declarative language* and *imperative language*, we can simply present the dictionary definitions that contain both terms (in this case, the definition for *declarative language*).

The relationships are also weighted. This is based on the structure of the definition. Broadly speaking, these weightings are interpolated from the position of the related concepts in the definition. In effect, this represents the amount of work required to discover the relationship. Lower weightings are given to words that appear earlier in the definition, to represent a stronger link. We don’t require as much

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Child</td>
<td>Specialisation or narrower concept.</td>
</tr>
<tr>
<td>Parent</td>
<td>Generalisation or broader concept.</td>
</tr>
<tr>
<td>Dissim</td>
<td>Dissimilar to, or antonymous concept.</td>
</tr>
<tr>
<td>Sibling</td>
<td>Analogy, reference or synonymous concept.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>known:</th>
<th>very sibling</th>
</tr>
</thead>
<tbody>
<tr>
<td>central:</td>
<td>strong child</td>
</tr>
<tr>
<td>example:</td>
<td>normal child</td>
</tr>
<tr>
<td>examples:</td>
<td>normal parent</td>
</tr>
<tr>
<td>include:</td>
<td>strong parent</td>
</tr>
<tr>
<td>opposite:</td>
<td>strong dissim</td>
</tr>
<tr>
<td>opposed:</td>
<td>very dissim</td>
</tr>
<tr>
<td>analogous:</td>
<td>very sibling</td>
</tr>
<tr>
<td>assumes:</td>
<td>normal child</td>
</tr>
<tr>
<td>requires:</td>
<td>strong child</td>
</tr>
<tr>
<td>qualifying:</td>
<td>very child</td>
</tr>
<tr>
<td>corequires:</td>
<td>strong sibling</td>
</tr>
<tr>
<td>prohibits:</td>
<td>strong dissim</td>
</tr>
<tr>
<td>teaches:</td>
<td>normal parent</td>
</tr>
<tr>
<td>contrast:</td>
<td>strong dissim</td>
</tr>
</tbody>
</table>

Table 5.1: Relationship types in Mecureo.

Figure 5.4: An excerpt from the Mecureo keywords configuration file.
work to discover the related term as it appears early on in the definition. This is in contrast to other systems that weight stronger relationships with a higher number, such as in (Reimer, Brockhausen et al. 2003). A consequence of this is that, in the ontology pruning step, one could simply add up the weightings along a path till a threshold “distance” was met, and extract this subset of concepts as a smaller ontology. This is described in more detail in the following section.

5.4 Pruning

Pruning involves adjusting the ontology for its’ prime purpose. We want to create human manageable portions of the ontology that represent a more specialised area of interest. Our goal is to create ontologies that can structure user models, and ultimately provide an interface to them via SIV. It is important in appreciating what SIV presents to a user as the most usable parts of their model at any point in time. In the user studies for VIUM, it was shown that users could explore and work with concepts in the order of several hundreds. Therefore, we are interested in being able to extract portions of the ontology within this order of magnitude that is suitable for visualisation of user models in a specific domain.

Point Queries

Mecureo provides a functionality called point queries, a way to query the generated ontology about a particular concept. A full description of this process can be found in (Apted and Kay 2004). Point queries are important as they allow us to extract a small portion of the ontology as a separate ontology in itself. Point queries allow us to generate human manageable ontologies of smaller size in an easy manner. At the same time, smaller ontologies means it is easier for us to deal with the ontology interface problem.

A point query can be executed from any concept (node) in the ontology. A point query involves selecting a node in the graph, and a distance to expand out by. A spanning tree from this point is created, up to a user specified threshold or distance. This subset of nodes then has the links regenerated between them to give a new directed acyclic graph. We are now left with a subgraph of the larger ontology. Multiple subgraphs can also be merged into a single graph.

Figure 5.5 shows a screenshot of the Mecureo point querying interface. We specify the node we want as the source of the point query in the text input on the top bar. In this case, we query for the concept declarative language. Mecureo then asks for a maximum distance, and also if we want to exclude nodes with less than a user specified number of relationships. In this case, the maximum distance is set to 0.75, and we chose not to exclude any nodes.
We can see a list of the nodes up to the maximum distance of 0.75 on the right frame of Figure 5.5. This has been extracted to show the complete list (Figure 5.6). In brackets next to each concept name is the distance, starting from 0.0 for the concept declarative language on line 1, to 0.734 for the concept algorithm on line 25. If we were to use a lower value for the threshold distance, say 0.5, then only nodes with a value equal to or lower than that lower threshold distance value would be included in the node list, in this case, up to the node ABSET on line 11, as all of those nodes have a distance of 0.5 or less.

**Visualisation of Point Queries**

The use of graphical or visualisation support has shown benefits in the development and deployment of ontologies. Many tools, such as Protégé (Gennari, Musen et al. 2003) and WebOnto (Domingue 1998), feature visualisations to support users during ontological engineering tasks. In particular, the Cluster Map visualisation (Fluit, Sabou et al. 2003) has been developed to specifically aid user tasks when working with light-weight ontologies that exhibit large number of instances.
Mecureo can output the point queries into the DOT format (Gansner and North 2000). This provides an immediate visual output for the structure of the ontology, allowing us to inspect relationship strengths and types, and the fan-out of the extracted sub-ontology. Thus it provides us with an initial impression of the concepts and relationships to determine if they are suitable for the task the ontology has been created for.

In the DOT output, different relationship strength and types are aggregated to parent, child, synonym, antonym, sibling, and unknown. The conversions are shown in Table 5.2. Of note is synonym: this is reserved only for strong siblings sharing a common substring in the term name. For example, the synonyms procedural language and imperative language share the common substring language and also match the text pattern from the keyword definitions to be strong siblings.

Figure 5.6: The contents of the node list frame from Figure 5.5, showing an expansion of the concept declarative language at threshold 0.75.
The DOT outputs for the concept *declarative language* are shown in Figure 5.7. Two outputs are shown, one at threshold 0.5 and the other at threshold 0.75. The concepts on the DOT graphs correspond to the concepts up to the thresholds from the node listing in Figure 5.5 and Figure 5.6. We easily can see from the graphs that the larger threshold results in a larger sub-ontology.

As our goal was to have a scrubtable user model visualisation underpinned by a light-weight ontology, this implied we would have to be able to visualise the light-weight ontology in the process. We intended to use SIV to visualise the user models, so a first step was to use it to visualise the ontology, and provide a solution to the *ontology interface problem*.

We modified Mecureo to generate OWL output (McGuinness and Harmelen 2003) and wrote a corresponding parser for SIV. There were a number of reasons why we chose OWL. Firstly, it was the preferred standard at the time we were deciding how to represent our ontologies (during 2003). We were anticipating that future ontologies would support an OWL serialisation, and this would mean that we could easily switch ontologies in LOSUM to ones created by other sources. The SIV visualisation only requires the URI of the OWL file to be able to visualise it, and could potentially use any arbitrary OWL file as input. This means that the ontology does not need to be generated by Mecureo. Serializing the ontology in OWL format also means that the output is accessible to other ontology tools, for example the ontology editor Protégé (Noy, Sintek et al. 2001).

Since then, SKOS (Miles and Brickley 2005) has become available as a stable standard for representing classification schemes and taxonomies. SKOS features much lighter-weight relationships than OWL, although they are expressive enough for the types of light-weight ontologies we are creating. This makes SKOS a more appropriate format for the representation of our ontologies. SIV has subsequently been enhanced to support SKOS serialisations; this is described in (9.4).
The first task in making use of OWL was to define a set of concept and relationship definitions in OWL that matches the Mecureo output relationship types from Table 5.2. The OWL header we created is shown in Figure 5.8. The relationships are specified in a configuration file in the Mecureo parser, so under a different configuration, the relationship names and properties would be different. For the studies described in this thesis, we have used the relationship types shown in Figure 5.8.
Each concept is then represented in OWL, as shown in Figure 5.9 which gives the representation of the concept *imperative language*. We can see the properties *hasSynonym*, *hasChild*, and *hasAntonym* from the Mecureo namespace shown in Figure 5.8 being used for related terms. These contain a resource attribute that is the URL describing the related term (the additional encoding such as the %20
is present to make the term names legal URL suffixes). For example, imperative language has a relationship to the concept declarative language. The relationship type is hasAntonym. So declarative language is an antonym of (i.e. dissimilar to) imperative language.

SIV is simply configured to parse the ontology specified in OWL, reading in the concepts and relationships to form the underlying graph structure for the data. This is shown in Figure 5.10 for the

```xml
<Concept rdf:ID="imperative%20language">
  <dc:Title>imperative language</dc:Title>
  <hasSynonym rdf:resource="#procedural%20language" />
  <hasChild rdf:resource="#DACTL" />
  <hasAntonym rdf:resource="#declarative%20language" />
</Concept>
```

Figure 5.9: The OWL representation for the term imperative language showing relationships to other terms.

Figure 5.10: SIV screenshot visualising the Mecureo generated ontology for declarative language.
concept *declarative language*. This can be navigated in the same way as discussed in Chapter 3 for focusing on a concept and exploring the ontological relationships.

We wrote an output to Mecureo to serialize the concept weightings in the format described for user model overlays from Chapter 4, and used this for both the colour and horizontal overlay.

In Figure 5.10 the colour and horizontal positioning specified in the overlays correspond to the distance weightings assigned by Mecureo. In this case, the weightings and node list are the same as the ones shown in Figure 5.6 for *declarative language* and the related concepts. The weightings are normalized to a value between 0 and 1, with declarative language having a value of 1 (most green) and terms going from green to red as the distance increases. The concept *algorithm* is the furthest away from *declarative language* and is given a value of 0 (and a colour of yellow in this case). All other concepts are normalized to have a value inside this range.

### 5.5 Refinement

Ontology refinement allows us to tune the resulting concepts and relationships at a finer granularity for the desired task. In this section we discuss a short-coming in the ontologies called the *restricted ontology problem*, and present a way to refine the ontology to address this problem and maintain scrutability.

**The Restricted Ontology Problem**

One of the weaknesses of the whole approach underlying Mecureo is that it guarantees the ontology will have nodes representing concepts only for the terms that appear in the chosen source dictionary or glossary. We call this the *restricted-ontology problem*.

There are a number of reasons why concepts that are needed for a user model may not be in a Mecureo-generated ontology:

- In a fast changing area like computing, we cannot expect that they will have all the concepts of importance for marking up learning objects.

- Technical and specialized dictionaries often omit core and common concepts that are widely understood. For example, one of the important concepts in our course is the needs of novice users and the usability techniques that are applicable to them. The concept, *novice*, does not appear in the Usability First glossary even though it is used in the definition of many other terms.
• For educational contexts an instructor might want to model additional pedagogical concepts such as assignment names or tutorial activities.

In a learning context, the second and third reasons are especially important. Such concepts will often be important for modelling student knowledge.

Evaluations of Mecureo (Apted, Kay et al. 2003; Apted, Kay et al. 2003) indicate that it should be possible to build an excellent foundation ontology from a specialised existing dictionary-like resource, such as Usability First (and FOLDOC, used in our earlier work). However, the restricted ontology problem means that we need a graceful means to enhance the ontology that is not only easy to do, but also maintains scrutability in a similar way that Mecureo-generated ontologies could simply refer back to the source glossary.

Local Definitions

For the approach in this thesis, the metadata annotation step (described in the next chapter) provides an ideal opportunity for users to identify such core concepts that are not in the ontology. However, we do not want make the task of metadata annotation require more effort by having users edit the ontology itself to add in the missing concepts and their relationships to other existing concepts. Our solution returns to the source of the ontology: the glossary with the definitions for the terms that make up the metadata vocabulary.

It would make sense that if a user wanted to add an additional concept into the ontology, that they would have knowledge of what it is about and what other concepts it would be related to. Therefore, it would make sense that rather than extending the ontology itself, users could define their own definition in a similar style to the definitions of the source glossary or dictionary, and leave Mecureo to extract the ontological structure. In effect, they create their own local dictionary definitions.

Earlier in this chapter, we discussed the process taken by Mecureo to generate a directed graph of the concepts in the dictionary by making each concept a node. It then scans through each of the definitions for concepts that are also nodes and generates a weighted link between them. The graph is gradually built up as more definitions are parsed until there are no more to do.

It would be a straightforward process for Mecureo to not only parse the source glossary, but also the local definitions defined by the user, and treat them as if they were just additional definitions in the source glossary. This means that relationships are created between the concepts defined by the user and concepts in the source glossary by Mecureo’s usual matching process. In some cases, even a minimal definition or even just the concept itself with no definition might be enough to have the new
concept sufficiently connected to the other concepts in the ontology. We explore this in the following section.

5.6 Applying the Ontology

By applying the ontology to real tasks, we can get a sense of the validity of the ontology and identify improvements and adjustments. This section describes two evaluations on the Mecureo ontology learning process. The first is an evaluation of the ontologies as visualised in SIV to gain insights into the quality of the ontologies and an initial test of the use of SIV to address the ontology interface problem. The second evaluation examines the approach to ontology refinement through the addition of local definitions. We follow this with an analysis of the ontology we have created for the UIDP course and address the fan-out problem by improving our local dictionary definitions.

5.6.1 Evaluation of Light-weight Ontology Visualisation

We designed the evaluation in the spirit of our scenarios from chapter 1: we wanted to create user tasks which reflect a situation where the user has provided a small amount of information about their preferences and they want to scrutinise the ontological inferences made from this. This task adds as a test of the interface provided by SIV to see this form of inference. This makes it a meaningful approach to gain insights into how SIV addresses the granularity problem. In addition, it evaluates the quality of the Mecureo-created ontology in two ways:

- Firstly, the ontology plays a critical role in structuring the graph over the user model: if users can make use of the SIV interface to explore the ontological interface on their user model, this indicates that the Mecureo-generated ontology addresses the ontology construction problem.

- Secondly, since the Mecureo-generated ontology is used for the ontological inference, the users’ response to this task will also give some informal evaluation of the quality of the ontological inference. It gives the opportunity for users to judge whether the inference seems plausible.

Task Design

Having developed ontologies that we could visualize in SIV, it was important to evaluate whether the interface and the ontologies being visualized were usable and made sense to real users. Essentially, we needed to assess whether Mecureo and SIV together do address the ontology construction problem and the ontology interface problem.
Experimental Design

We wanted to see if they could make sense of the idea that a user might provide a single piece of information and the system would make many inferences from this. We also wanted to assess whether they could use the SIV display to scrutinise the user model meaningfully. At the same time, this would also be a partial evaluation of the ontology. We designed a small scale qualitative experiment where each user was asked to explore two, quite different examples of point queries using SIV. These were Microsoft Word and Python, both graphs having about seventy concepts.

Users were presented with the scenario shown in Figure 5.11. This task was framed in terms of two hypothetical users so that participants would be less likely to judge the relevance of the task for them, personally. This meant that each user was doing the same task with the same goals. It clearly has the
disadvantage that users are not answering questions about themselves or exploring their own user models.

Questions A and D encouraged them to simply explore the display. We observed them doing this and encouraged them to think aloud, explaining what they were doing and what they thought. This also gave some insights into the user’s views about the SIV display and the ontological inference visible. Questions B and D encouraged participants to explore the idea of restricting the amount of inference while Questions C and D asked them consider making more extended inference. All of the Questions B to D also, and more importantly for us, assessed whether participants could make sense of such questions in this context. Questions E and F were mainly an exploration of the participants’ views of the whole enterprise; at the same time, it gave them more opportunities to talk about the task at hand and to indicate whether they appreciated just what was involved in the process.

Participants

We conducted the evaluation with 10 participants: five upper high school students attending a summer school in which they would learn to program in Python, and five undergraduate computer science students. These users are representative of a quite computer literate group.
Results

Table 5.3 summarises the responses to Question A. These responses and observations of the participants indicate that they could work out how to interpret the purpose of the display and could explore it. As noted by Participant 10, the interface is unfamiliar and novice users need to spend a few minutes trying it to become confident in understanding it. Similarly, Participants 1, 2, 5 disliked the clutter that is part of this approach to the visualisation but were able to explore the display effectively. For the central purposes of this evaluation, the critical thing is that the participants appeared to make sense of the task and were able to make use of the interface to explore the interests of the hypothetical James. Only Participant 8 described the display as confusing and Participant 3 needed more detail about the visualisation interface. Some users were not entirely happy with the ontology: Participant 2 hoped to find the concept Clipart and spent some time looking for it, without success; Participant 6 had expected to be able to see competing products.

Responses to Question B are summarised in Table 5.4. This required the participants to adjust the SIV display so that it imposed a tighter constraint on how closely linked the concepts should be to the initial query. This also indicates the participant’s ability to use and interpret the display. At the same

<table>
<thead>
<tr>
<th>User</th>
<th>Summary of responses for Question A:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Easy to use, good colour scheme, distinctly see what James wants through colours, don’t like overlap</td>
</tr>
<tr>
<td>2</td>
<td>Yes. Hard to read fonts. Some things are not there even though I think they are associated with Word, e.g. no &quot;Clipart&quot;</td>
</tr>
<tr>
<td>3</td>
<td>Slider labels are not clear. Its a good idea to be able to limit information. Why have both indenting and colours meaning the same thing?</td>
</tr>
<tr>
<td>4</td>
<td>Possibly a good thing, see everything directly on the screen.</td>
</tr>
<tr>
<td>5</td>
<td>Looks cramped, depth is not a good description. Don’t like black background.</td>
</tr>
<tr>
<td>6</td>
<td>Mostly - except that competing products are not mentioned (and not included in the sample?). Is user interested in Microsoft Word specifically, or is user after generic word processing solution?</td>
</tr>
<tr>
<td>7</td>
<td>Good to see other relevant topics. - it appears interesting that the company is doing a reasonable thing - the display on the left can user lighter colours.</td>
</tr>
<tr>
<td>8</td>
<td>No display is confusing. Personalisation idea okay.</td>
</tr>
<tr>
<td>9</td>
<td>Yes - an interesting and useful concept.</td>
</tr>
<tr>
<td>10</td>
<td>Haven’t seen anything like this around before. User may have difficulty in understanding the purpose of the system/interface.</td>
</tr>
</tbody>
</table>
time, since it restricts the number of user model components selected, it makes it easier for the user to focus on their accuracy. Participant #10 was the only one to consider that the smaller number of concepts was appropriate. All participants made sense of the task. As none commented on problems with the terms visible, this smaller display failed to disclose any problems there might have been in the ontology.

Table 5.5 shows the results for the opposite question. Interestingly, Participants #1, #2, #7

Table 5.4: Responses to question about a more restricted inference distance in the ontology

<table>
<thead>
<tr>
<th>User</th>
<th>Summary of responses for Question B:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>“See if you consider it would be better if the system were set at a smaller value, say 1?”</td>
</tr>
<tr>
<td>1</td>
<td>It’s too basic.</td>
</tr>
<tr>
<td>2</td>
<td>Not enough information.</td>
</tr>
<tr>
<td>3</td>
<td>I want to see other stuff there.</td>
</tr>
<tr>
<td>4</td>
<td>The number of concepts is too narrow.</td>
</tr>
<tr>
<td>5</td>
<td>Could miss out on words.</td>
</tr>
<tr>
<td>6</td>
<td>1 seems quicker to navigate - although a lot of depth is lost.</td>
</tr>
<tr>
<td>7</td>
<td>Not smaller.</td>
</tr>
<tr>
<td>8</td>
<td>Maybe.</td>
</tr>
<tr>
<td>9</td>
<td>No. Probably somewhere in the middle. Not too detailed but enough to get some ideas.</td>
</tr>
<tr>
<td>10</td>
<td>Yes, anything higher than 1 seems to be overloaded with information -&gt; user becomes confused.</td>
</tr>
</tbody>
</table>

Table 5.5: Responses to question about a more extended inference distance in the ontology.

<table>
<thead>
<tr>
<th>User</th>
<th>Summary of responses for Question C:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>“What about a larger value, say 3?”</td>
</tr>
<tr>
<td>1</td>
<td>Depth 4 or 5, more information.</td>
</tr>
<tr>
<td>2</td>
<td>Still missing stuff.</td>
</tr>
<tr>
<td>3</td>
<td>Overload of information, too many things not legible.</td>
</tr>
<tr>
<td>4</td>
<td>It’s too overwhelming to read.</td>
</tr>
<tr>
<td>5</td>
<td>Hard to read without moving the mouse over it.</td>
</tr>
<tr>
<td>6</td>
<td>Too busy.</td>
</tr>
<tr>
<td>7</td>
<td>Could be better to have larger value.</td>
</tr>
<tr>
<td>8</td>
<td>No.</td>
</tr>
<tr>
<td>9</td>
<td>3 is good. Not too large, not too much at a basic level that it isn't understood. Gives user a chance to think to cover a wider area for that interest.</td>
</tr>
<tr>
<td>10</td>
<td>No definitely no. Everything seems to be highlighted. It defeats the whole purpose of this interface - i.e. to assist the user with searching/browsing interested topics.</td>
</tr>
</tbody>
</table>
recommended an even more extensive inference distance because there were still concepts missing. It seems that this much of the ontology did not match all the elements they had hoped to see. At the same time, the participants’ ability to deal effectively with this question is additional support for the general plausibility of the ontology and the effectiveness of the scrutability support that SIV gives for exploring the inferred student model. Answers to Question D confirmed this general conclusion. Overall, the participants seemed to manage this task somewhat better and seemed to have higher approval of the ontology displayed for Riichiro and the Python-based inference.

The last two questions, concerning the general idea of making inferences and reducing the need for users to explicitly provide information, were consistently answered in favour of the value of inference. Responses are shown in Table 5.6 and Table 5.7. Only Participant #2 mentioned the need for user control. However, the participants appear to have presumed that the SIV display would be available to users like James and Riichiro to see their user models. Some participants expressed concern at the need for some training in the use of the interface.

Table 5.6: User assessments of the overall idea of inference

<table>
<thead>
<tr>
<th>User</th>
<th>Summary of responses for Question E:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>“What do you think of this overall idea of using the approach to save people having to give lots of information but then being able to use something like this tool to easily provide a richer set of things they want to know about?”</td>
</tr>
<tr>
<td>1</td>
<td>The screen is good to look at and use.</td>
</tr>
<tr>
<td>2</td>
<td>Good, help busy people who don’t have time, e.g. at school or at work.</td>
</tr>
<tr>
<td>3</td>
<td>Depth adding information is good, but can’t see the point of using the standard slider. Why not have vertical be the ranking instead? Or group concepts along the vertical?</td>
</tr>
<tr>
<td>4</td>
<td>Good idea, but company might miss out on a piece of information even though they can adjust it.</td>
</tr>
<tr>
<td>5</td>
<td>No response given.</td>
</tr>
<tr>
<td>6</td>
<td>Pretty cool, actually. I think I’d prefer a layout -&gt; [tree diagram]</td>
</tr>
<tr>
<td>7</td>
<td>It is better approach because it allows users to see related information and it would be easier for them to work from a centralised site.</td>
</tr>
<tr>
<td>8</td>
<td>User interface not intuitive enough. Wizard kind of setup would be easier to follow. Idea of helping them specify interests is good though.</td>
</tr>
<tr>
<td>9</td>
<td>It is a good idea. People also do less, which is good. Users generally hate filling in forms, this eliminates the use of that.</td>
</tr>
<tr>
<td>10</td>
<td>The idea is good but the interface is quite confusing at first with regard to its purpose. This is particularly the case for the left panel. If settings inappropriate it looks very messy and unreadable.</td>
</tr>
</tbody>
</table>
Overall, the experiment suggests that the participants considered the two point-ontologies plausible. Also, they could readily appreciate the general principle of the system, taking a single piece of user input and using that to generate a much larger user model. In addition, although several participants made negative comments about aspects of the interface, all appeared to be able to use it effectively for the experiment.

### 5.6.2 Evaluation of Local Definitions

We evaluate our approach of adding local definitions to refine the ontology and overcome the *restricted ontology problem* in this section. We hypothesise that adding in local definitions as an addition to the source glossary can create sufficient linkage to existing concepts in the ontology generated by Mecureo. Lengthier and more descriptive definitions should create better linkage.

#### Approach

The user-defined local dictionary definitions are stored in a file format that can be read by Mecureo. The parsing process was discussed in section 5.1.2. When the parser is run, the file containing the user...
defined definitions is merged with the dictionary and parsed by Mecureo to create the ontology graph. In this experiment, we use the Usability Glossary as our initial glossary source, and use local definitions in a set format. An example of a definition in this format is given in Error! Reference source not found..

We want to examine the amount of linkage based on the amount of information provided in the local dictionary file. We examine a number of concepts and their linkage to other concepts in the ontology based on how descriptive the entry is. We have divided this up into three categories; these are:

- **Concept name only**: We use just the concept name with no following definition.

- **Concept and Definition**: We provide a short definition for each concept. We do not force any related concepts using the chain brackets (such as in line 6 of Error! Reference source not found.). This means that links to other existing concepts can only be inferred from the words appearing in the definition.

- **Concept, Definition and Related**: We use two keywords (related concepts identified with the use of chain brackets, as in line 6 of Error! Reference source not found.), in addition to the definition as just described. This allowed us to ‘force’ a relationship between our new concept and one or more existing concepts in the dictionary. The other relationships have come from parsing the definition we provided for the concept, and the definitions in the concepts that Mecureo has found to relate to this concept.

We had three sets of local definitions based on the categories above, each with six different terms representing core concepts in the UIDP course: **novice users, discretionary users, casual users, exploration, usability technique, testing process**. We used the contents of the online lecture slide that demonstrate the corresponding concept as the definition as required. These terms and definitions sets

```
1 Exploration
2 [#exploration]
3 Simulate the way users explore and learn about an environment.
4
5 Related: {cognitive modeling} {learning curve}
6 Categories: <Usability Methods>
```

Figure 5.13: An entry for the concept exploration (declared on line 1). The second line is the URL identifier for the concept, followed by the definition and the related (existing) concepts in the dictionary and which categories this concept belongs to, respectively.
have been reproduced in Appendix A.

**Results**

Table 5.8 shows the results of this very lightweight approach. The columns correspond to the amount of information in the definition as described above.

<table>
<thead>
<tr>
<th>Concept</th>
<th>Concept name only</th>
<th>Concept and Definition</th>
<th>Concept, Definition and Related</th>
</tr>
</thead>
<tbody>
<tr>
<td>novice users</td>
<td>2</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Discretionary users</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>casual users</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Exploration</td>
<td>9</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>usability technique</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>testing process</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

The simplest way to add a new concept is to just use the concept name with no definition, although the linkage is minimal, especially when there is more than one word in the name. In the case of multi-word concepts, Mecureo fails to find enough (if any) related concepts, causing orphan nodes in the ontology graph. In the resulting ontology, the concepts *discretionary users*, *casual users* and *usability technique* are all orphan nodes.

Bootstrapping the parsing process by giving the pseudo-concept some existing related concepts and a short definition minimizes this effect and gives more favorable results. For the longer user defined concepts, the lower number of links occurs because of the text matching level in the parser (which controls things such as case-sensitivity and substring matching).

**5.6.3 Analysis of the Fan-out Problem in the HCI Ontology**

In this section we provide a more detailed discussion of the local definitions added to the ontology during the annotation process of the learning objects that teach the theoretical aspects of usability and user interface design in the UIDP course. There was total of 9 learning objects with 161 slides between them. We added a total of 105 local definitions to enhance the metadata vocabulary provided by the original source glossary. Table 5.9 shows a breakdown between the number of concepts defined locally and the ones used from the original source glossary.

| Table 5.9: Distribution of concepts and peers

115
Although more than half the concepts used were added as local definitions, we realised the connectivity of these concepts to other concepts in the ontology was very low, with an average of 3 relationships (or peers), compared with 9 for terms in the foundation glossary (last column of Table 5.9).

In Figure 5.14, we show an analysis of this distribution. It indicates that although the most terms have less than 20 peers, we still have around 80 concepts with no peerage at all and some outliers having over 100 peers. This extremity in peer distribution causes our visualisation to suffer from two problems:

![Figure 5.14: The distribution of terms in our ontology grouped by the number of peers (fan-out) they have. There are a significant number of terms having a low peerage or an extremely high peerage.](image-url)
• Concepts with too many peers or related terms, causing the visualisation to clutter. This makes it hard to see any related terms at all because there is too much overlap between the concepts.

• Concepts with no peers mean that when one of these concepts is clicked on, everything else on the visualisation is blurred out. This makes the simple task of navigating the ontology very difficult as the user must constantly backtrack when they encounter one of these concepts.

Both these extremities are part of the *fan-out problem*, where the variance in the number of relationships from a concept is high, causing problems for visualisation of the graph structure. In Chapter 4, we discussed approaches to address the *fan-out problem* for concepts with high peerage through visualisation-based filtering and heuristic methods. In this section we focus on approaches based on the ontology creation process to improve the connectivity of concepts with low peerage.

We believe visualisation will be most effective if each concept has around 5 to 20 peers. Accordingly, we focus on this region of Figure 5.14, as shown in Figure 5.15. Clearly, we have significant numbers of terms with very low peerage. In fact, 475 terms (almost 40%) have less than 5 peers. We observe that a lot of these low peerage concepts are due to the local definitions we have added, which had a low average number of peers as we saw in Table 5.9. This is particularly serious as we took action to

![Terms grouped by number of peers upto 20](image)

**Figure 5.15**: A close-up of Figure 5.14, with cut-off at 20 peers.
specifically add them to the dictionary because they are important concept terms for the course.

**Improving the peerage of added concepts**

A simple way to address the problem of low fan-out is to improve the definitions of the added terms. We can do this quite easily, because in our case, we have only added new terms with minimal descriptions.

Figure 5.16: Janus Term editor, for editing the glossary definitions. The concepts are categorized on the left by the number of peers as generated by Mecureo.
For the concepts we added to the ontology and subsequently as metadata to the UIDP course, we created Janus, as shown in Figure 5.16. Janus is an application that provides a list of the concepts in the dictionary text file grouped by the number of peers in the final Mecureo generated ontology as seen on the left-hand side. The green bars are a visual representation of the number of concepts that have a certain peerage. For example, in the screenshot, we can see that there are a majority of concepts with only one peer. The list of concepts that have 3 peers is currently expanded and links to the definitions are visible (in the screenshot, these are feedback, speed, storyboard, terminology, etc).

When a user clicks on one of the terms, it is displayed on the right-hand side in editable form fields, similar to the interface for adding a new term. Here the user can edit the definition and add in any additional details they want (such as an extended description or some hard-linked terms). In the screenshot we can see the user editing the concept exploratory learning.

Janus reads and outputs the definitions in the format required by Mecureo for parsing. After editing the definitions, we simply re-run Mecureo to generate a new ontology, and Janus will resort the list of concepts by their new peerage. It then becomes an iterative process of editing the definitions, and regenerating the ontology to examine the new peerage.

5.7 Summary

This chapter addressed our need for an ontology for the LOSUM system, where it contributes to a solution for the scrutable user model interface problem: the ontology provides a way to structure the user model for visualisation, and capitalise on the fact that the ontology provides a human understandable conceptualisation of the domain to directly support understandability of the user model. To achieve this, we have overcome a number of problems.

The first problem we dealt with was the ontology construction problem. We chose Mecureo to generate ontologies automatically from existing dictionary sources. We have discussed the processing of the dictionary, generation and querying of the ontology, and output serialisations in this chapter. This has been described in the context of a five step framework for ontology learning developed by Maedche and Staab (2001). In particular we want the ontology to remain scrutable to both the users of the system and the maintainers. Mecureo uses a text based dictionary as the input for generating ontologies, so users can always refer back to the original definitions.

We were also required to overcome the ontology interface problem, as we had a need to visualise the ontology in a way that could later be extended for overlay-based user model visualisation. We modified Mecureo accordingly to generate ontology output that was usable by SIV; the changes to SIV were described in Chapter 4. In particular, we used point queries to create smaller human-manageable sub-ontologies that addressed the requirements by SIV for visualisation.
The first evaluation in this chapter shows promise as SIV can readily visualise the raw ontologies. Extending it with an overlay of user model data as discussed in (4.4.2) should be feasible, as we have demonstrated the use of overlays for Mecureo weighting values in the evaluation. The evaluation also demonstrated the visualisation of ontological inference based on the relationships between the concepts in the ontology. It showed that the ontologies created were plausible for this sort of inference. Responses from the participants showed they appreciated the use of the interface for scrutability.

An issue we specifically addressed in this thesis was with a major shortcoming in the ontologies that we term the restricted ontology problem, whereby technical glossaries tended to lack some of the concepts needed for effective user modelling. We identified a need for a scrutable solution: this involved the manual addition of these terms back to the source glossary so they are incorporated into the ontology.

The second evaluation and analysis of the ontology we would use for the basis of our learner models showed that the approach to the restricted ontology problem provided feasible results, although a short coming we identified was the need for longer and more descriptive definitions to overcome the fan-out problem, to ensure that the added concepts exhibited sufficient connectivity to the other concepts in the ontology that made it suitable for visualisation. We subsequently addressed this issue in the analysis of the HCI ontology that became the eventual ontologies we would use for the learner models in the UIDP course.

In this chapter we have described the ontology refinement and parts of the application steps in the context of an ontology of HCI terms. This ontology forms the basis of the user models for the students in the UIDP course. In the next chapter, we discuss how we applied this ontology to provide a controlled vocabulary to annotate the learning objects, and also as a way to aid user cognition and discovery of metadata terms during the metadata annotation process.
Chapter 6

Metadata Annotation utilising Light-weight Ontologies

As this thesis explores the way light-weight ontologies can support scrutable user modelling, an important component of our work is the need to be able to associate domain content with the user model. This is achieved through the addition of metadata to define what the content is about so that it can be machine processed.

For hand-crafted metadata, we need to address the metadata annotation problem, that is, to provide an interface to make the task of metadata annotation easier and less tedious. Therefore, this chapter provides the following important contributions to our work:

- Firstly, we address the metadata annotation problem with a method that exploits light-weight ontologies to enhance user cognition of metadata terms so that a metadata author can create metadata more easily and effectively.

- We integrate our approaches into a tool called Metasaur. It is novel in that it integrates a visualisation tool (SIV) into the metadata annotation process, and is also a further demonstration of the flexibility of SIV to be incorporated into applications beyond its original purpose of user model visualisation.

We have highlighted in Figure 6.1 where Metasaur (in dark blue) fits in with the other components in LOSUM. We build upon the work in the previous chapters using the ontology to provide the terms for the metadata vocabulary and a SIV-based ontology visualisation as an integral part of the Metasaur interface to help users find relevant concepts that they might not have considered.
The Metadata Annotation Problem

Metadata tagging is a problem, especially in systems with many existing documents and a large metadata concept vocabulary (Thornely 1999). This task is challenging and non-trivial because it is hard to be thorough and consistent, and the task is both demanding and boring. The task becomes even harder when the documents might be multimedia objects such as an audio clip.

Since it is such a tedious task to add the metadata by hand, there is considerable appeal in finding ways to automate part of the process. Even in this case, there is likely to be a need for human checking and enhancing of the metadata. We need interfaces that can support both the checking of metadata, which was created automatically, as well as hand-crafting of metadata. We call this the metadata annotation problem.

In the case of e-learning environments, there are additional pedagogical issues involved. When a teacher creates a learning environment, this is purposeful, with each learning topic designed to support learning about course goals. There are potential benefits if the teacher can easily attach metadata to learning topics, indicating why they were included in the learning environment. Firstly, it means that the learning activities can feed into a learner model. In addition, it may help the teacher maintain clear links between intended learning outcomes and the choice of a learning object.
6.2 Ontologies for Metadata Annotation

Ontologies provide an important tool in allowing people to create metadata by providing a common vocabulary of concepts and relationships for a domain; a solution to the metadata vocabulary problem. The Semantic Web initiative has played a large role in increasing the general awareness and perception of the importance of metadata. Although there have been a recent interest in the visualisation of ontology-based metadata, such as the visualisation developed by Mutton and Golbeck (2003), we are currently unaware of any visualisations used in the metadata annotation process. Although the visualisation of ontologies is a challenge in itself, there is potential in extending ontology visualisations as a way to make the task of metadata annotation easier.

We believe the general structure of ontologies can be exploited to aid people in discovering related concepts when annotating content with metadata. This is because related concepts deserve consideration as potential metadata candidate concepts themselves. These might be concepts that are not directly mentioned in the content, and would require a metadata author to spend time thinking about them.

For example, a learning resource might discuss the concept cognitive walkthrough, describing the process of how to “think” like a user in the process of evaluating an early prototype interface. An instructor annotates the content with the corresponding concept cognitive walkthrough. They then examine ontology and examine other concepts deemed related through ontological relationships. They settle on one such concept, mental model, and decide it is indirectly taught in the learning content, so they annotate the learning content with that concept as well.

An additional benefit from the use of the ontology in an interface for metadata annotation is that it also allows users to verify the relationships and concepts as they search and explore it. From the example above, if the user noticed concepts related to cognitive walkthrough that they were unsure should have been there, they can always check the original glossary that was the source of the ontology and scrutinise why the relationship was there. Thus it provides an inexpensive way for human checking of the automatically-generated ontology. This is important, as the process of ontology maintenance is usually a costly process when dealing with automatically generated ontologies.

6.3 Metasaur

We now describe Metasaur, a novel interface that makes use of SIV to support the creation of metadata for domain content. SIV has been incorporated in the role of an ontology visualisation, and can provide the aforementioned benefits of utilising an ontology during the metadata creation process.
Although the configuration of Metasaur described in this thesis is in the context of the UIDP course, it can be generalised to any domain as an approach to ontology-supported metadata annotation.

**User View of Metasaur**

Metasaur is linked to the UIDP domain as described in Chapter 3, and utilises an ontology that has been generated using Mecureo, as described in Chapter 5. It incorporates a number of elements that make up the final interface, designed for the markup of learning objects in the UIDP course, as shown in Figure 6.2:

- The SIV visualisation takes up the left-hand side of the interface. It provides a means to be for navigation of the ontology during the process of metadata annotation. Users can navigate the ontology by clicking on concepts, and also adjust the **Term Expansion** on the top bar, controlling the number of concepts visible on the visualisation. We can see in Figure 6.2 that the Term Expansion is set to the second lowest level and the concept *think-aloud protocol* is in focus.

- The **Select Topic** box on the top right-hand side allows users to navigate through the available learning objects. These are the topics that learners have to learn as they progress through the course, and were created by the course instructor. In the screenshot, the currently selected topic is *Empirical Usability*.

- The middle box on the right-hand side contains a slide from the learning topic, with the associated audio. This slide is identical to that presented on the course website where learners access it. The slide *Empirical techniques* is shown in Figure 6.2.

- A bar underneath the slide allows users to find metadata terms in the ontology. Users can either type in a term and click **Search**, or they can highlight text in the slide content itself and click **Search Selected**. Being able to highlight text and then search allows users to quickly use terms without the need to manually type in search queries.

- The box on the bottom right-hand side shows the metadata terms already associated with the slide. In Figure 6.2, the concept *think-aloud protocol* is associated with the slide *Empirical techniques*. Users can click on any term in this box to put it in focus on the visualisation. This allows users to quickly see terms related to ones they have already added, without the need to manually search for them again.

Essentially, this interface makes it very easy for the use to make use of several sources of ontology search term: the learning resource itself; the terms the user may think of; and existing metadata terms.
Once a term has become the focus of the display at the left, the user can readily see the terms most closely related to it in the ontology. These can then help the user consider additional terms as metadata candidates. All elements of the display can be directly applied in any context where the course has a set of learning topics and web-based resources.

**Walkthrough of Metasaur usage to annotate a learning object**

This section provides a walkthrough demonstrating the typical interactions with Metasaur for annotating a slide with metadata and using the ontology to aid in the discovery of concepts that are not directly mentioned on the learning content.

Suppose that a user, Tanya, wishes to annotate the slide titled *Empirical techniques*, as shown in Figure 6.2. We can see it already has the concept *think aloud protocol* as a metadata term. She reads the bullet point “Observation of natural use” and highlights the word *observation*. She then clicks on...
the Search Selected button to perform the search. Results are shown on the visualisation on the left hand side of the interface. We can see the results observational study, observation room, participant observation and random observation shown in Figure 6.3.

Tanya now scans through the search results, deciding that the concept observational study is appropriate to describe the slide. She clicks on the concept observational study in the visualisation, and it changes to show that it is in focus. Tanya then clicks on the Add Metadata Element button. A popup asks for confirmation before the concept is associated with the slide (Figure 6.5). The concept is added to the slide’s metadata list (Figure 6.5).
Tanya scans the other visible, and so, related, concepts on the visualisation to see what else might be appropriate to describe the slide. Figure 6.6 shows a close up of a portion of the ontology visualisation. Looking at the left-hand side close up, we can see a number of visible concepts such as *ethnography*, *experiment*, *inter-observer reliability*, and so on. Tanya scans through these visible
concepts and decides she wants to add the concept *ethnography* (near the top of the left-hand side close up). She selects it by clicking on it. It becomes the focus concept in the visualisation, and we can see from the right-hand side close up in Figure 6.6 that the concepts have had their font size and spacing changed to represent this.

The concept is then added in the same way as the previous one by clicking on **Add Metadata Element**. The slide now has three concepts associated with it: *think aloud protocol*, *observational study* and *ethnography*. Tanya can now proceed to associate more metadata with the slide in either of the two approaches she has just used: by selecting or manually searching for keywords on the slide content, or by exploring the ontology through the visualisation. At a later stage, Tanya can also click on the metadata terms associated with the slide to put them back into focus on the visualisation and see their related concepts in the ontology again.

When she is happy with the terms added to this slide, she can easily navigate to the previous or next slide in the learning object.

### 6.4 Metasaur Architecture

We now provide an internal description of Metasaur and discuss further how it integrates with the other components in LOSUM. An overview of the structure of Metasaur is shown as a data flow diagram in Figure 6.7.

At the top is a data source for the existing dictionary. To generate ontology for the UIDP domain, we utilised the freely available, online Usability First glossary of HCI terms\(^{21}\). A screenshot of the

The glossary is shown in Figure 6.8 showing the concept *cognitive walkthrough*. Each concept in the glossary has some pre-defined relationships (under the headings *see also*, *categories* and *related links*) in addition to the textual description.

Mecureo parses the glossary and generates a light-weight ontology in the same process as described in Chapter 5. This ontology is shown in the second data source from the top in Figure 6.7. The resulting ontology contained 1129 concepts and 10345 relationships between them.

The ontology is then parsed by SIV to give a visualisation of the ontology. The SIV interface is integrated into Metasaur so users can access it alongside the learning content. In Figure 6.7, the teacher (at the bottom) can use it to explore the vocabulary as they use the interface to add metadata. Metasaur generates metadata that is shown as the right most data store in the diagram.

It is of note that at this stage, Metasaur is designed purely to support the addition of metadata to the learning content. It does not allow users to add new metadata such as *author* or *dates*. This is because we are primarily interested in the annotation of the domain content that we can link back to the domain ontology, and later exploit to relate it to learner models. Although metadata information such as *author* of the learning content is relevant in a pedagogical context, it does not play a role in defining the components of our eventual learner models for the purpose of this thesis.
Evaluation of Metasaur

As the goal of this thesis is to exploit light-weight ontologies in the process of creating scrutable user models, we performed a qualitative evaluation of Metasaur to see how the addition of the light-weight ontology in the metadata annotation process would be used by instructors, who would be the eventual users of such a tool. The hypotheses we aimed to address were:

- Would SIV enable users to be able to quickly find and add metadata to learning content from a controlled vocabulary (the ontology)?

- And secondly, could SIV assist users in finding concepts that did not directly appear on the learning content but could be found through exploration of the ontology?
6.5.1 Experimental Design

Context

The Metasaur interface was configured as in the description and example given in 6.3 for the UIDP course. The ontology was generated with Mecureo from the Usability First glossary. There were two slight differences from the interface:

- The first was the omission of the audio data, as our goals were to explore the usage of SIV in the metadata annotation process to find concepts based on the visible slide content. Including audio content would have slowed the process and been more demanding on users with no major benefit for the evaluation of our hypothesis.

- The second was the addition of a Back button to allow users to go back to the previous concept that was in focus in the ontology. This was identical to the Back functionality in the original VIUM action menu, and provides a way for users to back-track through their exploration of the ontology.

Participants

Participants were seven people who had tutored courses for the School of Information Technologies at the University of Sydney. We have designated the participants with the letters A to G. Participants B, D and E had previously used SIV but not as part of Metasaur. Participants B, D, F and G had familiarity with the UIDP course, being previous instructors and tutors. This familiarity with the domain content means that

<table>
<thead>
<tr>
<th>Participant</th>
<th>Previously used SIV</th>
<th>Familiarity with UIDP course</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>E</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>G</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
Task

The task our participants performed is akin to that of a librarian cataloguing a book. In both cases, the participants or librarians are not reading the whole book (or listening to the audio on the slide).

The instructions in Figure 6.9 were read aloud to each participant, with references to the interface for each particular element described. The participants were asked to annotate a single set of lecture slides about an Introduction to empirical usability by reading each of the slides in the lecture, and finding appropriate metadata terms. The participants had to find concepts in the visualisation that best describe what the slide was about.

Monitoring and logging was used to collect click stream data. We were interested in the type of actions participants performed as they went through and annotated the slides:

- The number of concepts they added. We are interested in both concepts added as a result of being directly mentioned on the slide contents, and also ones which the participants found through exploration of the ontology.

Annotate the lecture slides with metadata concepts. Choose concepts that best describe the slide from the glossary. If you can't find any, then just go to the next slide. There are only 8 slides. The first slide is the title.

A Glossary has been provided on the left hand frame to help you.

You can search the glossary from the search form on the right.

You can select some text on the slide and click "Search Selected" to search as well.

The display on the left shows a list of concepts you can add. You can select a concept by clicking on it. Related concepts are shown in progressively larger fonts.

You can increase or decrease the number of concepts on the display using the depth slider.

When you find a concept you want to add, click the concept labelled “Add Metadata Element”.

Click "finish" down the bottom when you are done.

Figure 6.9: Instruction for participants to annotate a series of slide with metadata.
• Usage of the SIV to aid in the metadata annotation task:

  o The number of clicks on concepts in SIV, as an indication of their amount of usage and exploration of the ontology.

  o The number of times they clicked Back to go back to a previous concept that was in focus in SIV.

  o The number of searches performed, both manually input or via the highlight and search method.

  o The number of times they changed the Term Expansion to change the number of concepts visible in the visualisation.

  o The number of “click conversions”, that is, the number of concepts that the participant added that was a result of following relationships in the ontology rather than as the result of a search.

6.5.2 Results

Table 1 shows usage by each participant. It has the time taken for them to annotate all the slides and the interactions described in the experimental design. These are: the number of concepts each user added, the number of clicks (on concepts in the visualisation, on the back button, on the search button, and the depth buttons), and also the number of ‘click conversions’. The table has been ordered by the number of concepts added as metadata.

Consider the case of the participant F on the first row; they took nearly 35 minutes to annotate all the

<table>
<thead>
<tr>
<th>Subject</th>
<th>Total Time (min)</th>
<th>No. Concepts</th>
<th>No. Clicks</th>
<th>No. Back</th>
<th>No. Search</th>
<th>No. Depth</th>
<th>Click Conversions</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>34.37</td>
<td>43</td>
<td>20</td>
<td>0</td>
<td>84</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>G</td>
<td>22.85</td>
<td>36</td>
<td>15</td>
<td>2</td>
<td>55</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>26.07</td>
<td>32</td>
<td>12</td>
<td>9</td>
<td>42</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>20.85</td>
<td>19</td>
<td>8</td>
<td>1</td>
<td>55</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>11.88</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>38</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>10.77</td>
<td>13</td>
<td>3</td>
<td>4</td>
<td>34</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>17.63</td>
<td>11</td>
<td>1</td>
<td>0</td>
<td>26</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 6.2: Summary of Metasaur Evaluation Logs
A total of 14 metadata concepts were added. Eighty-four searches were made (by either typing in a concept and clicking on the button, or by highlighting text on the slide and using Search Selected), and they did not use the Back function to backtrack through the concepts they had selected in the visualisation. The participant clicked on a total of 20 concepts on the SIV display (not including search results) and the final column, Click Conversions, shows they added two concepts from these clicks.

It is interesting to note that the participants who added the most concepts (F, G and D) were all familiar with the domain. They were also the only participants who made use of the ontology to discover concepts by navigating the ontology, as seen in the ‘click conversion’ column.

A more in-depth analysis of some slides was carried out to determine whether the concepts added by participants reflected what was on the slides. This analysis was done on the contents of slides 4 and 5, which are shown in Figure 6.10 and Figure 6.11.
The concepts added by the participants have been tabulated in Table 6.3 and Table 6.4. Users who were familiar with the domain (B, D, F and G) have are designated with an additional star (*) after their letter in the header row for each table.

The results in the tables have been divided into three sections:
Yellow for concepts that appear on the slide. For example, on Slide 5, the concept *speed* appears in the first bullet point of the slide, and so the row is coloured yellow in the results in Table 6.4.

Green for concepts that are a near match with concepts on the slide. For example, in Slide 4, the bullet point “Observation of natural use” is synonymous with the concept *observational study* – a near match to what is written on the slide, so this has been shown in green on Table 6.3.

Blue for concepts not on the slide. For example, in Slide 4, the concept time-stamping was added by participant D in Table 6.3, even though it is not explicitly mentioned on the bullet points of the slide.

**6.5.3 Summary**

All participants made extensive use of the text searching facilities and added on average 1 to 3 concepts per slide. However, one significant problem was that not all the participants were familiar with the domain, and were hesitant to add words when they did not understand the exact meaning (even if it was a word on the slide). But overall, every user could find appropriate terms in the ontology and made extensive use of the search features as well as exploring the ontology, suggesting there is merit in the inclusion of SIV (and light-weight ontologies) in the metadata annotation process.

The results indicate that only three users (D, F, and G) made use of the exploratory design of the visualisation to find new concepts (the Click Conversions column in Table 1). They represent three of the four users who were familiar with the user interface design and programming domain. These three users also tended to add more metadata concepts than the other participants. This suggests that the other users needed more time to gain familiarity with the interface and also a better understanding of the domain and task.

When the participants could not find concepts in the vocabulary that directly appeared on the slide, they had to rely on their understanding of the domain to infer what the slide was about. Table 6.3 shows the participants who understood the domain (designated with the * in the header row) tended to add more near match concepts to the slide compared to the other users. This addresses the second hypothesis that participants could find concepts not appearing directly on the slide, though there seems to be a correlation of this with their familiarity with the course. This is understandable. Only those participants who had familiarity with the domain could confidently conclude which concepts really were relevant to the overall concepts taught in each slide.
It is interesting to note that in Table 6.4, the three participants who were not familiar with the domain did not add many (if any) concepts that were matches to ones in the ontology. Those who were familiar with the domain added more concepts on average for this slide in addition to direct matches.

### 6.5.4 Metasaur support for local definitions

In the previous chapter, we discussed the refining the ontology to address the *restricted ontology problem*. Our approach was based around the core idea of being able to augment the original glossary source with new terms and definitions that defined concepts that were not in the ontology. We identified that the metadata annotation stage was an ideal opportunity for this to occur, as it was the part of the process that the user would be thinking about concepts that were required specifically to mark-up the learning content. Therefore, we modified the Metasaur interface accordingly to accommodate the integration of new definitions for the source glossary (and subsequently the ontology). Figure 6.12 shows a data flow diagram of this approach: the Metasaur architecture diagram as described in 6.3 has been enhanced with support for adding local definitions.

The interface contains two additional panels on the bottom right (Figure 6.13). The first panel allows

![Diagram](image-url)
users to add pre-defined local definitions to the slide as metadata. The second panel under this has a button users can click if the concept they wish to add does not exist in the ontology and has not been previously defined. The pop-up box in the middle shows a form the user is asked to fill in to add a new local definition (after clicking on the Add New Dictionary Definition button). Only the concept name and category are required. However a more complete linking to other concepts in the ontology will occur if they provide a definition as discussed previously.

Figure 6.13: The revised Metasaur interface, with an added section for users to add their own dictionary definitions on the bottom right. The user has filled in a new local definition of the word exploration in the pop-up box.
We used this enhanced interface to create our own local definitions as we marked up the learning objects for the course. An analysis of the resulting ontology and definitions was provided in the previous chapter.

6.6 Summary

Metadata is an important component in our system to be able to create scrutable user models as it forms a link between the user model, domain content and ontology. This chapter has addressed a number of important issues in the process to annotate content with metadata:

- We addressed the metadata annotation problem with a method that exploits light-weight ontologies and visualisation in the annotation process. By integrating an ontology visualisation interface into the metadata annotation tool, users can immediately exploit the ontology structure to enhance cognition of metadata terms.

- We integrate our approaches into a tool called Metasaur. This demonstrated the practicality of our approach, and after a user trial, it was used to annotate 9 learning objects from the UIDP course. Metasaur provides both a typical text input search interface in addition to pure mouse-based exploration of the ontology and annotation of the content.

In the next chapter we address the issues of using the learning objects as evidence sources for the concepts we annotated them using Metasaur, and a method to infer about concepts that were not directly used as metadata on the learning objects.
Chapter 7

Scrutable User Model Maintenance

One of the valuable ways that light-weight ontologies can support scrutable user modelling is in the creation and maintenance of the user model, as well as providing a means to make ontological inferences within the model. There are a number of challenges that present themselves, as shown in Table 1.2. The first is the user model definition problem, where we require a suitable set of concepts to form components in the user model. We addressed this in the previous chapter by using the subset of concepts used as metadata as a basis for the components in the user model, as they provide coverage of the concepts taught in the course. We must then address the user model maintenance problem, as the maintenance of user models based on usage of existing systems is a challenging task, with the following sub-problems:

- The evidence normalization problem: define how the available data from the different types of evidence sources contribute to concepts in the user model, addressing the issue of varying types and amounts of evidence from the evidence sources;

- The evidence combination problem: combine available evidence, dealing with the issues of varying reliability and number of evidence sources, for a single component concept;

We must also deal with the granularity problem: as we want to be able to reason about the high level coarse grained concepts, inferring about them based on the evidence supporting fine grained and more specialised concepts, as well as rather weaker inference from coarse grained across fine grained specialization relationships.

We address these issues and present our approach in this chapter to build scrutable user models. Our overall goals are to be able to provide a process that can be applied to large user models, but maintains scrutability. We utilise a comparative measure to compare and normalise evidence across different evidence sources, and a scrutable technique to combine evidence in such a way that it can be adapted to different learning styles. Finally we present an approach we call onto-increment to infer about concepts with little or no evidence in the learner model.
We demonstrate our solutions with an implementation using LOSUM components as shown highlighted in blue in Figure 7.1. This chapter is primarily concerned with the interactions between the user model, evidence sources, and the interface to the user model – we implement our evidence normalisation and combination approaches in this component. We address the granularity problem in SIV as it reads the user model data from the server. As demonstrated in the previous chapters, SIV exploits the underlying ontology in its visualisation, and the ontology is used again in the granularity reasoning process. We also report evaluation of our approach in this chapter. This is based on a qualitative usability study, where users demonstrated good, intuitive understanding of the student model visualisation with system inferences.

We present our solution in the context of a small example learner model in Figure 7.2. Although we are dealing with large user models for the goals of this thesis, this example learner model can be considered as a portion of a larger model, and our techniques extensible to them.

The learner model shown in Figure 7.2 illustrates how evidence feeds mainly into the fine-grain concepts. It represents a small hierarchy of concepts that form part of a domain model for a course in Usability, and evidence comes in the form of web log data and tutorial marks. In this case, the overall learning goal is the knowledge of both predictive and empirical usability evaluation techniques. A course instructor has associated the evidence sources with concepts in the domain model through metadata.

We can see that evidence may feed into a single concept (E1, E2, E3 and E5) or multiple (E4). Evidence may also feed into higher level (non-leaf) concepts of the ontology (E4). There is also a non-uniform distribution of evidence sources between the concepts (E1, E3 and E5 are from web log
The amount of evidence and its reliability may also vary, and in addition, it may also vary between different learners. The higher level concepts, *Usability* and *Predictive* have no direct evidence sources. This is typical of the data available from blended learning environments, where the high level coarse concepts represent overall learning goals or broad topic areas, but the majority of evidence contributes to fine grain concepts (De Bra and Calvi 1998).

We now discuss each of the problems associated with the user model creation problem: the evidence normalisation problem, the evidence combination problem, and the granularity problem, and our approach to overcoming each of these in more detail. In our description, we will use the student model shown in Figure 7.2 with two types of evidence: the amount of time students spent listening to audio for online learning objects mined from web log data, and the marks they received for weekly tutorial sessions.

### 7.1 The evidence normalisation problem

Each evidence source contributes evidence data to one or more concepts in the learner model, as shown in Figure 7.2. Our goal is to be able to consider the list of evidence from a single source and turn it into a normalized value that we can then use to compare with the values for other concepts.

For example, in Figure 1.2, we can have a single value to represent the accumulated evidence from the Web Log Data source E1 for the concept *Cognitive Walkthrough*. Similarly, we want a single value from the Tutorial Mark source E2 for the concept *Heuristic Evaluation*. However, not only are each of these evidence sources different from each other, but the amount and nature of evidence they contribute to the learner model also varies.

This introduces an important issue: we have to deal with the problem of varying quality of evidence from different sources and varying amounts of evidence.
7.1.1 The Student Standard

We introduce the notion of the Student Standard as a way to normalize the varying amount and quality of evidence, and represent it as another learner model that the system can use to compare against. Using a comparison to the Student Standard learner model we end up with a relative measure rather than an absolute one, reducing the effect of the varying amounts of evidence for the concepts.

This is similar to work done in the past where student behaviour is compared against the gold standard, for example the expert overlays in genetic graphs (Goldstein 1982). The difference is that in our system there is no single Student Standard model, rather one or more models the teacher considers meaningful. The teacher may choose a Student Standard that models the perfect student, or make available several Student Standards that match different learning goals such as achieving a bare pass or gaining advanced knowledge. In a case where several Student Standards are available, it would be appropriate for students to decide their own goal and choose the appropriate Student Standard for themselves.

In Figure 7.2 we can consider the case of a “bare-pass” standard where the student is not required to visit the web pages for Cognitive Walkthrough (highlighted with a bold border in the figure), whereas an “advanced student” standard does. In the following examples we show the approach we chose to normalize audio and tutorial evidence. We take the Student Standard as the student who attains full marks in the tutorials and listens to all the audio on the online lectures.

7.1.2 Normalising Audio Evidence

For the audio evidence, the length of audio narrative for each slide is known. We assume the Student Standard learner model will contain evidence that represents a student who has listened to the full slide (and have an extra bit of leeway time for taking notes, etc). We can compare the length of time a user has spent on each slide to that of the Student Standard time, and assign a score based on this. The weightings we assign range from 0.0 to 1.0 and the breakdowns are shown in Table 7.1. A further analysis of the audio web log data can be found in Appendix B.

We have chosen these values for a number of reasons. Firstly, up to the 100% audio time range, the value increases almost linearly to the time spent by the student on the slide. Although we could have quantised the values further, we decided against this to keep the reasoning mechanism simpler. The Overheard weighting is slightly lower than the Full Heard. This is to account for the times when students have become distracted with other activities and have left the browser open. These values are, of course, customisable, and are just what we chose based on the behaviour of students and the
Table 7.1: Understanding of audio slides based on duration stayed.

<table>
<thead>
<tr>
<th>Interpretation of audio evidence and slide</th>
<th>Duration on slide as percentage of Student Standard Time</th>
<th>Weighting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seen</td>
<td>Student Time &lt; 10%</td>
<td>0.1</td>
</tr>
<tr>
<td>Partial Heard</td>
<td>10% &lt;= Student Time &lt; 80%</td>
<td>0.5</td>
</tr>
<tr>
<td>Student Standard</td>
<td>80% &lt;= Student Time &lt; 150%</td>
<td>1.0</td>
</tr>
<tr>
<td>Overheard</td>
<td>150% &lt;= Student Time</td>
<td>0.8</td>
</tr>
</tbody>
</table>

nature of the audio. A different course instructor might require students listen to all of the audio and stay on the slide for duration of 125% of the audio time before being able to score a weighting of 1.0.

The next task is to consider which pieces of evidence to include. For example, a student’s usage may generate five pieces of evidence from an evidence source about the concept Cognitive Walkthrough with values {0.1, 0.8, 1.0, 0.1, 0.1}, corresponding to one short visit (duration < 10% of audio time), one very long visit (duration > 150% of audio time), one visit that is approximately the audio time of the slide, and two further short visits. We may find that the mean of the values for this student may not give the most accurate result as there are several short duration values generating the weighting of 0.1. We also notice this student has actually achieved a score of 1.0 for one of their accesses to the audio slide. So instead we choose to consider the “best heard” value, in this case 1.0 as the value for Cognitive Walkthrough.

This process is repeated for all the audio evidence sources contributing to the concept, and then we find the mean these values to represent an aggregation of their knowledge of the concept across several sources of audio evidence, as all of these different sources teach different aspects of the concept. This results in a final value from 0 to 1.0; a perfect student will have listened to every slide as a Full Heard at least once, resulting in a value of 1.0 for the component. We call this the Normalised Audio Score.

7.1.3 Normalising Tutorial Evidence

For the tutorial evidence, the students receive a mark out of 10. A perfect student should get full marks for every tutorial in our course. So, in effect, a mark out of 10 is already a comparison against that of the Standard Student (and we can just take the percentage value as a score between 0.0 and 1.0). In a “bare pass” standard, the teacher might set the marks so that any mark above 5 out of 10 will result in a value of 1.0.
We can then simply calculate the mean all the values which are already between 0.0 and 1.0 to get a final value between 0.0 and 1.0 for the final value for tutorial evidence. We call this the Normalised Tutorial Score. A similar approach of taking a mean of the mean marks to create a normalised score for a concept is in the CourseVis system (Mazza 2004), an indication that this is an intuitive approach for combining this type of evidence.

7.1.4 Maintaining scrutability

In both these approaches for extracting and normalising audio and tutorial evidence, we have compared student results to a customisable Student Standard, and have also allowed flexibility in the combination of evidence values. In potential systems, these customisations can be made by both the student and the teacher to match appropriate learning goals. This is a way to provide scrutability: with the aid of a suitable interface, as the user modelling process can be changed by the learner to match their needs, and also by the teacher to still be able to provide a range of centrally created Student Standards.

7.2 The evidence combination problem

There is a large body of work on different methods to combine evidence in user models. Jameson (1996) provides an overview of the major techniques: Bayesian networks, Dempster-Shafer theory, and fuzzy logic. However these techniques are often not easy to explain to users or tune to individual needs.

One of the goals of this thesis is to provide a method to combine evidence in a manner that is scrutatable: a simple approach that is easy to explain, and allows users the flexibility to fine tune it to their needs. Therefore, we choose a non-standard reasoning method that is intuitive to support scrutability.

We expand upon the approach just discussed, showing the normalisation of audio and tutorial evidence. Our approach uses a simple formula to determine each evidence type’s contribution to the final score, taking account of the fact that some evidence sources have higher reliability than others.

\[
\text{Score} = k_1 \times (\text{Normalised Audio Score}) + k_2 \times (\text{Normalised Tutorial Score}) \text{ where } (k_1 + k_2) = 1.
\]

Based on an intuitive sense of the reliability, \(k_1\) has been set to 0.25, and \(k_2\) has been set to 0.75 when there is tutorial evidence. This weighting is to represent the fact that we would consider the tutorial evidence to be of higher reliability (as it is marked by a human tutor) compared to the audio data (which is an aggregation of website visits on the student’s own initiative). Again both these values

146
might vary depending on the instructor or nature of the course or evidence, in this case these values were chosen in respect to the reliability of the evidence and evidence sources. This formula can be easily generalised to any number of evidence sources. In this case, where we have $n$ evidence sources, we have a corresponding number of $kn$ weightings that add up to 1, where we use the $k$-value as a simple representation of evidence reliability.

Moreover, in our case of $k_1=0.25$ and $k_2=0.75$, this is a very simple weighting. It would be straightforward to tell a user that the tutorials are weighted as three times as reliable as the online lectures. A simple interface might allow a user to adjust this weight if they felt that was more appropriate for them. For example, a student may miss a tutorial session that teaches a concept, but subsequently catch up by listening to the audio lectures.

### 7.3 The granularity problem

The different granularity of concepts in the user model means that there is an imbalance of evidence source distribution among the concepts. In learning systems, this is most readily seen in the fact that learning material tends to center on the fine grained concepts. For example, in Figure 7.2, there is no direct evidence for the concept usability as evidence sources contribute to concepts of finer grain. Meanwhile, learning goals are formulated in terms of a smaller number of coarse grained higher level concepts. It is therefore important to be able to model both coarse and fine grain concepts in the learner model, so that learners can see their overall learning progress and goals through the coarse grained concepts, and also determine what elements of work contribute to these high level goals through the fine grained concepts (McCalla and Greer 1994).

We need to be able to model the user’s knowledge of higher level concepts, in a way that maintains scrutability. We also want an approach that exploits the underlying ontological structure, in an effect similar to spreading activation techniques (Salton and Buckley 1988), where we can propagate values from concepts with a large amount of evidence to a concept with little or no evidence.

One simple method to deal with this is to build a spanning tree from the leaf concepts (the fine grain) and recursively pass their values up to the parent concept till we reach the higher level coarse grain concept we want to reason about. At each stage when the values are passed up the tree till we reach the root concept we are inferring about, some calculations can be done to account for the distance from the course grain concept in the tree, as well as the amount or type of evidence. We now present a detailed description of one such approach we have devised.
The onto-increment approach exploits the relationships in the underlying light-weight ontology (though it could work on any graph structure) and applies a simple reasoning technique where information from a generated spanning tree is passed to the root concept (the concept we want to infer about). This information from child concepts is used to provide a ‘boost’ to the value of the concept we are inferring about.

An example of the boost provided is shown in Figure 7.3. In the case where the concept we want infer about has no evidence, then the boost from child concepts gives it a score. When the concept has a
value that is not 1.0, then a positive boost is added. When the concept has evidence that results in a score of 1.0, then no further boosting of the score is possible.

**Onto-increment Algorithm**

For a particular concept $v_a$, we take an average of the values of the child concept values $\{v_{a,1},..., v_{a,n}\}$. This value is then multiplied with $(1 - \text{value of root concept})$ and added to the value of the root concept to give a proportional boost, but always maintaining a value between 0 and 1. The lower the score of the root concept, the higher proportion of inference the value will take. Equation (2) summarises the onto-increment formula for a concept $v_a$ with $n$ related concepts, where $n \geq 1$. In the case of $n = 0$, $v_a' = v_a$ (i.e. there is no inferred contribution to the final score for this concept).

$$v_a' = v_a + (1 - v_a) \times \left( \frac{1}{n} \sum_{v_{a,i} \in v_a \_\text{child}} v_{a,i} \right) \quad \text{where} \quad v_a \_\text{child} = \{v_{a,1},...,v_{a,n}\}$$

Consider the example portion of a student model shown in Figure 7.2. We want to infer about concept *Predictive*. Assume the two related sub-concepts *Cognitive Walkthrough* and *Heuristic Evaluation* have values of 0.6 and 0.4 respectively, and *Predictive* has a value of 0.1. We substitute these values into formula (2) and arrive at the value 0.65 as the new value for Predictive – essentially the sub-concepts can boost the value based on the knowledge of the fine grain concepts (3 & 4).

$$v_{\text{predictive}}' = v_{\text{predictive}} + (1 - v_{\text{predictive}}) \times \frac{v_{\text{cognitive walkthrough}} + v_{\text{heuristic evaluation}}}{2}$$

$$v_{\text{predictive}}' = 0.1 + (1 - 0.1) \times \frac{0.6 + 0.4}{2} = 0.65$$

The onto-increment approach is designed to provide a simple method for reasoning across granularities. While not as complete as a formal probabilistic reasoning technique, it provides a simple way to propagate values across concepts in the user model that utilises the underlying light-weight ontology. A further extension of the algorithm could vary the onto-increment based on the link types.

**7.4 Evaluation**

This evaluation built upon the work done in creating light-weight ontologies (Chapter 5) and annotating the learning objects with metadata (Chapter 6). The goal of the evaluation was to assess whether the user models built as a result of the approaches we have discussed in this chapter were
reasonable. This includes the evidence gathering, processing into learner models, and the inference across granularities using SIV.

7.4.1 Context and Implementation

The domain ontology was automatically constructed by Mecureo from the online Usability First Glossary. This has 1,129 terms and categories. We augmented these with 105 additional definitions using the extended version of Metasaur with local definition support and Janus Term Editor, resulting in a total of 1,234 concepts and 10,690 relationships between them. We provided an analysis of this ontology in Chapter 5.

The metadata for the learning objects was created using Metasaur, based on the set of concepts in the domain ontology, associating these with each lecture-slide and tutorial. We annotated the first slide of each learning object with high level concepts, and omitted any hits to these pages when analysing the web log data as the slide itself only showed the title of the learning topic and the audio data did not actually include any learning content, just an introduction to the topic and the speaker. This meant that we could maintain a set of higher-level concepts with each set of online lectures.

Personis

We chose to use PersonisLite, a light-weight version of Personis (Kay, Kummerfeld et al. 2002) to fulfill the role of the user modelling repository in our system. This part of the system can be seen on the top right of the system diagram in Figure 7.1. We can see that the user modelling system utilises the system metadata to define the components in the user model, and the domain content generates evidence based on user actions or performance. It is the role of the user modelling server to output the relevant user model data to the user model visualisation.

Defining the user model and adding evidence

The subset of the ontology concepts used in the metadata automatically defined the components of the student model definition in the Personis user modelling representation containing a total of 190 concepts with 423 relationships between them. This means that only 190 of the 1,234 concepts in the augmented glossary were included in the user model because they were the only concepts to be used as metadata for the course. The tools that collected evidence from web accesses and tutorial performance were used to add evidence to each student's learner model.

Evidence normalisation and combination implementation

The reasoning methods were implemented as resolvers in Personis. Personis stores a collection of raw evidence for each component in the user model. Resolvers are used to interpret evidence at runtime to
conclude a value for a component. When a user modelling application wishes to get a value for a component, it will call the appropriate resolver.

Figure 7.4 shows an overview of the internal structure of Personis with our implemented resolvers. We have implemented the audio and tutorial normalisation approaches as separate resolvers (the yellow part of the diagram) and the evidence combination algorithm as another resolver (in the orange part of the diagram). We can call the evidence normalisation resolvers separately to conclude values based on just that type of evidence, or we can call the combined resolver which will automatically call the normalisation resolvers and output a combined result.

**Onto-increment reasoning implementation**

Inference was implemented at the visualisation end rather than as a resolver. A primary reason is that the inference is internal reasoning of the system. It can always be calculated as needed. Personis is used for the evidence coming from external observations and data about the user.

There were also efficiency concerns: the implementation of Personis did not run as a persistent server object but instead instantiated when a call was made to access a user model, meaning that it would
have to read the Mecureo-generated ontology file each time (accessible via a URL). On the other hand, SIV kept a runtime representation of the ontology so it was much more efficient to implement the algorithm on the client side simply as another overlay on the domain model. The implementation of the overlays was discussed in Chapter 4.

### Interface

The result of this process is available for the learner to scrutinise, with the SIV interface. Figure 7.5 has a screenshot and explanation of its elements. The 190 concepts displayed in the visualisation, and their colour gives an indication of the student score for that concept. The list of evidence contributing to the concept score is at the right – in this case there is no tutorial evidence, and the score for the concept is 0.86. The inferred evidence is determined using the averaging formula (2). A user view of this interface is presented in Chapter 3.

### 7.4.2 Experimental Design

We wanted to evaluate whether the resulting user models were reasonable, and if the approaches we used to process the evidence made sense to users. We used a think-aloud evaluation to gain qualitative results. Participants were six senior level undergraduate students. They made good candidates to judge
the quality of the user models as they were familiar with the domain and all had experience as teaching assistants, meaning that we could ask them to comment and compare models from a teacher perspective.

They were asked to take the role of tutors and were presented with the information sheet below. They were asked to think-aloud as they performed the task in Figure 7.6. In particular, we were interested in whether they:

- could use the interface and understand it;
- would consider the results of the inference reasonable;
- could see if the related concepts contribute to the reasoning.

Two pseudo-students, A and B, were created, both based on a real student at the middle of the class ranking in the User Interface Design and Programming course. They were identical, except that we removed the web data for student B for several lectures. In addition, we lowered the tutorial marks for

Students A and B have quite different competence for the User Interface Design and Programming course. The course coordinator has requested that students struggling in this area will be invited to attend an additional catch-up tutorial session.

As a tutor for the course, you want to see how well the students understand the concepts in the area of predictive usability, in particular the concepts cognitive modeling, heuristics and user interface guidelines.

You need to fill out a form to allow them to attend the tutorial session as there is a limited number of places.

Unfortunately there is little direct evidence for these concepts, though there are plenty of more specialized concepts (such as the fact they have listened to a lecture on cognitive walkthrough, which is a subtopic of cognitive modeling) with evidence that could contribute to their understanding of the concepts you are after.

You want to select these topics on the signup sheet (and maybe some additional ones) relating to this area of study and see what the system infers about the student’s knowledge.

Decide if Student A and/or Student B should attend the catch-up tutorial with a justification for why they should attend on the signup sheet.

Figure 7.6: The task description for the evaluation given to the participants.
student B below those of student A’s marks. Using SIV inference for course grain concepts, B’s scores were consistently lower than A’s.

Table 7.2: The inferred values for each concept

<table>
<thead>
<tr>
<th>Concept</th>
<th>Student A</th>
<th>Student B</th>
</tr>
</thead>
<tbody>
<tr>
<td>cognitive modeling</td>
<td>0.50</td>
<td>0.22</td>
</tr>
<tr>
<td>heuristics</td>
<td>0.87</td>
<td>0.23</td>
</tr>
<tr>
<td>user interface guidelines</td>
<td>0.62</td>
<td>0.33</td>
</tr>
</tbody>
</table>

For both pseudo students, A and B, cognitive modeling, heuristics, and user interface guidelines had no evidence; hence a resolved score of zero in the user model, resulting in bright red font and, as these had no evidence, they were at the far right in the visualisation. Table 7.2 shows the values for the three concepts after inference based on evidence for related concepts. Student A’s overall higher scores for fine grain concepts are also reflected in the inferred values.

7.4.3 Results

All the participants successfully completed the task in less than 10 minutes and from the results in Table 7.3, unanimously decided that student B should attend the extra tutorial session.

All participants started with the search tool to look for the topics and quickly correlated the colour of the topics with the degree of knowledge for the students. All participants based their judgment of student B's poorer understanding compared to student A as due to student B’s inferred scores all being lower.

Some pointed out, upon seeing student B’s user model, that student B was not as good as student A based simply on the distribution of the colours when the concepts were expanded. Participant 5 said for the concept user interface guidelines, “In this case, there are more greens for this topic for student A [than student B]”.

All participants indicated that the inferred values matched their expectations. Participant 1 selected cognitive modeling for student A and instantly said “Cognitive modeling comes up red. I infer because the other concepts are green”. For student B on the same topic, Participant 1 stated “cognitive modelling appears correct [coloured red], but I will infer to make sure”. These comments were made before the participants used the Infer button to see the inferred value.
Participants could also correlate the inferred value with the values for related concepts. For example, Participant 6 was asked if they could see why the inferred value for heuristics indicated that Student A knew this concept, to which they replied “I guess because all the related stuff is green”.

### 7.5 Summary

This chapter has presented our approach to address the user model creation problem; constructing individual and group user models from widely evidence sources. In this process we have addressed a number of problems:

- The evidence normalisation problem meant that we had to find a way to easily aggregate and compare varying amounts of evidence of different reliability for a concept. We used a relative measure against Student Standard to allow us to normalise the evidence.

- The evidence combination problem involved dealing with the issues of varying reliability and number of evidence sources, for a single component concept. We addressed this problem with a way to weight the evidence sources based on the reliability, but at the same time keeping the weightings scrutable to user’s own goals.

- The granularity problem arose when we wanted to reason about the high level coarse grained concepts, inferring about them based on the evidence supporting fine grained concepts. Our
approach involved a recursive onto-increment algorithm that passed values from the related concepts to the concept we were inferring about based on the structure of the underlying domain ontology.

We also presented a small user study. Based on the results of the study, the approach we propose for building our user models seems promising. This study contributes to the assessment of the other elements of the system:

- The fact that users could easily do this task and make no negative comments about the ontology lends some support to the plausibility of the Mecureo-generated ontology;

- Participants readily used the SIV interface to explore the user models without any noticeable difficulties; this supports the design of SIV as a tool for scrutinising user models;

- The relatively large user models consisting of 190 concepts show that the ontological structuring with the user model overlay was intuitive enough for the participants to understand based on their comments.

- Even though the differences between the user models lead to no ambiguity over which student required additional tutorials, the participants agreed that the inference seemed plausible given the pseudo-student’s level of knowledge.

Although the study was limited through the use of pseudo-student models and participants with a high level of domain knowledge, the next chapter (Chapter 8) presents a large scale evaluation that incorporates all the elements described in this chapter for building user models in an authentic learning environment with real students.
Chapter 8

Evaluation

This thesis explores how light-weight ontologies can be used to support scrutable user modeling, therefore we present a summative evaluation of our work, demonstrating that light-weight ontologies can be used in the process of constructing real user models, and secondly, they play a role in supporting the scrutability of the user model in an authentic context.

The evaluation presented in this chapter is in the context of the UIDP learning environment, as described in Chapter 3, collecting real data, from real students, from a semester long course to build learner models. We wanted to make use of the type of data that is commonly available in blended e-learning contexts, with a range of grain-size, purity and reliability. The user model creation process was supported with the tools described in earlier chapters which utilised light-weight ontologies.

Open learner models are a valuable asset in helping students master learning material, as indicated by the number of systems described in Chapter 2. We therefore conduct our evaluation in this field, and examine how LOSUM and in particular, SIV, can provide open learner models for reflection (Schon 1983). We are interested in how our system can aid students in thinking about their own learning, and how the learner model can aid them in directing their future learning.

We utilise the UI-SIV configuration of LOSUM (as described in Chapter 3) for the evaluation. The user models were constructed for each student in the course following the same process described in Chapter 7, using a domain ontology built from the Usability First glossary by Mecureo (as in Chapter 5) and the course content annotated with Metasaur (as in Chapter 6).

8.1 Hypotheses

The heart of the evaluation is based on the hypothesis that the learner models we have created can be used for reflection. Students should be able to make comments on their own learning and progress during the course, and direct their future learning from examining the learner model. This
demonstrates a practical aspect of the learner models we have created, and shows that the approach we have taken with light-weight ontologies can produce useful and usable learner models.

Indications of user’s understanding of the system and the user modeling process which, lead to reflection involve the following sub-hypothesis:

- **H1. Find concepts of importance**: Students should be able to find concepts that the learner model indicates they know and don’t know. This is also essential for reflection since it enables students to determine how well they are doing. This also evaluates the usefulness of the visualisation and the underlying light-weight ontology as a navigation tool to enable the users to find concepts of interest. The learner’s displayed models are broad, giving an overview. Students should be able to navigate and find particular concepts easily.

- **H2. Explore evidence**: This follows on from the last hypothesis. Students should be able to easily find evidence to support why the learner model believes their knowledge is at a particular level. This evidence points to actions that the learner can take to address problems: for example, if the problem is that they have not attended the online lectures, this should be evident and can easily be addressed. This hypothesis also directly assesses how well the concepts in the light-weight ontology match the course content and also the scrutability of our evidence processing.

- **H3. Use inference**: Students should be able to infer about learner model concepts which have little or no evidence. This is critical to overcoming the granularity problem, and evaluates both the underlying ontological structure of the learner models as well as the onto-increment algorithm for inference. This is also essential for reflection because we know of the limitations of the evidence available for building the learner models, both in this particular context and in broader uses of e-learning.

If, for example, a student wants to work on an assignment about *usability*, they would naturally want to see the overall assessment of their knowledge in this course-grained concept. To provide this, we need to apply ontological inference from all the fine-grained concepts that constitute aspects of usability so that evidence from these can contribute to a useful display of this part of the learner model. This is also part of the overview of the user model since this ontological inference makes it possible for coarse-grained concepts to provide an overview, or coarse-grained view of the learner’s knowledge. This complements the role of SIV in giving an overview of larger user models.

The following sections describe the experimental design, then the results, relating them to the above hypotheses.
8.2 Experimental Design

Previous evaluations described in Chapters 5, 6 and 7 reported several small scale qualitative studies. For our main evaluation of the hypothesis above, we wanted to design a larger scale evaluation in a context with authentic usage. Our evaluation involves a field-trial deployment of learner models built through LOSUM tools, with SIV as the interface, and analyse the primarily qualitative data for authentic usage patterns that indicate reflection.

We chose to do this in the UIDP course which was introduced in Chapter 3, along with UI-SIV, the configuration of LOSUM for this domain. The evaluation was conducted in the last weeks of Semester 1, 2005 (from the week starting 7th March to 6th June, with one mid-semester break), followed by one week of study vacation (stuvac) and then the final exam the week after. This would allow us to examine overall performance in the course based on the final exam mark compared to the interactions with UI-SIV during this period.

The overall use of UI-SIV was intended to support students in reflecting on their learning in the last lab class of the semester as a basis for planning relevant study for the final exam. The relevant events in the course at that time were:

- Week 12: Last week of set tutorial activities (with focus on the topic, GOMS).
- Week 13: Last week of tutorial sessions and contact with tutor.
- Week 14 (Stuvac): Study vacation for all subjects.
- Week 15 (Exam Week): The UIDP final exam held on Tuesday.

Week 13 provided an ideal time to conduct an evaluation, where the students could be asked to participate in an activity involving their learner model during their weekly face to face tutorial session. Running the evaluation at this time gave a context where learners had good reason to reflect on their learning. The final lab class was usually devoted to revision and this made a perfect context to provide a tool to support reflection: this was the time students should have been revising the course and reflecting on their performance with the upcoming study vacation and final exam.

Coupled with this, familiarity or exposure with the system in Week 13 would allow us to examine their use of UI-SIV in the stuvac week and if students choose use it to direct their final revision and study for the examination. This could provide insights into authentic use of UI-SIV as a reflective aid, from which students could plan their revision and study.
We created learner models of the students with evidence from the website access logs and marks in weekly tutorials, excluding the tutorial marks from Week 12 (so the concept GOMS and other related concepts would only show a small contribution from audio evidence). This would allow us to ask students questions about a concept that they should have known about, but with minimal evidence present in their learner model. In the stuvac week we would rebuild the models to include all tutorial evidence over the semester.

Due to the nature of the field study being run at an important time during the semester, the UI-SIV interface utilized a static image of the Personis-based learner models. We wanted to minimize potential risk of data corruption to the live learner models in case anything went wrong with UI-SIV during this lead up time to the final examination. One limitation, however, is that the system would not automatically update the learner models as the students visited concepts.

Also important is that the learner models only dealt with one half of the course, on interface design and usability. It did not include the half that dealt with programming aspects. This was due to our choice of domain ontology – we chose to build the ontology from a glossary of HCI concepts that did not include programming concepts such as JavaScript or HTML. The distinction between design and programming is very distinct in the structure of the online lectures. We envisioned that students would therefore use the learner model to plan their studies in the design and theoretical aspects of the UIDP course.

8.3 Participants

The participants were students enrolled in the UIDP course in 2005 at the University of Sydney. All were senior level undergraduate students with considerable technical skill. We realise that observations based upon this user population may not generalise to people who are technically less skilled, but it is the right audience for the learner models we had constructed.

At the same time, the earlier evaluation of the basic VIUM interface (Uther 2001) was performed with a broad range of users, varying in age and educational level as well as technical expertise. This meant that the basic visualisation interface evaluation was not the core of our evaluation. Rather, we focused on the ways that the learner models could be used to support reflection though the underlying ontology and reasoning methods presented in this thesis, and also constrained to make use of only the authentic and varied granularity and reliability data available.

8.4 Experimental Tasks

The evaluation was run as a field-trial; UI-SIV was introduced as an additional tool students could use straight off the UIDP course website. The study consisted of two main phases. The first involved
student use of UI-SIV in a self-paced, online tutorial in the Week 13 class. The second was free use of UI-SIV in the stuvac period, during preparation for the final examination.

The UI-SIV tutorial was linked from the student’s profile page of the course website. In the Week 13 class, tutors instructed students to follow the link and work through the tutorial. Participants were presented with their learner model based on the best evidence available at the end of the previous week and asked to run through a small tutorial on the system. The first screen, Figure 8.1, shows the UI-SIV model visualization of a student at the left and the instructions in the lower right.

The tutorial consisted of 10 small tasks (labelled task T0 to T9), introducing students to elements of the UI-SIV interface and collecting feedback on student’s understanding of the response to UI-SIV. A complete description of the tasks is provided in Appendix C. The tasks were primarily interface and model navigation, with two tasks asking for qualitative feedback. Table 8.1 shows the tutorial tasks and indicates how they map to the evaluation hypotheses. Several of the tutorial steps gave students the opportunity to reflect on modelled aspects that should have been of importance to them (Hypothesis 1).

T0 introduced the interface: students were asked to select their own learner model from the views available (Figure 8.1). T1 introduced the use of colour in SIV, green for concepts modelled as known
and red as not known. Students should be able to relate this coding to the system belief of their knowledge of the concept, and demonstrate this in later tasks that ask them to find concepts of importance. Finding concepts they know well or don’t know well is an initial step in being able to reflect on their knowledge (Hypothesis 1). T2 explained how concepts related to the focus one had larger font size and asked participants to find concepts related to the one currently in focus. T3 introduced Search and asked students to perform a search for GOMS. T4 asked them to examine the evidence for it (Hypothesis 2). Participants were introduced to the idea of comparative and group learner model views in T6 and T7, and asked to find concepts relating to those views. Finally, T8 explained how to infer knowledge for the concepts with little or no direct evidence: they were asked to use Infer on a concept that the system indicated they were performing poorly in due to low evidence (Hypothesis 3).

T5 and T9 involved qualitative answers. Participants were introduced to the Select/Deselect functionality in T5, being asked to select concepts the learner model showed they did not know well, and then they were asked to comment on whether the system scores and evidence were appropriate or not (Hypothesis 2). T9 asked for general comments about UI-SIV. This gave an opportunity for

<table>
<thead>
<tr>
<th>Table 8.1: Elements of evaluation related back to hypothesis</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UI-SIV Interaction</strong></td>
</tr>
<tr>
<td><strong>Interface and learner model navigation tasks</strong></td>
</tr>
<tr>
<td>T0 – Introduction to interface, asked to select learner model.</td>
</tr>
<tr>
<td>T1 – Introduction to colour, asked to find concepts known well.</td>
</tr>
<tr>
<td>T2 – Introduction to font size, asked to list some related concepts.</td>
</tr>
<tr>
<td>T3 – Introduction to search, asked to search for GOMS.</td>
</tr>
<tr>
<td>T4 – Introduction to evidence and examining concepts, asked to scrutinise GOMS.</td>
</tr>
<tr>
<td>T5 – Selection of concepts that the learner model believed they were not doing well in (part 1)</td>
</tr>
<tr>
<td>T6 – Change to comparative model of their learner model vs. the class average, asked to find concepts scoring below average.</td>
</tr>
<tr>
<td>T7 – Change to model for the class average, asked to find concepts class doing well in.</td>
</tr>
<tr>
<td>T8 – Introduction to infer, asked to infer about a concept not doing well in.</td>
</tr>
<tr>
<td><strong>Qualitative tasks</strong></td>
</tr>
<tr>
<td>T5 – Comments on accuracy of learner model for concepts selected (part 2)</td>
</tr>
<tr>
<td>T9 – General comments.</td>
</tr>
<tr>
<td><strong>Other exploration during tutorial tasks</strong></td>
</tr>
<tr>
<td><strong>Stuvac free use</strong></td>
</tr>
</tbody>
</table>

162
comment on the learner model, its usefulness and anything else that the participants wanted to mention.

Two other elements of the evaluation relate to the student’s independent use of UI-SIV. These are shown at the bottom of Table 8.1. The first relates to exploration within the time of the tutorial but outside the set tasks set. For example, a student may have become interested in a particular part of the model and diverted from the tutorial to follow this. Equally, some students do not follow instructions and simply prefer to explore a new interface independently. In such cases, there was potential for us to gather evidence towards assessment of all three hypotheses, and overall assessing the main hypothesis that UI-SIV aided students in reflection.

A second source of information about UI-SIV comes from use during the stuvac week. Importantly, students may have incorporated use of UI-SIV in preparing for their examinations. For those who completed the tutorial, this is evidence that they considered UI-SIV sufficiently useful to return to it in their study for the exam. Other students may have noticed the link on their profile page and explored it. As shown in Table 8.1, stuvac use had the potential to provide evidence about all the sub-hypothesis and also our main hypothesis.

We included monitoring in UI-SIV. The logs during the tutorial and stuvac were subsequently processed to get the raw results for the elements listed in Table 8.1. Each participant’s behaviour and answers to the UI-SIV tutorial was extracted and compared with an additional source of evidence, the final exam mark, to find correlations.

Essentially, our analysis of the student activity was qualitative. We were able to see the model as the students saw it at any point in their activity and we could use the log of their activity to retract their footsteps. From this series of observations, we compiled detailed observation summaries of each student’s activity, noting the interesting features and classifying the activity. We further expanded this into a case study of participants who showed authentic usage of the system during the stuvac free use period, outside of the tutorial environment.

8.5 Results

We first report the numbers of students who used UI-SIV and the context of that use. As shown in the table below, this defines four populations of students who used UI-SIV.

The right-hand side column of Table 8.2 shows that, in a cohort of 114 students, 23 only did the tutorial, 40 only used UI-SIV in stuvac and 14 students did both. The overall population splits into three similar sized groups: those who did the tutorial; those that did not but did use UI-SIV in stuvac; and those who never used UI-SIV. We attribute the small proportion of students actually working
through the tutorial activity to the fact there were other class activities taking place during Week 13 (students were supposed to do sample exam questions as well as a group activity on the topic cognitive walkthrough). The UI-SIV tutorial was listed as the last activity, and many students did not have time to complete the tasks in Week 13.

**Overall Observations**

The bar chart of Figure 8.1 shows the distribution of these groups of UI-SIV usage across the range of marks. In general, across all mark categories, there is a similar distribution of UI-SIV usage with the majority accessing their learner models (during the tutorial and/or stuvac) in all mark ranges except
for the lowest band (20-29).

We now report the amount and nature of activity by students, with comparisons across two groups who made use of UI-SIV during stuvac. This represents totally voluntary use and, in the case of those who had completed the tutorial, was a return to UI-SIV. Although Figure 8.1 shows a large proportion of students accessing their learner model during the stuvac period (54 students), our analysis shows that many of these students visited their learner model briefly and left after minimal interaction. The tutorial was available for use during the stuvac period, and although a small proportion of students went to the tutorial, they did not provide answers to the tutorial questions.

We have split students into two groups, based upon whether their level of activity was above a threshold of 50 interface actions. This threshold combines interaction across several dimensions that might be orthogonal, our qualitative analysis of the log data indicated that students making more actions than the threshold appeared to engage seriously with UI-SIV. Table 8.3 shows the breakdown of these 54 students. The table shows these groups split by whether they did the tutorial as well as stuvac use or not: about a quarter of each group were in the high usage group.

We then analysed the patterns of activity across the two stuvac groups. Table 8.4 shows the type of actions performed by participants using UI-SIV during the stuvac period. All of these actions are analogous to the tutorial activities, except for “Change concepts visible”, where participants used **Term Expansion** to change the number of visible concepts; this was not part of the tutorial. We can see that in the high usage group, the first pair of columns of results in Table 8.4, more of the students used the main parts of the UI-SIV interface. Half used the search, an indication they were exploring parts of the model they wanted to know about. About half of the students used inference, an indication of using the model to dig more deeply into the mastery. Notably, a much larger proportion of participants with high usage not only examined evidence, but also explored other model views.
We now focus on students who participated in the tutorial. They are of interest because of the richer data set for them and because their return to UI-SIV suggests reflection with it was useful enough to justify making it part of their exam study. Thirty-seven participants did the tutorial (Figure 8.1).

Analysis of their activity suggested four categories based upon the way that they completed the steps of the tutorial:

- Completely correct: the participant did the task as instructed, and provided a response that was correct.

### Table 8.4: Summary of actions during stuvac

<table>
<thead>
<tr>
<th>Action</th>
<th>Number of participants above threshold using action (N=14)</th>
<th>Number of participants below threshold using action (N= 40)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Navigate concepts</td>
<td>14</td>
<td>24</td>
</tr>
<tr>
<td>Examine evidence</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>Search for concept</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>Use inference</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>Change concepts visible</td>
<td>11</td>
<td>7</td>
</tr>
<tr>
<td>Changed to view of own model</td>
<td>13</td>
<td>8</td>
</tr>
<tr>
<td>Changed to view of comparative model</td>
<td>12</td>
<td>7</td>
</tr>
<tr>
<td>Changed to view of group model</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>Selected concept</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>Deselected concept</td>
<td>7</td>
<td>5</td>
</tr>
</tbody>
</table>

### Table 8.5: Summary of task completion for the 37 students who attempted the tutorial

<table>
<thead>
<tr>
<th></th>
<th>T0</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
<th>T6</th>
<th>T7</th>
<th>T8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completed correct</td>
<td>29</td>
<td>22</td>
<td>19</td>
<td>30</td>
<td>17</td>
<td>17</td>
<td>21</td>
<td>24</td>
<td>18</td>
</tr>
<tr>
<td>Correct in context</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>7</td>
<td>6</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>Performed incorrectly</td>
<td>2</td>
<td>6</td>
<td>4</td>
<td>0</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Skipped</td>
<td>5</td>
<td>8</td>
<td>11</td>
<td>5</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>% correct (out of 37)</td>
<td>81%</td>
<td>62%</td>
<td>60%</td>
<td>86%</td>
<td>54%</td>
<td>65%</td>
<td>73%</td>
<td>73%</td>
<td>65%</td>
</tr>
<tr>
<td>% correct (out of attempted)</td>
<td>94%</td>
<td>79%</td>
<td>85%</td>
<td>100%</td>
<td>77%</td>
<td>89%</td>
<td>96%</td>
<td>96%</td>
<td>96%</td>
</tr>
</tbody>
</table>

Analysis of Quantitative Answers from Tutorial Tasks

We now focus on students who participated in the tutorial. They are of interest because of the richer data set for them and because their return to UI-SIV suggests reflection with it was useful enough to justify making it part of their exam study. Thirty-seven participants did the tutorial (Figure 8.1). Analysis of their activity suggested four categories based upon the way that they completed the steps of the tutorial:

- Complete correct: the participant did the task as instructed, and provided a response that was correct.
Correct in context: the participant did the actions correctly, but in a different context. For example, a participant might have searched for the concept cognitive walkthrough instead of GOMS in T3.

Performed incorrectly: The actions were performed incorrectly or the response was incorrect. For example, entering the score for GOMS as 0.31 when it should have been 0.62 for that student’s learner model.

Table 8.5 summarises the number of participants who completed each task and the level. We exclude T9 as it is entirely qualitative. We see the range from 54% to 81% correct completion of each task over all the participants, and a range of 77% to 100% considering only participants who left an answer for the particular task.

The last row of Table 8.5 shows an uneven number of participants skipping tasks. There were a total of 18 participants who skipped one or more of the interface and learner model navigation tasks, 8 of them working up to a certain point (less than half the tasks in all but one case) and skipping the rest of the tasks. The remaining 10 participants skipped selected questions.

**Correlating UI-SIV interaction with learning outcomes**

![Graph of Exam Mark versus Interaction Level.](image)

Figure 8.3: Graph of Exam Mark versus Interaction Level.
To gain a sense of which level of students made most use of UI-SIV, we calculated a measure of total use (the interaction level) so we could compare that with examination marks. Although the final exam is presented as an aggregated mark with low purity, we consider it reflects overall knowledge. Figure 8.1 plots interaction against exam marks for participants. The level of interaction is based on an aggregation of the number of correct tutorial task actions, the amount of exploration during the tutorial tasks, and the amount of exploration during stuvac. We only considered participants who had taken part in the tutorial tasks, as we had a better sense of their engagement with the learner model (through answering the tasks). Appendix D describes how we arrived at an interaction level for the participants.

An important part of the interpretation of our results was providing qualitative commentary on participant interaction. Appendix D contains the complete commentary for each of the participants. We provide a summary of the commentaries broken down into the shaded regions from Figure 8.3 in Table 8.6.

Based on our own observations, we can see that the more reflective participants were found in the upper left quadrant (shaded in tones of purple). We gauged their level of reflection based on what concepts they clicked on and whether they took the time to examine the evidence for them. As we move down to the yellow region on the bottom left, we noticed that most of these participants were more keen in finishing the tutorial or skipped some parts of it, possibly indicating they were less

<table>
<thead>
<tr>
<th>Region</th>
<th>Participants in region</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>P03, P12, P14, P31</td>
<td>Tutorial attempted seriously, with mostly or entirely correct responses. Actions were reflective or seemed reflective.</td>
</tr>
<tr>
<td>2</td>
<td>P01, P07, P10, P11, P20, P23, P25, P26, P32, P33, P36</td>
<td>Less reflection than the participants region 1, but most had a high rate of exploration of the interface. Mix of tutorial task results.</td>
</tr>
<tr>
<td>3</td>
<td>P08, P15, P17, P21</td>
<td>Completed tutorial, an even mix of learner model reflection and user interface exploration.</td>
</tr>
<tr>
<td>4</td>
<td>P02, P06, P09, P19, P24, P27, P29, P30, P34</td>
<td>Interactions were mostly non-reflective. Participants either only concentrated on finishing the tutorial or skipped parts of it.</td>
</tr>
<tr>
<td>5</td>
<td>P04, P05, P13, P16, P18, P22, P28, P35, P37</td>
<td>This region is for all participants who had an interaction level less than 5. This region consisted of a mix of reflective and non-reflective, but most participants had skipped the tutorial tasks.</td>
</tr>
</tbody>
</table>
interested in reflection and treated the task as something they had to do quickly before leaving the tutorial session.

It is interesting to note that there is a large, empty area to the right of region 4 and below region 2. This is the area of the graph where the students would have had enough interaction with SIV to just complete the tutorial tasks, and also achieved a high mark in the exam.

We consider regions 1 to 4: these are all the students who successfully completed at least half the tutorial, indicating they had engaged with UI-SIV and made an attempt to use it. These portions of the graph seem to show a trend where students who actively engaged in interactions with their learner models tended to have higher exam marks. We calculated the correlation, arriving at a value of 0.61. This must be interpreted cautiously as the interaction measure combines elements from different types of interactions. The correlation of the full graph is 0.27.

Table 8.7 shows the types of actions the participants could perform while doing the tutorial tasks, along with the minimum number of actions expected if working through the tutorial tasks with no exploration. The next two columns show the average number of times these actions were performed by participants, firstly an average for all participants, and secondly for participants who completed at least half the tutorial. The latter group is plotted in the graph above (Figure 8.3). This shows that the participants actually did explore their learner models as they worked through the tutorial tasks. This appears likely to indicate a combination of curiosity about the new interface and an interest in their performance as reflected by UI-SIV. It is weaker evidence of their reflective activity than their usage.
during stuvac, as many extra actions may be biased by the tutorial tasks (e.g. participants who try using a feature multiple times to reinforce what the tutorial has shown to them).

Note that all recorded values are above the bare minimum. This is true, especially in the case of navigation: when done outside of the requirements to complete the tutorial, this could be considered reflective activity.

**Analysis of Qualitative Responses from Tutorial Task**

Table 8.8 and Table 8.9 show qualitative responses to tutorial tasks T5 and T9. The response rate was 59% and 49% respectively. Reflective comments have been underlined in both tables. Additional columns show different features in the responses (further described below). Totals for each column are at the bottom of the table.

For Table 8.8, the first column marks comments related to accretion, which suggest that the participants understood the role of evidence in defining the learner model; fourteen students did this. The next two columns mention tutorial and audio-lecture evidence respectively, showing understanding of the process but also showing differences in individual student judgments of how meaningful those evidence sources are. The final column marks the 13 responses that were positive about the system’s score.

As the responses for T9 were free form comments, Table 8.9 has columns showing how we coded the responses. The coding of the columns relate to the user interface itself (columns “UI liked”, “UI found confusing”), how easy it was to learn how to use the system or follow the tutorial (“Comments on learning curve”), reflective comments or comments on how beneficial the system is to learning (“Comments on usefulness or reflection”), and anything that did not fit the previous columns (“Other orthogonal comments”).

We have underlined some of the comments in Table 8.9 that indicate the value of UI-SIV for supporting reflection.

A number of comments indicate dislike of the interface due to the overlapping words and the unconventional presentation style of the concepts: this matches our previous experience that users find the interface unconventional and, while some like this very much, some users do not (the evaluation from Chapter 5 had similar responses to the interface).
Table 8.8: Qualitative responses when participants were asked to comment on concepts that the system thought they were performing poorly in.

<table>
<thead>
<tr>
<th>Participant</th>
<th>Response to Task 5: “Select some concepts you are not doing well in based on the score. Do you think the scores here reflect your knowledge of these concepts? Why/why not?”</th>
<th>Accretion</th>
<th>Tutorial Evidence</th>
<th>Audio Evidence</th>
<th>Generally Positive about System Score</th>
<th>Hypothesis 1 Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>P01</td>
<td>Yes, don't seem to be able to get the hang of them.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P03</td>
<td>Not reflective, because there is no evidence on concept with no tutorial or audio evidence to support doing well/bad either way.</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P04</td>
<td>No, based on audio evidence, not all lectures listened to.</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P06</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>P08</td>
<td>No, this is because although I might not have accessed the webpage long enough, I still had ample practice of it in labs and lecture notes without audio on it, with regards to cognitive walkthrough</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>P09</td>
<td>No, its only because I haven’t done many of the online lectures that are red</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P10</td>
<td>Yeah, it wasn't covered much.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P11</td>
<td>Some of them... because I’m not confident with some of those concepts, but some I am very confident in...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P12</td>
<td>No, was there any evidence to be measured? Human factors - when/how/where was there info or was it assessed? [user examines concepts with little or no direct evidence – tutorial has not introduced notion of inference yet]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P19</td>
<td>Not sure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P20</td>
<td>Paper and pencil prototype are one of the first steps I carry out when designing a system, be it for the user interface, class diagram or data relationships. I ignored any of this information in the course as I felt confident in it, and as a result the system believes my performance is worse than average. I believe tutorial evidence should be ranked highly as practical demonstration of a concept far outweighs the reading or listening to instructions about it. Had I carried out tutorial work, I understand that I would have been able to demonstrate this knowledge, so I only have myself to blame for the systems poor understanding of my performance.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>P21</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>P23</td>
<td>Somewhat. The score assumes I learn everything from the site, [on viewing concepts with no direct tutorial evidence sources]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P25</td>
<td>Not all evidence has been gathered properly and fully.</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P26</td>
<td>Yes, since it refers to the time I spent on the website and the lecture notes.</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>P27</td>
<td>No because this is already apparent to me.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>P29</td>
<td>No because work done doesn’t necessarily represent my knowledge</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>P31</td>
<td>No. Because there’s no evidence to base it on. [on viewing concepts with no direct tutorial evidence sources]</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P32</td>
<td>No because some of these concepts have not been covered that I am aware. For example, I do not believe we have covered minimisation.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>P33</td>
<td>No, because the tutorials for that concept have not been marked yet. [When browsing concepts taught in the week 11 tutorial session]</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>P37</td>
<td>I think it is quite close to what I understand about the module.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
Table 8.9: Qualitative responses when participants asked for additional comments on SIV.

<table>
<thead>
<tr>
<th>Participant</th>
<th>Response to Task 9: “Thank you for doing this small tutorial. We hope you can incorporate SIV to help you study for UIDP. Please enter any additional comments below.”</th>
</tr>
</thead>
<tbody>
<tr>
<td>P01</td>
<td>Seems to be useful. Was slightly lost as to what to do at first.</td>
</tr>
<tr>
<td></td>
<td>It would be nice if the text/concepts sorted with a mouse over; right now, they are too bunched up and overlapping. (Term expansion doesn’t help much).</td>
</tr>
<tr>
<td></td>
<td>Don’t need javascript alert for ‘Loading your user model’. It actually confused me. I wasn’t sure whether to click on ‘OK’ or leave the alert there until the page refreshed.</td>
</tr>
<tr>
<td>P03</td>
<td>The links should be made closer to the visualisation. Right now, they see to be completely external – they appear not to have any affect on the visualisation because of the difference in style.</td>
</tr>
<tr>
<td></td>
<td>Otherwise, this is very helpful in understanding progress in the subject, and could assist revision. Also shows how concepts are related through the course, giving an overview as well as a detailed view (good Schniderman reference!).</td>
</tr>
<tr>
<td>P04</td>
<td>Interesting</td>
</tr>
<tr>
<td>P06</td>
<td>It is very hard to read the left panel!</td>
</tr>
<tr>
<td>P08</td>
<td>This is a really good and inventive reflective system for the course.</td>
</tr>
<tr>
<td>P10</td>
<td>Hard to read text on left, took a while to develop a strong mental model of what it was.</td>
</tr>
<tr>
<td>P12</td>
<td>It’d be nice to be able to see the topics even when they’re not related to the one selected.</td>
</tr>
<tr>
<td></td>
<td>- They seem become too small or vanish completely too often. It would also be good to have access to at least a definition of each topic.</td>
</tr>
<tr>
<td>P14</td>
<td>Good site clear and shows good relation between my understanding of the course and the rest of the year’s general understanding based on marks.</td>
</tr>
<tr>
<td>P15</td>
<td>Good work! Well done!</td>
</tr>
<tr>
<td>P17</td>
<td>The Views: doesn’t seem to work and the Java display is very hard to use. All the items are squashed together and it seems impossible to reverse it</td>
</tr>
<tr>
<td>P20</td>
<td>The graphical display is a fascinating method for presenting the information.</td>
</tr>
<tr>
<td>P22</td>
<td>A little confusing at first, but I think I’m getting used to it now.</td>
</tr>
<tr>
<td>P24</td>
<td>Nice interface mate</td>
</tr>
<tr>
<td>P25</td>
<td>Hello world</td>
</tr>
<tr>
<td>P30</td>
<td>I have done all the online lectures. However, I prefer to print them out and write my own notes on paper rather than typing them online. Does this affect my scores?</td>
</tr>
<tr>
<td>P32</td>
<td>If I want to look at all the concepts, they overlap and are too small.</td>
</tr>
<tr>
<td>P36</td>
<td>Could have clearer instructions.</td>
</tr>
<tr>
<td>P37</td>
<td>A good addition to the course to the course as a whole. Gives me an overall outlook to the course. Also allows me to know which parts that I should read up on more.</td>
</tr>
</tbody>
</table>

UI liked: 10  UI found confusing: 7 Comments on learning curve: 4  Comment on usefulness or reflection: 7  Other orthogonal comments: 6
Case study of selected participants

We now discuss in detail the interactions of selected participants with UI-SIV. We concentrate on participants who seemed to have genuine interest in their learner models, as this demonstrates cases of authentic usage and look for indications of reflective activity.

Appendix D.6 contains annotated log files for the participants examined in this case study: S01, S02, P03, P14 and P24. Two of these participants (S01 and S02) had exposure to UI-SIV only during stuvac, both had done the tutorial during this period (the tutorial pages were still available as a link off their user profile page on the course website). The remaining participants in this case study (P03, P14 and P24) are users who engaged with UI-SIV during Week 13. Both P03 and P14 worked through the tutorial completely in Week 13, and also showed signs of additional exploration of their learner model during this period. P24 had only partially completed the tutorial tasks (some questions were skipped), and did no exploration of their learner model outside of the tasks during Week 13. In this section we provide an analysis of what they did along with our interpretations of their actions during these periods.

S01

What they did: This participant had usage of UI-SIV during the stuvac period, with the second largest number of actions logged. They did not do the tutorial tasks in Week 13, and their initial activity with UI-SIV involved going through the tutorial (and did most of the actions correctly, although their responses for T4 and T8 used the audio value only in the evidence, rather than the total value). They returned to UI-SIV afterwards and proceeded to examine various concepts, a number of which were red, and related to the online learning topics of Extended Cognitive Walkthrough and Ergonomics, spending a total of 16 minutes interacting with their learner model. They examined the evidence recorded by the system for those concepts and used the inference to reason about the concepts with no evidence. After this, S01 left the SIV interface, and proceeded to examine various weekly tutorial exercises. They then went and fully listened to all the slides for the Ergonomics learning topic and printed out the annotated lecture slides for the Extended Cognitive Walkthrough learning topic.

Interpretation: This student was interesting as they only had exposure to UI-SIV during the stuvac period. No actions were logged for them interacting with the learner model in Week 13. There is indication they engaged with UI-SIV in a reflective manner, by examining the concepts that were red in their learner model, and then engaging with the corresponding learning topics (attending one, and printing out the other). Either the participant planned to revise those particular topics and out of
curiosity for the system score, decided to scrutinize those concepts in UI-SIV, or they examined red concepts in UI-SIV and then sought to revise the learning topics that taught them.

The student achieved an examination mark of 58, putting them well in the pass range for the course, an indication they had shown genuine interest to learn the material. A hand-calculation of their interaction level based on their access of the tutorial tasks in their first session, coupled with exploration in the second session gives a value of 18. Although they did the tutorial tasks under different circumstances to participants in Week 13, S01 plots directly into region 2 of the graph in Figure 8.3.

S02

**What they did:** This participant interacted with UI-SIV only during the stuvac period. They showed the third highest level of interaction during that period, and achieved the highest examination mark in the course. The participant interacted with UI-SIV in different sessions. Initially they went to UI-SIV and went through the tutorial, and like participant S01, did not leave any answers for the tutorial tasks, working through them briefly.

The participant then examined concepts relating to *Screen Design* and *User Centered Design*, and subsequently attended portions of both those lectures in separate sessions. They also went back to UI-SIV to examine concepts relating to *User Centered Design*.

**Interpretation:** This participant’s learner model was mainly red, and from analysis of their semester long access logs, had printed off the printable versions of the learning topics for design rather than attending the online versions (they had attended the online lectures for the programming parts of the course). It is interesting that they actively attended the online topics relating to the concepts they examined, considering the participant had a poor online attendance record for that part of the course (despite showing mastery of the course based on their examination performance). This shows some level of reflective activity, especially when they returned later in the day to examine their learner model. Unfortunately, as mentioned in the experimental design, the system did not live-update the learner models, so they would not have seen any change to the colours of the concepts.

This student achieved an exam mark of 70, and a hand calculation of their interaction of the tutorial task and exploration gave a value of 16; S02 plots directly into region 2 of the graph in Figure 8.3. Unlike participant S01 who did the tutorial in similar circumstances, this participant did not leave any responses and skipped some of the tutorial tasks, resulting in a lower score for their interaction.
**P03**

**What they did:** This participant did the tutorial task in Week 13 correctly, and also did additional exploration of their learner model outside of the tutorial tasks. The participant also provided feedback on the interface itself, and offered comments on the usability (Table 8.9).

They returned to their updated learner model in stuvac and scrutinised various concepts, and examined various concepts. The concepts scrutinised were mainly red and were mostly related to the *Heuristic Evaluation* topic. The participant printed out the printable version of the online lecture notes for *Heuristic Evaluation* in that session, as well as vesting the slides for the concepts that were red.

**Interpretation:** This user appeared to use UI-SIV as a reflective tool during stuvac, despite their initial critical comments of the interface in Week 13. This is strong indication they had a willingness to use the interface to identify concepts they did not know well in the course. This is supported by the fact they also printed out and viewed relevant slides relating to the concepts that were red in their learner model.

**P14**

**What they did:** This participant took part in the tutorial (performing all tasks correctly and showing a high level of interaction with UI-SIV, as described in the results in Appendix D) and subsequently went back to their learner model during stuvac. They had the highest level of interaction during this period. This participant’s learner model in UI-SIV was completely green during this period; the system believed they had mastery of the course based on their performance and attendance of online learning topics during the semester. Despite this, the participant still actively explored the concepts in the learner model and examined their corresponding evidence. They did not listen to any online learning topics during this session (they had done so during the semester already), but they did access the printable versions of all of the available learning topics.

**Interpretation:** This participant is interesting in that they had exposure to UI-SIV during the tutorial and subsequently returned to explore it again during stuvac. This user had a learner model that was 79% green in week 12 (the tutorial week). During stuvac, their learner model was 100% green. The fact they returned and scrutinised concepts in the learner model indicate a genuine interest in their performance and mastery of the concepts. In follow up correspondence, the participant stated that they understood that the scores for concepts in the learner model were based on evidence of them attending online lectures. They returned to their learner model in the stuvac period to examine whether they had missed out on revising any concepts, looking out for any concepts that appeared red.
P24

**What they did:** This participant had exposure to UI-SIV during week 13: they worked through the tutorial tasks with minimal exploration of the interface; their interactions during this time outside of the tasks seemed to be more out of initial curiosity. However, they returned to their learner model during the stuvac week and scrutinised various concepts in their learner model. They appeared to use UI-SIV to find concepts that were red, and then examine the evidence for them and visit corresponding online lecture slides that taught those concepts. The participant then proceeded to scrutinise concepts and then access online lectures for the topics *GOMS, Screen Background*, and *User Centered Design*.

Later on in a second session with the learner model, P24 examined the lecture slides that taught various concepts relating to screen design. They had visited these slides previous to this session. A third session resulted in the participant printing out the printable version of the online lectures, and whilst doing this, further examined concepts in their learner model by navigating and following the links to related concepts.

**Interpretation:** Although this participant exhibited non-reflective usage during the week 13 tutorial session, they appeared to use UI-SIV in an authentic manner during stuvac. The fact that they sought out concepts that were red and scrutinised the evidence for them, and followed it up by attending the relevant online lecture shortly after (directly from the link on the UI-SIV evidence page), is an indication that UI-SIV was used for some reflective activity.

P24 continued to explore their learner model and went to the relevant slides in the learning topics that taught those concepts for several different concepts. All accesses to the learning topics during that session were directly from UI-SIV.

As the participant had appeared to show genuine interest in their own learning (from the previous interactions of the first session), this is indication they possibly sought to revise those particular concepts, using UI-SIV to help them find those concepts.

### 8.6 Summary of Results

We now return to our three hypotheses and review the overall data presented above. For each, we have summarised what we have learnt from each of the tutorial tasks and exploratory use. In Table 8.1, we showed which tasks and exploration data were relevant to each hypothesis; we now collect the elements under those headings from the analysis above.
8.6.1 Hypothesis 1: Find concepts of importance

- **T1 – Introduction to colour, asked to find concepts known well.**

  Of those who attempted it, 79% of participants selected appropriate concepts (Table 8.5), indicating they were able to find concepts they know well by just visual exploration of the overview of their learner model.

- **T5 – Selection of concepts that the learner model believed they were not doing well in (quantitative).**

  More participants completed this task than T1; of those who attempted it, 86% selected concepts that were red on their learner model (Table 8.5). This indicates participants had increased understanding of the representation of their knowledge in the model (or chose to be more compliant, having already satisfied some initial curiosity).

- **T5 – Selection of concepts that the learner model believed they were not doing well in (qualitative).**

  The support for Hypothesis 1 column in Table 7 shows 19 out of 22 responding participants could make sense of the representation through their qualitative responses.

- **T6 – Change to comparative model of their learner model vs. the class average, and asked to find concepts scoring below average.**

  and

  **T7 – Change to model for the class average, asked to find concepts class doing well in.**

  For T6 and T7, 96% of participants attempted these tasks and were able to change to the appropriate learner model view and identify suitable concepts to answer the questions (Table 8.5). This suggests that, not only did participants show a strong grasp of the knowledge representations in the learner model, but they also could apply them when viewing models in other contexts.

- **Free exploration during tutorial.**

  The average usage shows that participants used the features beyond what was required to answer the questions (Table 8.7), in particular the number of concepts navigated, and the changing of the number of visible concepts. This gives some indication that participants could
use the interface to find concepts of interest, though as mentioned previously, these could have been in relation to completing the tutorial tasks and exploring this new interface.

- **Free exploration during stuvac.**

The relatively high use of navigation, examining evidence, and changing model views indicate that the participants were making use of the features of the system that showed them concepts of interest (Table 8.4). In particular, half the participants in the significant usage group made use of the search functionality to find concepts not immediately visible on the visualisation, suggesting they wanted to check their performance model for aspects of interest to them.

Of the participants described in the case study, we can see that S01 and S02 could find concepts of importance, and subsequently appeared to access corresponding parts of their learner model. The actions of P24 and the follow-up comments from P14 in using UI-SIV to discover concepts that were red and then revising them is a strong indication that these participants were familiar enough with UI-SIV to find concepts of importance a week after doing the tutorial tasks.

Overall, it appeared that participants could readily find concepts of importance in the interface. This reinforces our earlier evaluations (in particular Chapter 5 where users navigated a much larger structure using UI-SIV) that the underlying ontological relationships aided in structuring the data.

**8.6.2 Hypothesis 2: Examine evidence**

- **T4 – Introduction to evidence and examining concepts, asked to scrutinise GOMS.**

This task had a 77% success rate, for those who attempted it (Table 8.5), as the first task that required participants to examine evidence. Several participants performed the task incorrectly (usually by examining evidence for a different concept and answering the question with that value instead of the one for GOMS).

- **T5 – Selection of concepts that the learner model believed they were not doing well in (quantitative).**

This task had an 89% success rate, an increase over the first task, T4, which introduced and required participants to examine evidence (Table 8.5).

- **T5 – Selection of concepts that the learner model believed they were not doing well in (qualitative).**
Table 8.8 shows a majority of participants left responses commenting on the accuracy of the scores in the learner model; 15 participants had mentioned the role of evidence directly in their responses (10 participants explicitly mentioned the evidence sources). This gives an indication that learners not only were able to examine evidence using the interface, but also interpret what it meant.

In some cases showed considerable understanding of the way the learner model combined evidence, such as in responses P08 (where the participant gave indication of understanding that the amount of time spent on the audio correlated to the system interpretation of their knowledge for that concept) and P20 (the participant indicated that they had knowledge of a concept already and did not access the relevant online learning topics or tutorials during the course, and later stated because of this, the system had a poor understanding of their knowledge of that concept due to the lack of learner model evidence). In both cases, participants challenged the interpretation of the evidence and believed more weighting should have been put on tutorial evidence rather than audio evidence, an indication they could understood the LOSUM user modeling process.

From the qualitative responses for this task (Table 8.8), we can see several comments that showed reflection-on-action (underlined in responses from participants P08, P09, P14, P20, and P26). While some of these comments supported the system’s belief of learner knowledge, the ones that disagreed offered explanations relating to their actions during the semester.

- **T8 – Introduction to infer, asked to infer about a concept not doing well in.**

  In this task, 93% of participants were able to use the inference functionality to reason about a concept that was red. More participants completed this task than earlier tasks requiring participants to examine evidence, T4 and T5 (Table 8.5). This indicates that users had a strong grasp of how to examine the evidence and could use it for more complex tasks.

- **Free exploration during tutorial.**

  We can see from Table 8.7 that on average, participants examined evidence 5 to 6 times. It gives us indication that participants could examine evidence for concepts they found interesting or considered important.

  Table 8.7 also shows a reasonable level of engagement with the learner model as all of the actions performed were higher than that required to complete the tutorial. Although some of this can be attributed to trial and error with learning how to use the interface, we can see that
the high level of usage by some of the functionality such as navigating and examining evidence. This indicates participants showed some curiosity in their own learner model.

- **Free exploration during stuvac.**

Table 8.4 shows that 93% of participants with high interaction, and 35% of the other participants had examined evidence during their sessions with UI-SIV in this period.

The qualitative analysis in the case study strongly suggests that those participants could readily examine evidence. A week after using UI-SIV for the first time, P24 would examine evidence for red concepts and then directly access the online lecture slide from link in the learner model. Participants S01 and S02 could also readily examine evidence and access relevant learning topics from the main course website rather than UI-SIV.

Overall, it seemed participants could readily examine evidence. The tutorials showed that it was straightforward to learn how to use, and the case studies showed that participants would use it to examine particular concepts that they needed to revise. Participant P24 readily accessed learning material directly from the links provided in the learner model for concepts that the system thought they did not know well.

### 8.6.3 Hypothesis 3: Use Inference

- **T8 – Introduction to infer, asked to infer about a concept not doing well in.**

This task introduced participants to the notion of inference in the learner model, and builds on what they were shown in T4 and T5. Table 8.5 shows that 93% of participants could find a concept they were not doing well in and use the inference function to see an inferred value. The participants who got the question correct in context tended to make small typographical errors. For example, P27 mistyped the value after using the inference function (“47” instead of “0.47” or “47%”). Only one participant provided an incorrect value in the input box. P19 inputted the value of a red concept without inferring about it. This user did not use the inference function at all (although it appeared this participant had trouble understanding a lot of the tasks due to language barrier).

- **Free exploration during tutorial.**

We can see from Table 8.7 that the participants used inference an average of 2.41 times, with participants completing more than 50% of the tasks using it 3.0 times. It appears that participants made extra use of the inference to reinforce what they learnt in T8. As inference
was a novel feature of the interface (and it would be unlikely participants would have come across it in their normal usage of learning systems), we find these results rather promising. It indicates participants made sense of this functionality and were prepared to try it out or explore it further than was required in the tutorial task question.

- **Free exploration during stuvac.**

During stuvac, 10 out of the 54 participants used the inference function when they interacted with their learner model. However, in Table 8.4, we can see that 7 of these participants exhibited significant usage of the system. These participants used the function an average of 10 times each (though there were occurrences of repeated clicks on the button for the same concept).

The students showing the most usage had not done the tutorial in week 12, and a lot of their usage of this function may have been due to initial curiosity with the interface. Two participants who did the tutorial (and completed the inference task T8) came back during stuvac and used the inference functionality.

The case study shows that several of the participants used inference on concepts that appeared red, for example, S01 and S02 tried to infer about red concepts as there was no evidence for them in the learner model. After inference, it would still appear red. Those participants then went and attended relevant online lectures.

Overall, the inclusion of inference functionality in UI-SIV to overcome the granularity problem seemed beneficial. Although it wasn’t used as often as other functions in the system, it seemed that some participants made genuine use of it (e.g. S01 and S02). The inference functionality also showed promise as a tool for instructors to use, as demonstrated in the evaluation in the previous chapter.

**8.7 Summary**

This chapter has presented a large scale evaluation of LOSUM implemented as UI-SIV for the domain and interface described in Chapter 3. All the tools described in previous chapters were used to construct the light-weight ontology and the user models, which were subsequently used as a reflection aid by students in a live context.

From the results above we can see that the system has matched our hypotheses:

- H1 investigated whether participants could find concepts of importance using UI-SIV. Overall, participants did not have a problem with this in any of the tasks, and they could also
find concepts that were not immediately visible on the visualisation during free exploration. This is an indication that the underlying light-weight ontology structure played a role in helping users explore their learner model and find concepts that were important to the learner.

- **H2** examined whether students could easily find evidence to support why a learner model believed their knowledge is at a particular level. The quantitative results showed a majority of students examined evidence when they explored their learner models, and the qualitative responses indicated they showed understanding of the relationship between evidence sources and the values in their learner models. These results show that the concepts in the light-weight ontology matched the course content well and our evidence processing was scrutinizable.

- **H3** investigated the use of the inference function. Although it was not used as much as the other features in free exploration, this was most likely due to inference not being a natural feature of systems that the students would have been exposed to in other interfaces and learning systems. However, students were still able to use it to infer about learner model concepts with little or no evidence and show merit in the onto-increment algorithm as a simple and scrutinizable approach to overcoming the granularity problem.

Overall, this evaluation explored a practical aspect of the learner models we created by investigating whether the learner models were useful for reflection. We can see that participants were able to readily find concepts of importance, examine their evidence and use inference: these are all steps that contribute to reflective activity. In addition:

- The high interaction with UI-SIV correlated to the student’s higher examination marks (Figure 8.3) along with the qualitative responses and case study, is an indication that the ability to access the learner model was beneficial in the revision process. This is especially interesting as UI-SIV did not explicitly offer suggestions or adaptations. This seems to indicate that the stronger students actually took advantage of the open learner model as a purely reflective tool. It agrees with previous work indicating students who spend more time engaging with their learner models generally perform better (Mitrovic and Martin 2002).

- The case studies of particular users show that UI-SIV could be used for reflective activities for a range of students. S01 and S02 both appeared to use UI-SIV to examine concepts they didn’t know and subsequently sought out corresponding learning material off the website. The plots for these participants onto the graph in Figure 8.3 seem to fit the trend (even thought they did the tutorial tasks under different circumstances), placing them in region 2. Participants P03, P14 and P24 returned to UI-SIV during stuvac, a week after being exposed to it in the tutorial. Their use of it to find concepts that were red is indication they were keen
to learn concepts they did not know (P14 stated this explicitly in follow up correspondence). P03 returned to their learner model despite their reservations about the interface. P24 directly accessed the learning content from the learner model, a strong indication of reflective activity.

Task T9 asked for general comments on the system, and again several participants provided responses relating to reflection (underlined in responses P01, P03, P08, P14, P20, P30 and P37). Although the comments are brief, they give a good indication that the system promoted some degree of reflective thought among some of the participants: these also show good understanding of the interface and learner model.

The results show that our approach to creating scruitable user models using light-weight ontologies, utilising the full LOSUM toolkit, can be applied to a practical situation, in this case an environment designed to promote learner’s reflection. There is an even stronger picture when the results of this larger scale, and mainly quantitative, field trial with authentic users is combined with the series of small scale qualitative evaluations reported in previous chapters: user exploration of models for two point queries (Chapter 5), the use of SIV as a cognitive aid to instructors in the adding of metadata (Chapter 6) and also for instructors to get an overview of the performance of individual students (Chapter 7). In Chapter 10, we review possible future work that addresses the issues of participation and duration of this experiment.

The next chapter shows how parts of the LOSUM have been applied to other domains and applications, indicating flexibility beyond the UIDP learning domain we have focused on for this thesis.
Chapter 9

Applications of LOSUM

In the exploration of the way light-weight ontologies can support scrutable user modelling, we created the LOSUM toolkit. One of the indications of the power and flexibility of LOSUM and in particular, the SIV tool, is in the reuse of LOSUM in diverse applications. This chapter describes these applications and how parts of the system have been used in their tasks. We also discuss our own insights about the LOSUM tools.

9.1 SITS-VLUM – A Visualisation-based Teaching Agent for SITS

SITS, for Scrutable Intelligent Tutoring System (Kay and Holden 2002), is a teaching system designed to reuse existing learning documents. The core components of SITS are Teaching Agents, which determine which task learners should learn next.

Concepts are the pieces of knowledge that SITS models in the learner model. For example, in a C++ STL course some of the concepts include: sort, copy, vector, and list. The concepts are simply a vocabulary; SITS has no relationship structure between concepts.

SITS uses Basic Metadata in a database of references to documents. Basic Metadata is metadata that is inherently associated with the documents, for example, the organisation structure in a table of contents. A reference to a LOM format metadata source is associated with the document if it is available. These documents and Basic Metadata are independent of SITS, and are simply reused by SITS.

In addition, SITS maintains a database of Learning Metadata which relates documents in the Basic Metadata to the Concepts. SITS is novel in that there can be several sets of Learning Metadata describing the same document collection. This is because the documents can be reused in different courses, and each course may have its own interpretation of the documents, thus each have their own set of Learning Metadata.
Learning Metadata is simply a list of Concepts associated with each Document. The concepts whose understanding is required in order to understand the document are listed as \textit{prereq} concepts. The concepts that are taught by the document are listed as \textit{shows} concepts. Finally, the concepts that are referenced by the document, but whose understanding is not necessary in order to make use of the document are listed as \textit{uses} concepts.

The Learner Model in SITS uses an evidence-based approach based on um (Kay 1999) in which evidence that the user knows or does not know concepts is stored. The learner model keeps track of the learner’s knowledge of each of the concepts in the Concept vocabulary and also preference information for the learner.

\textbf{9.1.1 SITS-VLUM Interface}

The original VIUM was modified as a visualisation-based teaching agent called SITS-VLUM (Holden, Kay et al. 2002) that accessed the SITS learner model as a way for learners to easily navigate between the learning documents and the concept space.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{Figure_9_1.png}
\caption{The SITS-VLUM interface, with the visualisation on the left and a document on the right.}
\label{fig:9_1}
\end{figure}
SITS-VLUM provided considerable foundational work for the enhancements to VIUM (which eventually became SIV) described in Chapter 4. It is a demonstration of the flexibility of the VIUM tool that allows its use in a different user modelling system.

An example of the complete SITS-VIUM interface is shown in Figure 9.1, showing the visualisation on the left-hand side, and learning material, in this case a SGI reference document on C++ shown on the right-hand side.

SITS-VLUM displays the graph in two separate columns, with documents on the left hand side and concepts on the right. Selecting a document will expand the prerequisite concepts. Selecting a concept
will expand documents that teach that concept, allowing users to navigate back and forth between the document and concept space. For the document column, the colour represents whether the user is ready to read it or not - red means they have not fulfilled the prerequisite requirements, green means they are ready to read that document. Once a document has been read, its colour is changed to yellow and it is indented slightly to the right. The concept column uses green to mean that the user is deemed to know this concept, and red to indicate that they have not and need to read the relevant documents.

As an example, suppose a student Jane wants to read a particular document *SGI: priority queue*, but the title appears red meaning she has not fulfilled all the prerequisites. She selects it (Figure 9.2), expanding the prerequisite concepts on the right hand side. The concepts *Assignable*,

---

Figure 9.3: SITS-VIUM with concept *DefaultConstructable* selected. Documents relating to this concept are shown on the left.
LessThanComparable and Container are green, indicating Jane has learnt these concepts. The remaining concepts DefaultConstructable, RandomAccessContainer and Sequence are shown in red, indicating Jane has not yet learnt them.

She then clicks on the unknown concept DefaultConstructable to see what documents show this concept, causing the SGI: DefaultConstructable document to be displayed (Figure 9.3). Jane decides she will read the SGI: DefaultConstructable document as it is green. This example shows how users can navigate through their user model and explore the relationship between the documents and the concepts they teach.

### 9.2 SITS-VLUM Implementation

SITS-VLUM accepts a graph where each node represents either a document or a concept. Each document connects only to concept nodes representing the concepts that the user must know, i.e. prerequisites, in order to be able to understand that particular document. Conversely, concepts have connections to documents that teach the concept. We use this to enable the user to see the

```xml
<rdf:Description about="show_content.cgi?ID=663;courseID=8">
  <dc:Title>SGI: random_shuffle</dc:Title>
  <gmp:componentType>url</gmp:componentType>
  <gmp:results rdf:parseType="Resource" gmp:dataset="gmp:average" gmp:mark="0.33"
  gmp:reliability="1.00"/>
  <gmp:peer gmp:peerType="shows" rdf:resource="FunctionObject"/>
  <gmp:peer gmp:peerType="shows" rdf:resource="RandomAccessIterator"/>
  <gmp:peer gmp:peerType="shows" rdf:resource="RandomNumberGenerator"/>
</rdf:Description>
```

Figure 9.4: Excerpt of a document type from the SITS-VLUM RDF serialisation. We can see on line 3 the designation of the component as a *url.*

```xml
<rdf:Description about="random_shuffle">
  <dc:Title>random_shuffle</dc:Title>
  <gmp:componentType>concept</gmp:componentType>
  <gmp:results rdf:parseType="Resource" gmp:dataset="gmp:average" gmp:mark="0"
  gmp:reliability="0.57"/>
  <gmp:peer gmp:peerType="shows" rdf:resource="show_content.cgi?ID=663;courseID=8"/>
</rdf:Description>
```

Figure 9.5: Excerpt of a concept type from the SITS-VLUM RDF serialisation. We can see on line 3 the designation of the component as a *concept.*
relationships between the documents and concepts associated with them. Additional elements were added to the RDF for typing the components and the relationships. Each component is designated as either a document or a concept by its own `componentType` element. Figure 9.4 and Figure 9.5 show excerpts for a document and a concept from the enhanced SITS-VLUM RDF format. We can see the `componentType` on line 3 of each figure.

### 9.2.1 Insights

In the process of designing, implementing and evaluating SITS-VLUM, we gained valuable insight into the ways visualisations could be supported with light-weight ontologies. Primarily, it was an opportunity to explore one approach for using light-weight ontologies to structure a document domain and display it as a user model.

The combination of the concept and documents into one visualisation was an interesting experiment in combining one form of light-weight ontology and real documents together. The ontology is constructed in terms of the relationships `prereq` and `teaches`. The set of `prereq` relationships form a more conventional concept ontology where the relationships are defined by the documents. Note that a document may have prerequisites related to the particular way the author chose to teach. They may choose to teach a particular concept, and make use of another related concept even though it was not necessary.

Although we chose not to utilise the dual column approach of documents and concepts in the eventual design of SIV, it did prove to us that the horizontal spacing could be used for a number of different purposes, and lead to the use of a horizontal overlay input file in addition to just the colour overlay file.

The relationships that link the concepts, `prereq`, `shows` and `uses` form a very basic ontology over what would otherwise be unstructured. The use of the ontology helps facilitate exploration of the learning domain and encourage reflection-in-action (Schon 1983).

Overall, SITS-VLUM provided considerable initial direction for SIV. Although it only went to a prototype stage, there is promise in being able to view documents and concepts in the visualisation with the advent of standards such as RSS (Libby 1999) and SKOS (Miles and Brickley 2005) that could benefit from being viewable and related on a single interface.

### 9.3 Metasaur

Metasaur (Chapter 6) is a demonstration of the use of SIV in a metadata authoring environment. The use of SIV for this purpose was not an initial goal of the thesis. However, our experiences found that
the task of metadata annotation was error-prone and time-consuming. Our original approach involved hand-annotating the content, manually typing in the term. This was made more difficult by the fact that we would have to constantly pause and re-listen to the associated audio content. There was a need to be able to easily explore the available term vocabulary in a way that was efficient and facilitated the easy discovery of terms. We realised that SIV could fulfill this role, and it was integrated into an application that became Metasaur.

This provided us to further explore SIV as an ontology visualisation tool and fine-tune the way it displayed concepts and relationships:

- The ontology displayed in SIV for Metasaur consisted of over 1000 concepts, considerably larger than the user models displayed in previous applications. This required tuning, particularly for concepts with large fan-out.

- Metasaur provided an opportunity to implement a JavaScript API for the application to control the visualisation, which would eventually help make the integration of SIV into other systems (such as Assess) a simple process.

- It allowed us to use SIV as an interactive way validate that the ontology was suitable for our user modelling goals. The way SIV visualised the related concepts, and the features to search the ontology made us aware of missing terms: the restricted ontology problem.

Metasaur also provided additional opportunities to evaluate SIV. The evaluation in Chapter 6 gave us the opportunity to evaluate an interface that incorporated SIV in a role other than user modelling. It showed that the participants were quite comfortable with the searching facilities provided by SIV, and was a good indication that people would be able to navigate and scrutinise an ontology-structured user model.

9.4 Assess

Assess (Li and Kay 2005) is a online tool that aims to promote learner reflection through self-evaluation. There are two core reflective components (Schon 1983) in Assess:

- **Reflection-in-action**: self-assessment tasks

- **Reflection-on-action**: the user profile (featuring SIV)

The first version of Assess was built in 2002, stemming from a prototype developed in 1998, but has evolved considerably since then. Further details of the Assess system including an evaluation can be found in (Li and Kay 2005).
9.4.1 Self-assessment Tasks

Learners work through a number of small programming tasks, providing their own solutions to problems presented, and then self-assess them with criteria provided by the instructor. A screenshot of the interface for inputting an answer for a problem is shown in Figure 9.6. There are a number of steps for learners to go through in the process of answering the problem. They are first provided with a problem (left panel) and set of learning objectives (top middle panel). Different steps are shown under the learning objectives panel.

The first step is the learner submitting their own solution. The learners asked to upload this in a file. Previous versions of Assess provided learners with a text input to code in their solution directly. The current version requires learners to use their own editor, thereby encouraging them to test their code before they submit. Once they have a solution, learners then click the “Save and assess” button, and are presented with the assessment interface, where they are then asked to rate their own solution based on criteria determined by the instructor. The self-assessment interface for the problem in Figure 9.6 is

![Figure 9.6: A problem in Assess showing the steps required for self-evaluation.](image)
The learner required to rate their solution according to each of the criteria with the options true, false and no opinion (the default is no opinion). The current version of Assess can also perform automated-testing of the learner’s solution. The solution, self-assessed ratings and the automated-testing results are all saved as evidence in a PersonisLite learner model.

Learners can also read and assess solutions provided by the instructor. This is accessed from the links provided in the panel Step 3: Examples from Figure 9.6 (bottom-middle panel). A sample solution is shown in. These solutions are not necessarily perfect solutions, but often demonstrate common misconceptions to encourage learners to think and evaluate their own understanding of the problems. These are assessed with the same criteria used to assess their own solution (in the case of Figure 9.8, this would be criteria from Figure 9.7). The student rating of these example solutions are also used as evidence of knowledge, feeding into their learner model.

9.4.2 User Profile Page

The learner model storing evidence as the learners work through the problems can be accessed through a User Profile page in Assess. A standard version of SIV has been incorporated into this page; a screenshot can be seen in Figure 9.9. SIV can be seen on the left-hand side. A list of the concepts taught in the self-assessment tasks is listed on the right-hand side. The horizontal positioning in SIV has been used to display the teaching/learning goals. The colour represents learner knowledge over several uses of SIV. Learners can use the open learner model as a tool to aid in their planning of topics to learn; it is intended that they practice the questions that relate to concepts that would appear red on their learner model to increase their mastery of those concepts.
Insights

Assess is another illustration of the use of SIV for learning. The evaluations of Assess showed that students could effectively use SIV: over 87% could navigate and examine evidence about a concept in the learner model (Li and Kay 2005). It supports our own findings from Chapter 8 that students could...
use the visualisation tool for reflection and appreciated the fact it could be used to show learning goals.

The development of Assess also provided feedback about the flexibility of SIV as a tool that could be incorporated into other applications with a minimum of effort. A JavaScript-based API was formalized to allow other applications to control and interact with SIV.

The applet and associated JavaScript was easily integrated into the Assess system with minimal effort, and the underlying ontology for the user model was created with a hand-crafted dictionary and
Mecureo. The dictionary consisted of both specialised concepts defined in the context of the learning domain, and also generic concepts with definitions lifted from previous course notes.

### 9.5 Advanced Technologies for Learning Course

The Advanced Technologies for Learning course at the University of Sydney is designed for students in their final year of a four year IT honours degree. One of the assessment criteria for this course is a project in a learning domain that explores technologies that can aid teaching. In this course, students were allowed to make use of components from the LOSUM toolkits in their projects.

Several students made use of SIV and Mecureo, The results they achieved indicated the ease with which LOSUM can be used – they learnt how to use the tools (as black box applications) very quickly and delivered systems that were easily extensible. We present below a online version of the Mecureo interface that was made available to students to allow them to generate light-weight ontologies to use in their projects, and then describe how this is used in one such project, MyMuseum.

### 9.5.1 Mecureo online interface

A simplified version of Mecureo was made available to students. It allowed students to upload a text file in the format Mecureo required and would produce ontologies serialized to a number of different

![Mecureo online interface screenshot](image)

Figure 9.10: Screenshot of Mecureo online interface where students can upload their own dictionaries.
formats. By using Mecureo to produce ontologies, students were encouraged to have scrutable applications as their domain model always linked back to the dictionary definition. Due to the small scale nature of the projects, the dictionary files were hand-crafted by the students. Changes to the ontologies were made by changing the definitions, as the students did not have access to the Mecureo configuration files to modify keywords or categories.

The screenshot in Figure 9.10 shows the Mecureo online interface for students to upload their own dictionaries. In this example, the text file “c:\atl_projects.txt” has been chosen. They can choose the matching level as well, ranging from no matching to full substring matching. This would affect the number of relationships generated in the ontology (in the example, 5: Substring Matching has been selected). Once the form is submitted, the webserver instantiates and runs Mecureo using the form inputs as parameters. The result is then cached as a direct graph file that the students can choose to output in a range of formats: dot, VIUM RDF (described in Chapter 4), SIV OWL (described in Chapter 5), and also directly to SIV (Figure 9.11).

Students could also use the source code for embedding the SIV applet in a HTML page on their projects.

9.5.2 MyMuseum

MyMuseum is a scrutable, personalised museum guide that offers adapted information about museum exhibits for visitors (Bright, Kay et al. 2005). The system is built on top of Cellerator, a framework
for developing websites\(^\text{22}\). MyMuseum features an adaptive hypertext system that allows users to scrutinise adaptations. In MyMuseum, the information about exhibits in a museum is adapted based on the information about a user, stored in a PersonisLite-based user model. Users have the option to scrutinize the adapted information and see what information has been tailored for them.

MyMuseum was developed in the context of the Nicholson Museum, a museum at the University of Sydney\(^\text{23}\) that specialises in classical antiquities. Users are asked to answer a question on their purpose for visiting the museum. The system will generate the rest of their user model based on pre-set stereotypes. Users can change the options in their user model at any stage. Once in the system, they can browse information about a particular context, with relevant references to the exhibits in the museum that demonstrate the subject matter. This information is designed to complement the information provided on the physical exhibit labels.

A screenshot of one such information page is shown in the upper screenshot of Figure 9.12 for the subject *Hector*, a hero from Troy during the Trojan Wars (left panel). On the top right, we can see the settings in the user model. Of note are the facts that the purpose of the user’s visit is to learn about Troy (*learn_troy*), and their level of expertise is basic. The system has only included introductory level information about *Hector* in the left panel. The user can scrutinise the adaptations by simply clicking the link “How was this page adapted to you?” at the bottom of the left panel, and see the information that was excluded. A screenshot of the same page with the additional adaptation information can be seen in the lower screenshot of Figure 9.12. The content has been colour-coded, with the information included on their adapted page highlighted in yellow, and content that was excluded highlighted in green.

The MyMuseum project makes use of ontologies generated by the online version of Mecureo. The relationships between the concepts are displayed on the pages as the *Related Topic* links to other content in the website. One important design feature for the use of automatically-generated ontologies is the fact that different visitors have different levels of understanding about the topics. A single static ontology would not cater for the tasks and needs of the different types of museum visitors. Therefore, by being able to generate ontologies from relevant sources for the visitors, the adaptation can be improved. For example, an ontology can be generated from academic sources to cater to university students studying history. Meanwhile, a totally different ontology can be generated from the content in school textbooks for school students.


Figure 9.12: MyMuseum interface for a page on Hector (upper) and showing the adaptations (lower).
Although the current version of the interface does not include SIV, the students did explore the ontology through the visualisation and a possible future system has been considered (Kay, Lum et al. 2005).

9.5.3 Insights

The implementation of a cut-down online version of Mecureo and a readily usable version of SIV allows LOSUM tools to be easily used in a teaching environment where students can learn to create their own user models based on our approaches. In the MyMuseum project, the students made use of Mecureo as a way to structure their content and integrate this into a system that had PersonisLite-based user models and adaptable content.

In the future, these tools can be made available for student projects to aid in the teaching of user modelling, personalisation and adaptive e-learning systems. This demonstrates the flexibility and adaptability of the tools and the ease at which they can be used as single components or as a whole toolkit.

9.6 TARO

TARO, for Teaching from an Automatically Retrieved Ontology, is the first application utilizing a next generation version of SIV and Mecureo. TARO demonstrates how ontologies can be used for independent learning, allowing learners to gain a deeper understanding of the domain through interaction and exploration of the concepts and ontological relationships. This is similar to TM4L (Dicheva and Dichev 2006); however, that system aims to make use of Topic Maps as a construct for teaching about a domain, whereas TARO aims to exploit SKOS-based light-weight ontologies.

Figure 9.13 shows the architecture of TARO. The light-weight ontology generation system, Scrutable Automatically Generated Ontologies (SAGO), is an evolution of Mecureo that can parse glossaries in the Simple Knowledge Organization System (SKOS) format that include an embedded definition, and output a corresponding SKOS file with Mecureo relationships added in as extra information (the relationships in Mecureo are discussed in Chapter 5).

In Figure 9.14 and Figure 9.15, we can see corresponding input and output for a fragment of the SKOS file representing the concept Write permission. Extra information about the relationships beyond the basic SKOS defined relationships can be seen in Figure 9.15 with the tags in the sago namespace. This extra information is generated during the ontology creation process by SAGO, with the goal of supporting user scrutiny of the ontology.
Of note is that the original embedded descriptions from the input SKOS file can have existing relationships defined already (using the standard SKOS relationships), and SAGO will add in additional relationships it has found. This means SAGO can be used to discover new or unspecified relationships in existing documents.

The reason that extra relationships are added is to achieve a finer granularity of relationship types to aid in scrutability of the light-weight ontology. We can see from Figure 9.15 that the relationships (for example, lines 9 to 16) contain additional metadata: the keywords or phrase from the definition that Mecurio found that was the basis for the relationship, and the Mecurio properties for it (strength, type and link weight). These are embedded inside a standard SKOS relationship, in the case of the example on line 9 to 16 this is a `skos:narrower` relationship.

An evaluation assessed if TARO provided learners with an environment where they could build upon their existing domain knowledge and reasoning about relationships between concepts by exploring the ontology. Participants used the interface shown in Figure 9.16 to carry out a task involving the construction of a concept map of the domain as they understand it. This was compared to a concept map they had created before exploring the ontology with TARO. Results showed an average improvement of over 70% in the number of ‘good’ propositions about the domain after using TARO. Further use of TARO is planned for exploring the effects of open domain and user models for supporting independent learning.
Figure 9.14: Fragment of SAGO input serialized in SKOS for the concept Write permission.

```xml
<skos:Concept rdf:about="write%20permission">
  <skos:prefLabel>Write permission</skos:prefLabel>
  <skos:definition>
    An access mode that allows a person to...
  </skos:definition>
</skos:Concept>
```

Figure 9.15: Fragment of SAGO output for the input shown in Figure 9.14. Note the added relationships and the information about them.

```xml
<skos:Concept rdf:about="Write%20permission">
  <skos:narrower rdf:resource="#File">
    <sago:keyword sago:source="Write%20permission">for example</sago:keyword>
    <sago:strength>strong</sago:strength>
    <sago:type>child</sago:type>
    <sago:linkWeight>0.435500</sago:linkWeight>
  </skos:narrower>
  <skos:related rdf:resource="#UNIX%20file%20permission">
    <sago:keyword sago:source="UNIX%20file%20permission">Associated</sago:keyword>
    <sago:strength>normal</sago:strength>
    <sago:type>sibling</sago:type>
    <sago:linkWeight>0.671950</sago:linkWeight>
  </skos:related>
  ...
</skos:Concept>
```
9.6.1 Insights

TARO effectively forms the next generation of two core tools in LOSUM: SIV and Mecureo. That the ontologies are automatically generated means TARO has all the benefits mentioned in Chapter 5 with Mecureo. The switch to using SKOS as the serialisation format for the ontologies is more appropriate to the type of data and the level of complexity of the light-weight ontologies used in LOSUM. The original format, OWL, provided many extra semantic constructs that were not utilised in our light-weight ontologies (for example, specifying disjoint relationships). In terms of the overall user perspective of the system, the change in serialisation has no impact on the interface.

Importantly, TARO also explores the way that SIV and the light-weight ontology can be used as a direct teaching tool. Coupled with concept mapping, it challenges users to think about the relationships between concepts and their role in the domain model.

9.7 Summary

This chapter has described several systems where the LOSUM toolkit has been integrated into applications at various stages in its evolution. These applications illustrate the use of LOSUM for
creating and using light-weight ontologies in user modelling. It indicates both the flexibility and power of the toolkit and also its adaptability to different application environments.

This thesis has been restricted to the exploration of ontologies for supporting core elements of scrutable user modelling. A core element of LOSUM is the visualisation tool, SIV. The applications and adaptations described in this chapter indicate some of the roles for a tool like SIV:

- In SITS-VLUM as a guide to finding documents that teach particular concepts, as well as displaying concepts associated with documents and finally, defining a novel and alternative way to build ontologies from suitably tagged documents. In this case, SITS provided the ontology, metadata, content and user models. SITS-VLUM provided a visualisation that gave access to these different components.

- In Assess, helping learners reflect, plan and monitor their learning. In Assess, SIV provided a user model view to Personis-based user models, with the Assess website providing the content and metadata.

- In TARO, using the ontology as a direct teaching tool. The application enhanced the generated ontologies by extending Mecureo to produce SKOS output. The relationships used by Mecureo were mapped to SKOS relationships and enhanced accordingly.

- Smaller versions of the tools as aids in the teaching of user modelling, personalisation and e-learning courses. These were all designed to allow users with little background in the different tools that make up LOSUM to learn how to use them and integrate them into their own projects.

These applications also presented opportunities for additional evaluations of SIV in scrutable systems beyond the domain in this thesis. These evaluations included SITS-VLUM, for the learning of C++ programming (Holden, Kay et al. 2002); Assess, for learner reflection of C programming (Li and Kay 2005); MyMuseum, in the context of the Trojan War (Bright, Kay et al. 2005).
Chapter 10

Conclusions

In this thesis we have explored the ways that light-weight ontologies can support scrutable user modelling. We have covered all of the core components from Figure 10.1: the ontology, metadata annotation of domain content, processing evidence into user models, and providing an interface to the user model. The system components we have utilised to make up LOSUM are shown in dark blue around the core data components.

The light-weight ontology has played a role in supporting or contributing to all aspects of the LOSUM toolkit as a demonstration of our approach. This chapter summarises our contributions with links back to previous chapters, then follows with a description of the limitations in our approach and possible future work.

Figure 10.1: Overview of the major system components and interactions between them.
10.1 Contributions

The contributions this thesis provides to the field of user modelling are:

- LOSUM, an implementation of a toolkit that utilises light-weight ontologies to support the user modelling process and in particular, large user models that comprise of hundreds of components. Chapters 3 to 7 described the elements of LOSUM and the quantitative evaluation.

- SIV, a visualisation tool that shows how light-weight ontologies can be used to structure and visualise a large user model, addressing the scrutability user model interface problem and also the ontology interface problem. The interface incorporates the visualisation and provides scrutability by linking the user model structure back to the source ontology, and the system beliefs back to the original evidence sources (Chapter 3). Building upon a solid base, VIUM, we introduced an ontology as a systematic way to structure the user model in the visualisation. At the same time, we made several technical enhancements that separated the elements of the user model and its various interpretations. This resulted in a much more flexible and adaptable tool for visualizing ontology-based user models (Chapter 4).

- Metasaur, an interface for the annotation of metadata, illustrating our approach to the metadata annotation problem through the inclusion of SIV as an ontology visualisation tool to aid users in discovering metadata terms, and also as an interface to address the restricted ontology problem in a practical way (Kay and Lum 2004). Evaluations of Metasaur showed that it could be readily used to perform metadata annotation on learning objects (6.5).

- The use of light-weight ontologies to address a number of problems, encountered in the user modelling process and the scrutability of large user models, and demonstrated in the LOSUM toolkit mentioned above. These are:
  
  - the issue of common vocabularies for metadata terms and user model components in the user model definition problem;
  
  - the granularity problem for which we utilise the ontology to support reasoning across granularities in the user model; we wanted to infer the value of components in the user model across different granularities of concepts and evidence sources. At the same time, we wanted the reasoning process to be simple enough to explain to users, and also easily customised to match their preferences or learning styles, and thus introduced the onto-increment approach (7.3.3). A formative evaluation with
instructors (Kay and Lum 2005) and the main evaluation with students showed they could use this feature in the process of scrutinizing user models.

- the interface issues associated with the *metadata annotation problem* and the *scrutable user model interface problem*. We use SIV as a tool to visualise the ontology and conducted formative evaluations (Apted, Kay et al. 2003; Apted, Lum et al. 2004) showing that users could understand how to use the interface and the ontologies were satisfactory in terms of quality (5.4 and 6.5).

- Scorable approaches to processing evidence based on comparison to a relative standard as a way to address the *evidence normalisation problem*, and a scorable method to combine evidence across different evidence sources to address the *evidence combination problem* (Chapter 7). The formula presented was simple yet flexible enough for both instructors and students to adapt to match their own preferences or goals (7.2.2). The main evaluation (Chapter 8) showed students could understand the user model creation and maintenance techniques presented in this chapter through their actions and qualitative responses.

- An evaluation of actual use and usability in the form of a field trial and case study (Chapter 8). Seventy-seven students had exposure to their learner models through a configuration of LOSUM called UI-SIV (Chapter 8). The results showed that students could use the interface (consistent with the results from the aforementioned formative evaluations) and understand the process of user model construction (8.6). Case studies of selected participants showed strong indications of reflective activities while using UI-SIV during the period of stuvac, just before the final exam.

- A demonstration of the flexibility and power of LOSUM as a toolkit for constructing user models from light-weight ontologies by its deployment in several applications (Chapter 9). We described three applications that integrated part or all of the tools in LOSUM: SV (9.1), Assess (9.2) and TARO (9.6). In addition, the use of LOSUM as an aid to teaching showed the potential for students to readily use and apply the tools to their own projects (9.5) such as MyMuseum (9.5.2).

Together these contributions constitute significant progress in the exploration of the roles light-weight ontologies for user modelling: in building user models; reasoning about them; defining metadata for learning objects; and finally supporting the essence of scrutability, structuring large user models.
10.2 Future Directions and Limitations

We highlight some immediate and future directions of the LOSUM system as well as limitations and possible solutions. We divide this section based on the core components from Figure 10.1 and also the evaluation.

10.2.1 Domain

Other Domain Ontologies

Another direction is the application of the tools to other domains such as for visualizing semantically related content. An important motivation for moving to Semantic Web standards is the ability to use any arbitrary OWL or SKOS format ontology (which was discussed in section 5.4). Ideally, LOSUM should be able to incorporate and use any ontology specified in OWL or SKOS.

At the moment, LOSUM requires well formed (or in the case of OWL, the use of the Mecureo namespace) serialisations. It was infeasible to incorporate more generic parsing of the files during the development of LOSUM due to scalability and time constraints, as well as the revision of the standards due to them being in development themselves at the time. Now that the standards are more mature and also more ontologies being made available, enhancement of the system to parse arbitrary semantic web ontologies would be an important addition to increase LOSUM’s interoperability with other systems.

10.2.2 Ontology

Semantic Web Standards

The TARO system (9.4) is the start of the next generation of some of the tools in LOSUM. One goal of TARO is to align the tools more closely with current developments and advancements in Semantic Web technologies. A start to this has been the serialisation of the light-weight ontologies in SKOS format. This will hopefully allow future development to also leverage tools and systems associated with SKOS, as the format is better suited to the Mecureo-generated, light-weight ontology structure than OWL.

Multiple Ontologies

With a more flexible parser that allows us to read in any arbitrary SKOS or OWL file, and other standards even such as RSS, we open the door to many semantic applications. Semantically linking content is already done in applications such as the Semantic Blog (Cayzer 2004). There is also possibility to do this on a wider scale incorporating multiple content sources.
An example of this would be to modify SIV to display RSS headlines from multiple feeds, as well as concepts of a domain common to all the feeds. Users would then be able to select a concept, and SIV would focus on relevant article headlines that related to the concept. This would be similar to the work done in SITS-VLUM (9.1).

**10.2.3 User Modelling Server**

*Resolver and Inference Algorithm Library*

Currently, all the Personis resolvers used in LOSUM are hand-crafted. There is currently very little code reuse between the resolver algorithms. For example, the resolvers to determine the audio and tutorial scores for a concept both use summation and averaging over a list of evidence in their separate algorithms. It would be useful to abstract these basic functions into a library of “resolver building blocks”.

A practical next step would be to create a library of typical resolvers that applications might use. For example, the algorithms presented by Hightower and Borriello (2004) in the research area of location awareness and ubiquitous computing can be adapted and abstracted to different Personis resolvers as follows:

- **Point**: Take the most recent piece of evidence for this concept (this is currently the default resolver in Personis).

- **Centroid**: Resolve the most recent piece of evidence for this concept and the last piece of evidence for immediately related concepts based on the domain model.

- **Smooth Centroid**: Same as Centroid, but weight each piece of evidence by its age.

We can see from the example above that Smooth Centroid generalises Centroid which in turn generalises the Point algorithm. It would make sense to be able to be readily able to pick and choose appropriate resolvers from the library to build more complex ones.

**Other User Modelling Servers**

A possible future direction is the integration of the SIV interface with other user or student modelling servers. A good example of this is the e-learning architecture KnowledgeTree (Brusilovsky 2004). It should be straightforward for SIV to be able to access user model data from the CUMULATE user modelling server (Brusilovsky, Sosnovsky et al. 2005) in the KnowledgeTree architecture. Table 10.1 shows a comparison of both Personis and CUMULATE, as well as a comparison of both servers applied to an e-learning domain. In the case of Personis, we describe it with respect to the UI-SIV...
system used for the evaluation in Chapter 8. For CUMULATE, we describe it with respect to QuizGuide (Brusilovsky, Sosnovsky et al. 2004), a question based online assessment system of learner knowledge. We can see that both systems are very similar and it would be interesting to bridge the two systems.

Table 10.1: Comparison of the user modelling servers Personis and CUMULATE.

<table>
<thead>
<tr>
<th>Personis</th>
<th>CUMULATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evidence storage</td>
<td>Event storage</td>
</tr>
<tr>
<td>Resolvers on evidence</td>
<td>Inference Agents on events</td>
</tr>
<tr>
<td>Python/HTTP API:</td>
<td></td>
</tr>
<tr>
<td>- update UM via HTTP or Python API</td>
<td>- update UM via HTTP</td>
</tr>
<tr>
<td>- results serialisable to HTML</td>
<td>- results serialized to HTML</td>
</tr>
<tr>
<td>Personis for LOSUM (UI-SIV)</td>
<td>CUMULATE for QuizGuide</td>
</tr>
<tr>
<td>Concept Based (up to ~700) on domain level</td>
<td>Topic Based (Order of 40-60) based on course level</td>
</tr>
<tr>
<td>Single evidence contributes to multiple concepts</td>
<td>Single event contributes to single topic</td>
</tr>
<tr>
<td>Resolvers: Best Heard, Tute Mark normalized [0..1]</td>
<td>Inference: Events normalized [0..1]</td>
</tr>
<tr>
<td>Inference algorithm: conservator, averager</td>
<td>Inference algorithm: averager</td>
</tr>
<tr>
<td>Scope: Infer about concept from other concepts</td>
<td>Scope: Infer about topics from events</td>
</tr>
</tbody>
</table>

10.2.4 User Model Visualisation

Multiple Views

The LOSUM system has been designed so that we can substitute different visualisations in its place, or provide multiple view visualisations (Baldonado, Woodruff et al. 2000). This would be most appropriate as SIV can only visualise 2 dimensions of the user model: on the colour and on the horizontal overlays. In the learner models in our evaluation, we use the system score for the colour and the importance of the concepts in the course for the horizontal. Potentially, other horizontal overlay files include: the amount of evidence for each concept, the certainty of the system (as in VIUM), the time of semester students were meant to first learn that concept, suggestions for the next concepts to learn, and so on.

These extra overlays could be incorporated onto a secondary visualisation to enhance usability. However, care would have to be taken to ensure that users are not burdened with the additional dimensions.
10.2.5 Evaluation

Participant Population

The participants for the study consisted entirely of students studying computing. They would have likely had a strong familiarity with e-learning systems and the use of computers in their education. It would be interesting to evaluate the system in a different domain where the users would have had less experience with e-learning systems, as the scrutability of the user model creation process might require more verbose or abstracted explanations.

Ontology Considerations

Although we had evaluated the Mecureo tool and manually enhanced the ontology, we had no rigorous evaluation of the domain ontology used in the evaluation. Another ontology might have provided different insights or reactions from the students.

Access to Historical User Models

The timing of the evaluation resulted in a limited time span for the participants to use SIV and learn about their learner models. Although models were produced out of the evidence at different stages of the semester, these were not viewable by participants during the user study. Students were only able to view their model at Week 13 and at Stuvac, a span of one week. An evaluation where students would have been able to access the view of the model as it was at previous weeks might have provided more opportunities for reflection.

Longer Term User Study

It would be interesting to see how the open learner model could further promote reflection and learning by having an “empty” learner model available to the students at the start of semester, and have it grow as the students progress through the material, and measure the level of usage of the SIV with their final performance.

Adaptive Functionality

In the evaluation, many participants commented that the open learner model was helpful for revision and reflection. However, UI-SIV did not provide any adaptation to provide a personalised learning experience. Evaluations of LOSUM in an environment that also provided personalised content based might provide additional reasons for participants to access their user model and result in different reactions and usage patterns.
10.3 Summary

This thesis has explored the ways light-weight ontologies can be used to support scrutable user modelling. We have shown that a number of important problems can be overcome with the aid of light-weight ontologies (10.1).

We also demonstrated our approach through the implementation of the LOSUM toolkit. A large scale evaluation was conducted to evaluate UI-SIV: a system built using the components in LOSUM for the UIDP e-learning domain. The main outcomes of evaluation were the provision of a tool that learners used to assist in revision, and showed that the learner models constructed with our approach could be used in an authentic environment.

We conclude that light-weight ontologies have an important role for scrutable user modelling. The simple, but effective, techniques have allowed the tools we have used and developed be used in other domains and applications already, and can be applied to many more to ensure scrutable user modelling.
Appendix A

Exploration of Local Definition Linkage in Mecureo

This appendix contains the definitions used for concepts added to the source glossary to explore the linkage created by Mecureo. This is discussed in Section 6.5.2.

In summary:

The definitions are formatted so they can be parsed by Mecureo.

There are three sets of definitions: Concept Name Only, Concept and Definition, and Concept, Definition and Related.

Each definition set features the same six concepts: novice users, usability technique, exploration, discretionary users, casual users and testing process.

The definitions themselves have been taken as the contents of the slides that teach the respective concept.
A.1 Concept Name Only

novice users

[$novice users]
related:
categories: <User Profiling>

usability technique

[$usability technique]
related:
categories: <Usability Evaluation>

exploration

[$exploration]
related:
categories: <Usability Methods>

discretionary users

[$discretionary users]
related:
categories: <User Profiling>

casual users

[$casual users]
related:
categories: <User Profiling>

testing process

[$testing process]
related:
categories: <Usability Evaluation>
A.2 Concept and Definition

novice users

Cognitive Walkthrough
good for novices, discretionary, casual users. assess user success in task completion. ability to recover from errors. typically not speed of performance. do not need to know sequence of operations.

related:
categories: <User Profiling>

usability technique

Cognitive walkthrough
Simulate the way users explore and learn about an interactive system.

related:
categories: <Usability Evaluation>

exploration

Cognitive walkthrough
Simulate the way users explore and learn about an interactive system.

related:
categories: <Usability Methods>

discretionary users

Cognitive Walkthrough
good for novices, discretionary, casual users. assess user success in task completion. ability to recover from errors. typically not speed of performance. do not need to know sequence of operations.

related:
categories: <User Profiling>

casual users
[#casual users]  
Cognitive Walkthrough  
good for novices, discretionary, casual users.  
assess user success in task completion.  
ability to recover from errors.  
typically not speed of performance.  
do not need to know sequence of operations.

related:  
categories: <User Profiling>

testing process

[#testing process]  
Summary of flaws so far.  
Potential confusion about need for steps 1 and 2 as part of the fare determination phase.  
No feedback on amount deposited.  
No means to get money back.  
No assessment of fare typing option.

related:  
categories: <Usability Evaluation>
A.3 Concept, Definition and Related

novice users

Cognitive Walkthrough
good for novices, discretionary, casual users.
assess user success in task completion.
ability to recover from errors.
typically not speed of performance.
do not need to know sequence of operations.

related: {user studies} {error}
categories: <User Profiling>

usability technique

Overview
Where it fits?
What class of users?
Theory foundations.

related: {usability engineering}
categories: <Usability Evaluation>

exploration

Simulate the way users explore and learn
about an interactive system.

related: {cognitive modeling} {learning curve}
categories: <Usability Methods>

discretionary users

Cognitive Walkthrough
good for novices, discretionary, casual users.
assess user success in task completion.
ability to recover from errors.
typically not speed of performance.
do not need to know sequence of operations.

related: {user studies} {error}
categories: <User Profiling>

casual users
Cognitive Walkthrough

good for novices, discretionary, casual users.
assess user success in task completion.
ability to recover from errors.
typically not speed of performance.
do not need to know sequence of operations.

related: {user studies} {error}
categories: <User Profiling>

testing process

Summary of flaws so far.
Potential confusion about need for steps 1 and 2 as part of the fare determination phase.
No feedback on amount deposited.
No means to get money back.
No assessment of fare typing option.

related: {usability evaluation} {error}
categories: <Usability Evaluation>
Appendix B

Analysis of Web Log Data

The website for the course collected a form of augmented web log. We have analysed this data to form one source of user modeling evidence. Essentially, if a learner has ‘attended’ an online lecture, we treat this as evidence supporting the conclusion that they know concepts taught in that lecture. This section describes our analysis of the web logs to model how well each student has ‘attended’ the lecture.

B.1 Data

We are interested in how students listened to and reacted to the content on the site. In particular, we are examining the length of time users stayed on each slide and whether this contributed to their understanding of the material or not.

We have chosen to aggregate the hits to particular slides in a lecture by the length of time a user has spent on it. Each row of the table corresponds to a single slide in that particular online lecture. The columns are described below:

- **Actual time** is the time of the audio (in seconds)
- **Seen** is the proportion of hits to the slide that stayed for less than 10% of the audio length, as a percentage of the total hits.
- **Partial Heard** is the proportion of hits to the slide that stayed more than 10% but less than 80% of the audio length, as a percentage of the total hits.
- **Standard Student** is the proportion of hits to the slide that stayed more than 80% to 150% of the audio length, as a percentage of the total hits.
- **Overheard** is the proportion of hits to the slide that stayed over 150% of the length of the audio.
- Total is the number of hits recorded in the web logs to that slide.

For each student, only their best result is counted. So if a student visits a particular slide twice with times of 15% and 82% of the audio length, we only record the 82% (giving them a rating of Standard Student).

B.1.1 GOMS Online Lecture Data (2003-2005)

Table B.1: Processed Log Data for GOMS Lecture (2003)

<table>
<thead>
<tr>
<th>Slide No.</th>
<th>Actual Time</th>
<th>Seen</th>
<th>Partial Heard</th>
<th>Standard Student</th>
<th>Overheard</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>0%</td>
<td>6%</td>
<td>45%</td>
<td>49%</td>
<td>152</td>
</tr>
<tr>
<td>2</td>
<td>27</td>
<td>0%</td>
<td>3%</td>
<td>60%</td>
<td>37%</td>
<td>145</td>
</tr>
<tr>
<td>3</td>
<td>88</td>
<td>1%</td>
<td>4%</td>
<td>79%</td>
<td>16%</td>
<td>139</td>
</tr>
<tr>
<td>4</td>
<td>42</td>
<td>1%</td>
<td>4%</td>
<td>63%</td>
<td>33%</td>
<td>139</td>
</tr>
<tr>
<td>5</td>
<td>68</td>
<td>1%</td>
<td>6%</td>
<td>70%</td>
<td>24%</td>
<td>136</td>
</tr>
<tr>
<td>6</td>
<td>67</td>
<td>1%</td>
<td>10%</td>
<td>71%</td>
<td>19%</td>
<td>136</td>
</tr>
<tr>
<td>7</td>
<td>81</td>
<td>2%</td>
<td>13%</td>
<td>74%</td>
<td>11%</td>
<td>135</td>
</tr>
<tr>
<td>8</td>
<td>34</td>
<td>0%</td>
<td>5%</td>
<td>62%</td>
<td>33%</td>
<td>133</td>
</tr>
<tr>
<td>9</td>
<td>84</td>
<td>1%</td>
<td>5%</td>
<td>74%</td>
<td>21%</td>
<td>130</td>
</tr>
<tr>
<td>10</td>
<td>379</td>
<td>2%</td>
<td>12%</td>
<td>74%</td>
<td>12%</td>
<td>129</td>
</tr>
<tr>
<td>11</td>
<td>143</td>
<td>5%</td>
<td>7%</td>
<td>67%</td>
<td>21%</td>
<td>129</td>
</tr>
<tr>
<td>12</td>
<td>165</td>
<td>2%</td>
<td>13%</td>
<td>68%</td>
<td>18%</td>
<td>125</td>
</tr>
<tr>
<td>13</td>
<td>47</td>
<td>3%</td>
<td>16%</td>
<td>59%</td>
<td>22%</td>
<td>125</td>
</tr>
<tr>
<td>14</td>
<td>89</td>
<td>6%</td>
<td>19%</td>
<td>59%</td>
<td>16%</td>
<td>125</td>
</tr>
<tr>
<td>15</td>
<td>79</td>
<td>4%</td>
<td>14%</td>
<td>65%</td>
<td>18%</td>
<td>124</td>
</tr>
<tr>
<td>16</td>
<td>170</td>
<td>5%</td>
<td>22%</td>
<td>60%</td>
<td>14%</td>
<td>124</td>
</tr>
<tr>
<td>17</td>
<td>127</td>
<td>6%</td>
<td>23%</td>
<td>62%</td>
<td>9%</td>
<td>121</td>
</tr>
<tr>
<td>18</td>
<td>81</td>
<td>6%</td>
<td>11%</td>
<td>48%</td>
<td>35%</td>
<td>120</td>
</tr>
<tr>
<td>Averages</td>
<td></td>
<td>3%</td>
<td>11%</td>
<td>64%</td>
<td>23%</td>
<td></td>
</tr>
</tbody>
</table>
Table B.2: Processed Log Data for *GOMS* Lecture (2004)

<table>
<thead>
<tr>
<th>Slide No.</th>
<th>Actual Time</th>
<th>Seen</th>
<th>Partial Heard</th>
<th>Standard Student</th>
<th>Overheard</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>0%</td>
<td>9%</td>
<td>50%</td>
<td>41%</td>
<td>110</td>
</tr>
<tr>
<td>2</td>
<td>27</td>
<td>1%</td>
<td>6%</td>
<td>41%</td>
<td>52%</td>
<td>95</td>
</tr>
<tr>
<td>3</td>
<td>88</td>
<td>4%</td>
<td>4%</td>
<td>57%</td>
<td>34%</td>
<td>93</td>
</tr>
<tr>
<td>4</td>
<td>42</td>
<td>3%</td>
<td>5%</td>
<td>36%</td>
<td>55%</td>
<td>91</td>
</tr>
<tr>
<td>5</td>
<td>68</td>
<td>3%</td>
<td>8%</td>
<td>47%</td>
<td>42%</td>
<td>90</td>
</tr>
<tr>
<td>6</td>
<td>67</td>
<td>3%</td>
<td>9%</td>
<td>60%</td>
<td>28%</td>
<td>90</td>
</tr>
<tr>
<td>7</td>
<td>81</td>
<td>7%</td>
<td>10%</td>
<td>57%</td>
<td>26%</td>
<td>89</td>
</tr>
<tr>
<td>8</td>
<td>34</td>
<td>2%</td>
<td>14%</td>
<td>43%</td>
<td>41%</td>
<td>87</td>
</tr>
<tr>
<td>9</td>
<td>84</td>
<td>5%</td>
<td>5%</td>
<td>59%</td>
<td>31%</td>
<td>86</td>
</tr>
<tr>
<td>10</td>
<td>379</td>
<td>7%</td>
<td>3%</td>
<td>63%</td>
<td>27%</td>
<td>86</td>
</tr>
<tr>
<td>11</td>
<td>143</td>
<td>5%</td>
<td>3%</td>
<td>71%</td>
<td>21%</td>
<td>86</td>
</tr>
<tr>
<td>12</td>
<td>165</td>
<td>4%</td>
<td>8%</td>
<td>74%</td>
<td>14%</td>
<td>85</td>
</tr>
<tr>
<td>13</td>
<td>47</td>
<td>5%</td>
<td>11%</td>
<td>54%</td>
<td>31%</td>
<td>85</td>
</tr>
<tr>
<td>14</td>
<td>89</td>
<td>10%</td>
<td>20%</td>
<td>54%</td>
<td>17%</td>
<td>84</td>
</tr>
<tr>
<td>15</td>
<td>79</td>
<td>6%</td>
<td>14%</td>
<td>58%</td>
<td>22%</td>
<td>83</td>
</tr>
<tr>
<td>16</td>
<td>170</td>
<td>7%</td>
<td>14%</td>
<td>61%</td>
<td>17%</td>
<td>83</td>
</tr>
<tr>
<td>17</td>
<td>127</td>
<td>10%</td>
<td>19%</td>
<td>53%</td>
<td>19%</td>
<td>81</td>
</tr>
<tr>
<td>18</td>
<td>81</td>
<td>8%</td>
<td>13%</td>
<td>48%</td>
<td>33%</td>
<td>80</td>
</tr>
<tr>
<td><strong>Averages</strong></td>
<td></td>
<td>5%</td>
<td>10%</td>
<td>55%</td>
<td>31%</td>
<td></td>
</tr>
</tbody>
</table>
Table B.3: Processed Log Data for GOMS Lecture (2005)

<table>
<thead>
<tr>
<th>Slide No.</th>
<th>Actual Time</th>
<th>Seen</th>
<th>Partial Heard</th>
<th>Full Heard</th>
<th>Overheard</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>0%</td>
<td>8%</td>
<td>42%</td>
<td>49%</td>
<td>83</td>
</tr>
<tr>
<td>2</td>
<td>27</td>
<td>0%</td>
<td>10%</td>
<td>45%</td>
<td>45%</td>
<td>69</td>
</tr>
<tr>
<td>3</td>
<td>88</td>
<td>6%</td>
<td>3%</td>
<td>63%</td>
<td>28%</td>
<td>68</td>
</tr>
<tr>
<td>4</td>
<td>42</td>
<td>3%</td>
<td>3%</td>
<td>51%</td>
<td>43%</td>
<td>67</td>
</tr>
<tr>
<td>5</td>
<td>68</td>
<td>4%</td>
<td>4%</td>
<td>52%</td>
<td>39%</td>
<td>67</td>
</tr>
<tr>
<td>6</td>
<td>67</td>
<td>3%</td>
<td>3%</td>
<td>72%</td>
<td>22%</td>
<td>67</td>
</tr>
<tr>
<td>7</td>
<td>81</td>
<td>3%</td>
<td>6%</td>
<td>67%</td>
<td>24%</td>
<td>66</td>
</tr>
<tr>
<td>8</td>
<td>34</td>
<td>0%</td>
<td>6%</td>
<td>48%</td>
<td>45%</td>
<td>66</td>
</tr>
<tr>
<td>9</td>
<td>84</td>
<td>2%</td>
<td>5%</td>
<td>55%</td>
<td>39%</td>
<td>66</td>
</tr>
<tr>
<td>10</td>
<td>379</td>
<td>2%</td>
<td>5%</td>
<td>68%</td>
<td>26%</td>
<td>65</td>
</tr>
<tr>
<td>11</td>
<td>143</td>
<td>5%</td>
<td>2%</td>
<td>63%</td>
<td>31%</td>
<td>64</td>
</tr>
<tr>
<td>12</td>
<td>165</td>
<td>5%</td>
<td>2%</td>
<td>75%</td>
<td>19%</td>
<td>63</td>
</tr>
<tr>
<td>13</td>
<td>47</td>
<td>3%</td>
<td>3%</td>
<td>52%</td>
<td>41%</td>
<td>63</td>
</tr>
<tr>
<td>14</td>
<td>89</td>
<td>3%</td>
<td>10%</td>
<td>69%</td>
<td>18%</td>
<td>62</td>
</tr>
<tr>
<td>15</td>
<td>79</td>
<td>2%</td>
<td>6%</td>
<td>71%</td>
<td>21%</td>
<td>62</td>
</tr>
<tr>
<td>16</td>
<td>170</td>
<td>3%</td>
<td>7%</td>
<td>74%</td>
<td>16%</td>
<td>61</td>
</tr>
<tr>
<td>17</td>
<td>127</td>
<td>2%</td>
<td>5%</td>
<td>73%</td>
<td>20%</td>
<td>59</td>
</tr>
<tr>
<td>18</td>
<td>81</td>
<td>2%</td>
<td>11%</td>
<td>58%</td>
<td>29%</td>
<td>55</td>
</tr>
<tr>
<td>Averages</td>
<td></td>
<td>3%</td>
<td>6%</td>
<td>61%</td>
<td>31%</td>
<td></td>
</tr>
</tbody>
</table>
## B.1.2 Cognitive Walkthrough Online Lecture Data (2003-2005)

Table B.4: Processed Log Data for *Cognitive Walkthrough Lecture* (2003)

<table>
<thead>
<tr>
<th>Slide No.</th>
<th>Actual Time</th>
<th>Seen</th>
<th>Partial Heard</th>
<th>Standard Student</th>
<th>Overheard</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>68</td>
<td>8%</td>
<td>19%</td>
<td>51%</td>
<td>22%</td>
<td>144</td>
</tr>
<tr>
<td>2</td>
<td>63</td>
<td>4%</td>
<td>16%</td>
<td>66%</td>
<td>14%</td>
<td>133</td>
</tr>
<tr>
<td>3</td>
<td>61</td>
<td>4%</td>
<td>10%</td>
<td>69%</td>
<td>17%</td>
<td>126</td>
</tr>
<tr>
<td>4</td>
<td>58</td>
<td>5%</td>
<td>10%</td>
<td>71%</td>
<td>14%</td>
<td>125</td>
</tr>
<tr>
<td>5</td>
<td>186</td>
<td>6%</td>
<td>10%</td>
<td>69%</td>
<td>15%</td>
<td>124</td>
</tr>
<tr>
<td>6</td>
<td>178</td>
<td>3%</td>
<td>12%</td>
<td>66%</td>
<td>19%</td>
<td>124</td>
</tr>
<tr>
<td>7</td>
<td>69</td>
<td>3%</td>
<td>10%</td>
<td>68%</td>
<td>18%</td>
<td>119</td>
</tr>
<tr>
<td>8</td>
<td>76</td>
<td>3%</td>
<td>9%</td>
<td>68%</td>
<td>20%</td>
<td>117</td>
</tr>
<tr>
<td>9</td>
<td>113</td>
<td>5%</td>
<td>9%</td>
<td>74%</td>
<td>12%</td>
<td>114</td>
</tr>
<tr>
<td>10</td>
<td>77</td>
<td>4%</td>
<td>9%</td>
<td>71%</td>
<td>17%</td>
<td>113</td>
</tr>
<tr>
<td>11</td>
<td>114</td>
<td>5%</td>
<td>17%</td>
<td>65%</td>
<td>12%</td>
<td>113</td>
</tr>
<tr>
<td>12</td>
<td>65</td>
<td>4%</td>
<td>14%</td>
<td>68%</td>
<td>14%</td>
<td>112</td>
</tr>
<tr>
<td>13</td>
<td>84</td>
<td>4%</td>
<td>18%</td>
<td>68%</td>
<td>10%</td>
<td>112</td>
</tr>
<tr>
<td>14</td>
<td>22</td>
<td>2%</td>
<td>15%</td>
<td>37%</td>
<td>46%</td>
<td>112</td>
</tr>
<tr>
<td>15</td>
<td>89</td>
<td>8%</td>
<td>23%</td>
<td>59%</td>
<td>10%</td>
<td>112</td>
</tr>
<tr>
<td>16</td>
<td>91</td>
<td>8%</td>
<td>21%</td>
<td>55%</td>
<td>16%</td>
<td>112</td>
</tr>
<tr>
<td>17</td>
<td>49</td>
<td>5%</td>
<td>24%</td>
<td>54%</td>
<td>16%</td>
<td>112</td>
</tr>
<tr>
<td>18</td>
<td>60</td>
<td>6%</td>
<td>21%</td>
<td>69%</td>
<td>4%</td>
<td>112</td>
</tr>
<tr>
<td>19</td>
<td>17</td>
<td>0%</td>
<td>16%</td>
<td>34%</td>
<td>50%</td>
<td>112</td>
</tr>
<tr>
<td>20</td>
<td>101</td>
<td>14%</td>
<td>17%</td>
<td>56%</td>
<td>13%</td>
<td>112</td>
</tr>
<tr>
<td>21</td>
<td>104</td>
<td>7%</td>
<td>13%</td>
<td>63%</td>
<td>17%</td>
<td>112</td>
</tr>
<tr>
<td>22</td>
<td>47</td>
<td>2%</td>
<td>10%</td>
<td>66%</td>
<td>23%</td>
<td>111</td>
</tr>
<tr>
<td>23</td>
<td>155</td>
<td>8%</td>
<td>24%</td>
<td>53%</td>
<td>15%</td>
<td>109</td>
</tr>
<tr>
<td><strong>Averages</strong></td>
<td></td>
<td>5%</td>
<td>15%</td>
<td>62%</td>
<td>18%</td>
<td></td>
</tr>
</tbody>
</table>
Table B.5: Processed Log Data for Cognitive Walkthrough Lecture (2004)

<table>
<thead>
<tr>
<th>Slide No.</th>
<th>Actual Time</th>
<th>Seen</th>
<th>Partial Heard</th>
<th>Standard Student</th>
<th>Overheard</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>68</td>
<td>4%</td>
<td>8%</td>
<td>60%</td>
<td>29%</td>
<td>141</td>
</tr>
<tr>
<td>2</td>
<td>63</td>
<td>1%</td>
<td>10%</td>
<td>60%</td>
<td>29%</td>
<td>136</td>
</tr>
<tr>
<td>3</td>
<td>61</td>
<td>0%</td>
<td>6%</td>
<td>62%</td>
<td>32%</td>
<td>133</td>
</tr>
<tr>
<td>4</td>
<td>58</td>
<td>3%</td>
<td>5%</td>
<td>72%</td>
<td>21%</td>
<td>131</td>
</tr>
<tr>
<td>5</td>
<td>186</td>
<td>3%</td>
<td>9%</td>
<td>64%</td>
<td>24%</td>
<td>129</td>
</tr>
<tr>
<td>6</td>
<td>178</td>
<td>3%</td>
<td>12%</td>
<td>61%</td>
<td>24%</td>
<td>126</td>
</tr>
<tr>
<td>7</td>
<td>69</td>
<td>2%</td>
<td>5%</td>
<td>68%</td>
<td>25%</td>
<td>124</td>
</tr>
<tr>
<td>8</td>
<td>76</td>
<td>3%</td>
<td>7%</td>
<td>71%</td>
<td>19%</td>
<td>121</td>
</tr>
<tr>
<td>9</td>
<td>113</td>
<td>4%</td>
<td>6%</td>
<td>71%</td>
<td>19%</td>
<td>119</td>
</tr>
<tr>
<td>10</td>
<td>77</td>
<td>3%</td>
<td>7%</td>
<td>73%</td>
<td>18%</td>
<td>118</td>
</tr>
<tr>
<td>11</td>
<td>114</td>
<td>3%</td>
<td>13%</td>
<td>74%</td>
<td>11%</td>
<td>118</td>
</tr>
<tr>
<td>12</td>
<td>65</td>
<td>2%</td>
<td>5%</td>
<td>68%</td>
<td>25%</td>
<td>118</td>
</tr>
<tr>
<td>13</td>
<td>84</td>
<td>4%</td>
<td>8%</td>
<td>71%</td>
<td>17%</td>
<td>117</td>
</tr>
<tr>
<td>14</td>
<td>22</td>
<td>1%</td>
<td>9%</td>
<td>41%</td>
<td>49%</td>
<td>116</td>
</tr>
<tr>
<td>15</td>
<td>89</td>
<td>7%</td>
<td>16%</td>
<td>63%</td>
<td>14%</td>
<td>115</td>
</tr>
<tr>
<td>16</td>
<td>91</td>
<td>5%</td>
<td>16%</td>
<td>68%</td>
<td>11%</td>
<td>114</td>
</tr>
<tr>
<td>17</td>
<td>49</td>
<td>3%</td>
<td>19%</td>
<td>58%</td>
<td>20%</td>
<td>114</td>
</tr>
<tr>
<td>18</td>
<td>60</td>
<td>4%</td>
<td>9%</td>
<td>68%</td>
<td>19%</td>
<td>114</td>
</tr>
<tr>
<td>19</td>
<td>17</td>
<td>1%</td>
<td>9%</td>
<td>38%</td>
<td>53%</td>
<td>114</td>
</tr>
<tr>
<td>20</td>
<td>101</td>
<td>9%</td>
<td>10%</td>
<td>55%</td>
<td>26%</td>
<td>114</td>
</tr>
<tr>
<td>21</td>
<td>104</td>
<td>4%</td>
<td>14%</td>
<td>63%</td>
<td>19%</td>
<td>115</td>
</tr>
<tr>
<td>22</td>
<td>47</td>
<td>1%</td>
<td>9%</td>
<td>66%</td>
<td>24%</td>
<td>115</td>
</tr>
<tr>
<td>23</td>
<td>155</td>
<td>7%</td>
<td>17%</td>
<td>64%</td>
<td>12%</td>
<td>115</td>
</tr>
</tbody>
</table>

Averages: 3% 10% 63% 23%
Table B.6: Processed Log Data for Cognitive Walkthrough Lecture (2005)

<table>
<thead>
<tr>
<th>Slide No.</th>
<th>Actual Time</th>
<th>Seen</th>
<th>Partial Heard</th>
<th>Standard Student</th>
<th>Overheard</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>68</td>
<td>0%</td>
<td>7%</td>
<td>55%</td>
<td>38%</td>
<td>108</td>
</tr>
<tr>
<td>2</td>
<td>63</td>
<td>1%</td>
<td>3%</td>
<td>71%</td>
<td>25%</td>
<td>102</td>
</tr>
<tr>
<td>3</td>
<td>61</td>
<td>2%</td>
<td>3%</td>
<td>59%</td>
<td>36%</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>58</td>
<td>1%</td>
<td>4%</td>
<td>66%</td>
<td>29%</td>
<td>99</td>
</tr>
<tr>
<td>5</td>
<td>186</td>
<td>1%</td>
<td>9%</td>
<td>66%</td>
<td>25%</td>
<td>93</td>
</tr>
<tr>
<td>6</td>
<td>178</td>
<td>2%</td>
<td>5%</td>
<td>68%</td>
<td>25%</td>
<td>93</td>
</tr>
<tr>
<td>7</td>
<td>69</td>
<td>1%</td>
<td>8%</td>
<td>72%</td>
<td>19%</td>
<td>93</td>
</tr>
<tr>
<td>8</td>
<td>76</td>
<td>0%</td>
<td>5%</td>
<td>77%</td>
<td>17%</td>
<td>93</td>
</tr>
<tr>
<td>9</td>
<td>113</td>
<td>2%</td>
<td>3%</td>
<td>77%</td>
<td>17%</td>
<td>92</td>
</tr>
<tr>
<td>10</td>
<td>77</td>
<td>0%</td>
<td>7%</td>
<td>74%</td>
<td>20%</td>
<td>91</td>
</tr>
<tr>
<td>11</td>
<td>114</td>
<td>2%</td>
<td>6%</td>
<td>69%</td>
<td>24%</td>
<td>89</td>
</tr>
<tr>
<td>12</td>
<td>65</td>
<td>0%</td>
<td>7%</td>
<td>72%</td>
<td>22%</td>
<td>88</td>
</tr>
<tr>
<td>13</td>
<td>84</td>
<td>1%</td>
<td>8%</td>
<td>67%</td>
<td>24%</td>
<td>87</td>
</tr>
<tr>
<td>14</td>
<td>22</td>
<td>0%</td>
<td>6%</td>
<td>59%</td>
<td>36%</td>
<td>87</td>
</tr>
<tr>
<td>15</td>
<td>89</td>
<td>1%</td>
<td>20%</td>
<td>67%</td>
<td>13%</td>
<td>87</td>
</tr>
<tr>
<td>16</td>
<td>91</td>
<td>3%</td>
<td>10%</td>
<td>72%</td>
<td>14%</td>
<td>87</td>
</tr>
<tr>
<td>17</td>
<td>49</td>
<td>1%</td>
<td>13%</td>
<td>68%</td>
<td>18%</td>
<td>87</td>
</tr>
<tr>
<td>18</td>
<td>60</td>
<td>0%</td>
<td>9%</td>
<td>79%</td>
<td>11%</td>
<td>87</td>
</tr>
<tr>
<td>19</td>
<td>17</td>
<td>0%</td>
<td>9%</td>
<td>48%</td>
<td>43%</td>
<td>87</td>
</tr>
<tr>
<td>20</td>
<td>101</td>
<td>6%</td>
<td>8%</td>
<td>67%</td>
<td>20%</td>
<td>87</td>
</tr>
<tr>
<td>21</td>
<td>104</td>
<td>0%</td>
<td>10%</td>
<td>74%</td>
<td>16%</td>
<td>87</td>
</tr>
<tr>
<td>22</td>
<td>47</td>
<td>0%</td>
<td>2%</td>
<td>72%</td>
<td>26%</td>
<td>86</td>
</tr>
<tr>
<td>23</td>
<td>155</td>
<td>6%</td>
<td>14%</td>
<td>62%</td>
<td>19%</td>
<td>86</td>
</tr>
</tbody>
</table>

Averages 1%  8%  68%  23%
## B.1.3 Summary of Online Lecture Access Averages (2003-2005)

### Table B.7: Summary of Averages (2003)

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Seen</th>
<th>Partial Heard</th>
<th>Standard Student</th>
<th>Overheard</th>
<th>Min. Accesses</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Centered Design</td>
<td>5%</td>
<td>8%</td>
<td>66%</td>
<td>21%</td>
<td>143</td>
</tr>
<tr>
<td>Screen Design Background</td>
<td>4%</td>
<td>15%</td>
<td>64%</td>
<td>17%</td>
<td>128</td>
</tr>
<tr>
<td>Screen Design Text</td>
<td>3%</td>
<td>14%</td>
<td>63%</td>
<td>20%</td>
<td>128</td>
</tr>
<tr>
<td>Screen Design Colour</td>
<td>2%</td>
<td>12%</td>
<td>53%</td>
<td>32%</td>
<td>118</td>
</tr>
<tr>
<td>Cognitive Walkthrough</td>
<td>5%</td>
<td>15%</td>
<td>62%</td>
<td>18%</td>
<td>109</td>
</tr>
<tr>
<td>Heuristic Evaluation</td>
<td>5%</td>
<td>18%</td>
<td>54%</td>
<td>24%</td>
<td>108</td>
</tr>
<tr>
<td>GOMS</td>
<td>3%</td>
<td>11%</td>
<td>64%</td>
<td>23%</td>
<td>120</td>
</tr>
<tr>
<td>Ergonomics</td>
<td>7%</td>
<td>12%</td>
<td>59%</td>
<td>22%</td>
<td>92</td>
</tr>
<tr>
<td>Extended Cognitive Walkthrough</td>
<td>3%</td>
<td>11%</td>
<td>54%</td>
<td>32%</td>
<td>83</td>
</tr>
</tbody>
</table>

### Table B.8: Summary of Averages (2004)

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Seen</th>
<th>Partial Heard</th>
<th>Standard Student</th>
<th>Overheard</th>
<th>Min. Accesses</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Centered Design</td>
<td>4%</td>
<td>8%</td>
<td>59%</td>
<td>29%</td>
<td>98</td>
</tr>
<tr>
<td>Screen Design Background</td>
<td>5%</td>
<td>13%</td>
<td>61%</td>
<td>21%</td>
<td>80</td>
</tr>
<tr>
<td>Screen Design Text</td>
<td>4%</td>
<td>10%</td>
<td>63%</td>
<td>23%</td>
<td>74</td>
</tr>
<tr>
<td>Screen Design Colour</td>
<td>2%</td>
<td>10%</td>
<td>52%</td>
<td>36%</td>
<td>98</td>
</tr>
<tr>
<td>Cognitive Walkthrough</td>
<td>3%</td>
<td>10%</td>
<td>63%</td>
<td>23%</td>
<td>115</td>
</tr>
<tr>
<td>Heuristic Evaluation</td>
<td>5%</td>
<td>13%</td>
<td>55%</td>
<td>27%</td>
<td>100</td>
</tr>
<tr>
<td>GOMS</td>
<td>5%</td>
<td>10%</td>
<td>55%</td>
<td>31%</td>
<td>80</td>
</tr>
<tr>
<td>Ergonomics</td>
<td>5%</td>
<td>10%</td>
<td>55%</td>
<td>29%</td>
<td>71</td>
</tr>
<tr>
<td>Extended Cognitive Walkthrough</td>
<td>4%</td>
<td>16%</td>
<td>43%</td>
<td>38%</td>
<td>27</td>
</tr>
</tbody>
</table>
Table B.9: Summary of Averages (2005)

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Seen</th>
<th>Partial Heard</th>
<th>Standard Student</th>
<th>Overheard</th>
<th>Min. Accesses</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Centered Design</td>
<td>5%</td>
<td>5%</td>
<td>61%</td>
<td>29%</td>
<td>93</td>
</tr>
<tr>
<td>Screen Design Background</td>
<td>5%</td>
<td>13%</td>
<td>61%</td>
<td>21%</td>
<td>88</td>
</tr>
<tr>
<td>Screen Design Text</td>
<td>3%</td>
<td>6%</td>
<td>68%</td>
<td>23%</td>
<td>81</td>
</tr>
<tr>
<td>Screen Design Colour</td>
<td>0%</td>
<td>4%</td>
<td>61%</td>
<td>35%</td>
<td>68</td>
</tr>
<tr>
<td>Cognitive Walkthrough</td>
<td>1%</td>
<td>8%</td>
<td>68%</td>
<td>23%</td>
<td>86</td>
</tr>
<tr>
<td>Heuristic Evaluation</td>
<td>2%</td>
<td>10%</td>
<td>62%</td>
<td>26%</td>
<td>78</td>
</tr>
<tr>
<td>GOMS</td>
<td>3%</td>
<td>6%</td>
<td>61%</td>
<td>31%</td>
<td>55</td>
</tr>
<tr>
<td>Ergonomics</td>
<td>3%</td>
<td>5%</td>
<td>61%</td>
<td>31%</td>
<td>53</td>
</tr>
<tr>
<td>Extended Cognitive Walkthrough</td>
<td>1%</td>
<td>6%</td>
<td>54%</td>
<td>39%</td>
<td>22</td>
</tr>
</tbody>
</table>

B.2 Analysis

We have analysed the nine online lectures that cover the theoretical aspects of interface design. There are some interesting features of our data. We have been able to identify some trends described in more detail below.

All of the online lectures follow a similar trend in the number hits to each slide as the lecture progresses. The first slide has a very high number of hits compared to any other slide in the lecture. The number of hits to the slides then gradually decreases till they reach a stable number. This is more than likely due to the fact that the students start the lecture and lose interest and stop early, or are just curious as to what the lecture is about, visiting the first few slides. The profile data on the website indicated to students whether they have visited the first slide of the lecture or not; so some students appear to have visited just the first page to get that check next to their name.

One of the striking things to be observed in all the GOMS and Cognitive Walkthrough lectures across the years, and the other lectures we have analysed, is the generally stable proportions of visits in each of the categories. With the exceptions we have already discussed, the proportions of each visit category duration seem very stable across slides. The values also correlate with the column averages shown at the bottom of each table. This seems to reflect individual user's patterns of use of these resources.
Appendix C

UI-SIV Evaluation Instructions

This appendix contains screenshots of UI-SIV and the tutorial tasks given to participants of the evaluation presented in Chapter 8. There are a total of 10 tasks with a mixture of quantitative and qualitative activities. These are presented in 0to C.10

We also include the news announcement on the UIDP website that informed students of the availability of the learner model during the stuvac week.
C.1 Task 0: Introduction

0. Introduction

This frame gives a brief tutorial to guide you through the interface.
The SIV display on the left shows your user model for UIDP. The course concepts appear down the display.
Some concepts are in a larger font. The one in the largest font is selected. At startup it should be Usability.

- Select My User Model now and have a look at your user model.

Figure C.1: Screenshot of Task 0: Introduction.
C.2 Task 1: Colour

The next question introduces users to the notion of colour representing the level of their knowledge according to the evidence collected about them. Users are asked to select concepts they know well. Participants should be able to look at their user model and provide a short list of concepts that appear in a green colour on their user model. Since users all have slightly different user models, the list of actual green concepts will be different for each user.

1. Colour
The colour shows your "knowledge", based on evidence, i.e. assessments, web site usage and tutorial marks.
- Concepts that are green are ones that you know well. Concepts in red are ones you do not know well, or we have not learnt in the course yet (or we may not have evidence for it). Enter some concepts you know well below:

![Screenshot of Task 1: Colour](image)

**Figure C.2: Screenshot of Task 1: Colour**
C.3 Task 2: Font Size

This question introduces the notion of font sizes representing the degree of relatedness between the selected concept and the other visible concepts on the visualisation. It is left up to the participant to infer that the blurred out concepts are ones that are not related to the currently selected topic.

2. Font Size

When a concept is selected, the font sizes of other concepts change. The larger the font size, the more related the concept is to the one you selected.

- Enter some related concepts to the currently selected one:

![Screenshot for Task 2: Font Size](image)

Figure C.3: Screenshot for Task 2: Font Size.
C.4 Task 3: Searching

Participants are introduced to the search function as a way to find concepts not immediately visible on the screen. Participants should be able to locate the search button and enter the term GOMS, and select the appropriate concept from the search results list.

3. Searching

We can search for concepts not directly visible using the Search button.

- Try searching for the concept GOMS. Click on Search at the top of the visualisation, type GOMS and press Enter.
- Now we see the search results. Click on GOMS to focus on it.

![Screenshot for Task 3: Searching.](image)
C.5 Task 4: Evidence

This question asks participants to examine the evidence for the concept GOMS. At this stage, the tutorial marks for the GOMS concept have not been entered into the system yet. Participants are made aware of this fact, so they should only see the audio contribution for this concept. This question introduces the idea of evidence accretion from different sources.

4. Evidence
Show evidence for this concept will display all the evidence in the system to support the score given for this concept. These are based on the website hits and tutorial marks during the semester.
- View the evidence for the concept GOMS. Notice the score is low because the tutorial mark has not been recorded yet. What is your score for GOMS? Write it in the box below.

![Figure C.5: Screenshot for Task 4: Evidence.](image-url)
C.6 Task 5: Selecting a concept

Participants are asked to select concepts using one of the buttons above the visualisation. Selection moves topics to the left of the visualisation so that they are more noticeable. This question asks participants to select concepts that the system believes they’re not performing well in, and they are asked to comment on whether the system has modeled their knowledge of those selected concepts correctly or not.

5. Selecting a concept

You may select concepts in the visualisation by clicking the Select/Deselect button. This makes them easier to find if you need to go back to them quickly. Selecting will move the topic to the left of the visualisation. Deselecting will move it back to its original position.

- Select some concepts you are not doing well in based on the score. Do you think the scores here reflect your knowledge of these concepts? Why/why not?

Figure C.6: Screenshot for Task 5: Selecting a concept.
C.7 Task 6: Comparing yourself to the average

This question asked students to compare themselves to the average of the class, and make some observations about their own performance compared to the class average. Students should have identified (if applicable) any subjects that appeared red, i.e. their personal score was less than the average of the scores of other students in the class.

6. Comparing yourself to the average
The Me vs. Average at the top of the screen shows your score compared to the average of the class. In this case, the class average was generated by averaging the scores of students with good attendance from two classes. If you are performing above average, then the concept will show up in green, if you are below average it will show up in red.
- Press the Me vs. Average button now. Write below some concepts you are performing worse than average:

![Screenshot](image.png)

**Figure C.7:** Screenshot for Task 6: Comparing yourself to the average.
C.8 Task 7: The Class Average

This question introduces the notion of group models. Participants are asked to firstly select the class average, and then input some concepts that the class as a whole knows well. Participants should be able to correlate that the colour and font sizes on the group model have the same meaning as those seen on their individual model, except this time the visualization shows an aggregation of the class.

7. The Class Average
We can view the class average without comparing it to your user model. Click on the Class Average button up the top. This shows the averaged scores for the classes.
- Enter below some concepts the class is doing well in:
- Click on the My User Model button to show your own user model again.

![Figure C.8: Screenshot for Task 7: The class Average.](image)
C.9 Task 8: Inferring about knowledge

This question asks users to use the infer functionality. It is expected that the participants should be able to choose a topic they do not know well and see how the related terms can contribute to its score.

8. Inferring about knowledge

Even though concepts show up red (such as through lack of evidence), there is a chance you may know it because you know the related terms quite well. SIV takes this into account through the Infer button. Pressing this will result in the system examining the scores for the related concepts to the one in focus, and see if it contributes any positive points to your score.

- Select a concept that you do not know well.
- Now click the Infer button. See how the colour has changed.
- Click on Show evidence for this concept.
- Enter below the value of inferred score contributes to your final score for this concept?

Figure C.9: Screenshot for Task 8: Inferring about knowledge.
C.10 Task 9: Further comments

Participants were invited to add their own comments on the interface, which would prove useful in the next iteration of the SIV interface.

9. Further comments

Thank you for doing this small tutorial. We hope you can incorporate SIV to help you study for UIDP.
Please enter any additional comments below:
C.11 Stuvac Free Usage

During the stuvac week, we posted a news item on the course website informing students that their learner models had been updated:

_In week 13 most of you had the chance to examine your own User Model for the course using the SIV interface. A version has been made available for you to use outside of University. Follow the links on your profile. The snapshot of your User Model has been updated as of 17th June 2005. As the snapshot probably won't be updated again before the exam, you might need to infer about some of the concepts to get a better representation of how well the system thinks you are doing._

This was posted five days before the exam, although SIV was still available for the students to use since week 13. We envisioned that students would take the opportunity to look at their model in the days before the exam to ensure they had a good understanding of the whole course.
Appendix D

UI-SIV Evaluation Raw Results and Processing

D.1 Processing of Task Results

We came up with a simple formula to measure their interaction levels with SIV. Firstly, we graded their answers or actions in response to each question on the following scale:

- -1: the question was not attempted
- 0: attempted, but performed incorrectly
- 1: performed correctly in pragmatic sense
- 2: performed correctly as asked in the question

In the case of the grade of 1 for a question answer, we considered the cases where the participants might have performed the correct series of actions but did not provide an answer, or provided an answer to the question but with a different context. For example, Question 3 of the tutorial asks participants to search for the concept GOMS and walks them through the process. However, a participant might choose to search for the concept colour instead, but still achieve analogous results that fit with the spirit of the question.

In some cases, participants took an opportunity to explore their learner models as they learnt more about the functionality of the SIV interface while working through the tutorial. For example, after learning about how to search for a concept in Question 3 of the tutorial, the participant might choose to search for some other concepts that they want to know how they are performing in, before returning to the tutorial. We grade the exploration level of students on three levels:
• 0: the student did none or minimal exploration of the model (0-1 action)

• 1: the student did a moderate amount of exploration (2-5 actions)

• 2: the student did a lot of exploration (5+ actions)

The exploration level was counted on two different contexts, firstly, did they explore as they worked through the tutorial questions (and in this case, the exploration was directly related to the context of the question). Second, we examined their exploration of the user model outside of the tutorial question context. In this case, it was commonly before or after doing the tutorial questions.

Finally, we counted the size (in kilobytes) of the log files for the students on their return visits to their user model during the week before their exam (Stuvac). This ranged from 0 to 4kb. Adding these scores together we arrive at the following formula for determining interaction level:

\[ \text{Interaction} = \sum \text{(question scores)} + \sum \text{(exploration scores)} + \text{(return visit log file size)} \]
D.2 Interaction Scores and Exam Marks

The table below shows the processed results for each task (T9 excluded) and the final interaction level as per the formula given in D.1 above.

Table D.1: Summary of SIV Interaction Scores and Exam Marks

<table>
<thead>
<tr>
<th>Participant</th>
<th>Exam Mark</th>
<th>T0</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
<th>T6</th>
<th>T7</th>
<th>T8</th>
<th>Explored SIV Outside of Questions</th>
<th>Explored SIV with Questions</th>
<th>UI-SIV Legible Size in Study (ke)</th>
<th>Interaction level with UI-SIV</th>
</tr>
</thead>
<tbody>
<tr>
<td>p01</td>
<td>53</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>p02</td>
<td>51</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>-7</td>
</tr>
<tr>
<td>p03</td>
<td>55</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>p04</td>
<td>62</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>2</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>p05</td>
<td>50</td>
<td>2</td>
<td>-1</td>
<td>-1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>p06</td>
<td>40</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>p07</td>
<td>61</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>19</td>
</tr>
<tr>
<td>p08</td>
<td>35</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>p09</td>
<td>45</td>
<td>2</td>
<td>0</td>
<td>-1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>p10</td>
<td>56</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>p11</td>
<td>62</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>p12</td>
<td>65</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>p13</td>
<td>41</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>2</td>
<td>0</td>
<td>-7</td>
</tr>
<tr>
<td>p14</td>
<td>54</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>p15</td>
<td>44</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>p16</td>
<td>54</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>-3</td>
</tr>
<tr>
<td>p17</td>
<td>44</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>-1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>p18</td>
<td>54</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>p19</td>
<td>33</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>p20</td>
<td>48</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>-1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>18</td>
</tr>
<tr>
<td>p21</td>
<td>39</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>p22</td>
<td>35</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>p23</td>
<td>50</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>p24</td>
<td>43</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>-1</td>
<td>-1</td>
<td>2</td>
<td>2</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>p25</td>
<td>58</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>18</td>
</tr>
<tr>
<td>p26</td>
<td>50</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>19</td>
</tr>
<tr>
<td>p27</td>
<td>48</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>p28</td>
<td>27</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>p29</td>
<td>33</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>p30</td>
<td>48</td>
<td>2</td>
<td>2</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>p31</td>
<td>61</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>p32</td>
<td>51</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>p33</td>
<td>52</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>p34</td>
<td>42</td>
<td>2</td>
<td>-1</td>
<td>-1</td>
<td>2</td>
<td>-1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>p35</td>
<td>60</td>
<td>2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-6</td>
</tr>
<tr>
<td>p36</td>
<td>66</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>16</td>
</tr>
<tr>
<td>p37</td>
<td>38</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>2</td>
<td>-1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

243
### D.3 Interaction levels and exam marks (stuvac participants)

**Table D.2: Summary of SIV Interaction Scores and Exam Marks (stuvac participants)**

<table>
<thead>
<tr>
<th>Participant</th>
<th>Exam Mark</th>
<th>T0</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
<th>T6</th>
<th>T7</th>
<th>T8</th>
<th>Explored SIV Outside of Questions</th>
<th>Explored SIV with Questions</th>
<th>UI-SIV Logfile Size in Stuvac (kb)</th>
<th>Interaction level with UI-SIV</th>
</tr>
</thead>
<tbody>
<tr>
<td>S01</td>
<td>58</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>18</td>
<td>1</td>
</tr>
<tr>
<td>S02</td>
<td>70</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>16</td>
<td>2</td>
</tr>
</tbody>
</table>
D.4 Summaries of Participant Interactions

A quantitative commentary for all the participants who used UI-SIV during the Week 12 tutorial session is presented below. For each participant, we also include the following notes:

Graph Region: The region of the Interaction vs. Exam Mark graph (Figure 8.3) that the participant was in. A summary of the regions is provided in Table 8.6.

Tutorial Task: A general overview of how the participant performed. Actual results can be seen in Table D.1.

Learner Model Exploration: We indicate whether the participant actively explored the model or only did minimal interactions to complete the tutorial tasks.

Signs of Reflection: A general indication of reflection is whether participants followed up after clicking on a concept with an examination of the evidence that contributed to the score. We are interested in any signs of reflection outside of the examination of evidence required to answer the tutorial task questions.

Commentary: A summary of the participant’s interactions based on the log data we automatically acquired as they accessed UI-SIV during the Week 12 tutorial session.

The participants are ordered by their participant number in the format Pxx, where xx is their number.

P01
Graph Region: 2
Tutorial Task: Mostly Correct
Learner Model Exploration: Yes
Signs of Reflection: Seems Reflective
Commentary: User follows tutorial instructions relatively closely. They initially explore mainly concepts that are red. When asked for concept they don't know, they search for Think Aloud rather than picking one that is visible. Participant was slightly confused at start due to interface, but found it useful. They inferred about a number of concepts during that question. It seems reflective as they explored things they did not know and did search. They also persevered past interface unfamiliarity.

P02
Graph Region: 4
Tutorial Task: Skipped
Learner Model Exploration: Yes
Signs of Reflection: Non-Reflective
Commentary: Initially participant spends a lot of time playing with the interface and selects concepts they know well. They spend a lot of time changing the number of visible concepts on the display, and then skip doing the questions. Does not seem reflective; the user seems more engaged with exploring the interface rather than the learner model itself.

P03
Graph Region: 1
Tutorial Task: All Correct
Learner Model Exploration: Yes
Signs of Reflection: Seems Reflective
Commentary: Participant follows instructions very closely. Mainly examine concepts they do not know well. Participant lists multiple concepts when asked to find concepts they know/don't know - they spend a noticeable amount of time thinking after a sequence of actions.

P04
Graph Region: 5
Tutorial Task: Mostly Correct, Some Skipped
Learner Model Exploration: Yes
Signs of Reflection: Non-Reflective
Commentary: Participant skips initial questions but does some of the actions (search). For the last few questions the user does not give answers but does the actions. No exploration of SIV beyond what is asked. Does not seem reflective, as it appears user is just examining the interface itself and finishing the tutorial tasks.

P05
Graph Region: 5
Tutorial Task: Mostly Correct, Some Skipped
Learner Model Exploration: Yes
Signs of Reflection: Non-Reflective
Commentary: Participant skips answering questions but does what is asked. Explores what select and infer buttons do after being introduced to search button. Participant then skips rest of tutorial and does no further exploration. Does not seem reflective, as it appears user is just examining the interface itself.

P06
Graph Region: 4
Tutorial Task: Mostly Correct, Some Skipped
Learner Model Exploration: Yes
Signs of Reflection: Seems Reflective
Commentary: They start with exploring the visualisation, clicking on concepts and changing the terms visible. Plays around with depth change more while answering questions 1 to 4, but leaves after that.
P07

**Graph Region:** 2

**Tutorial Task:** Mostly Correct

**Learner Model Exploration:** Yes

**Signs of Reflection:** Non-Reflective

**Commentary:** Follows instructions closely. They made use of the inference function before arriving at the task on about inference. They examine a lot of evidence, but mostly as a reaction to changing the user model view. User stated interface was hard to read, but still managed to do a large proportion of instructed actions. This seems non-reflective as a lot of the actions seem to be toying with the system and interface more than concentrating on their own performance in the course.

P08

**Graph Region:** 3

**Tutorial Task:** Mostly Correct

**Learner Model Exploration:** No

**Signs of Reflection:** Non-Reflective

**Commentary:** User explores different views before start of tutorial and ends up on wrong view until tutorial two when they change back. User didn't explore outside of the scope of the questions. Their actions seem non-reflective as they appear to be focused only on completing the tutorial.

P09

**Graph Region:** 4

**Tutorial Task:** All Correct

**Learner Model Exploration:** Yes

**Signs of Reflection:** Non-Reflective

**Commentary:** Works through tasks correctly but does minimal exploration. Examines concepts that are red but does not scrutinise them.

P10

**Graph Region:** 2

**Tutorial Task:** All Correct

**Learner Model Exploration:** No

**Signs of Reflection:** Non-Reflective

**Commentary:** Does questions correctly, but does not explore. Does not scrutinise evidence for concepts except for when the tutorial asked them to. This user does not seem to reflect, and only follows the instructions on the tutorial.

P11

**Graph Region:** 2

**Tutorial Task:** Mostly Correct
Learner Model Exploration: Yes
Signs of Reflection: Seems Reflective
Commentary: Initially spends a long time exploring the depth changes and the views as they click on concepts. Does large amount of clicking and selecting for tutorial task 5, selecting on concepts they don't know well, and believes some of them are modelled correctly.

P12
Graph Region: 1
Tutorial Task: All Correct
Learner Model Exploration: Yes
Signs of Reflection: Reflective
Commentary: Clicks and scrutinises a lot of concepts. User is unclear of evidence for concepts. User follows tutorial closely and does all the steps. Then explores interface after, searching for think aloud and playing with the depth. User finds think aloud and infers about it, then examines related concept empirical evaluation and also infers.

P13
Graph Region: 5
Tutorial Task: Skipped
Learner Model Exploration: Yes
Signs of Reflection: Seems Reflective
Commentary: User does a lot of clicking on different (mainly red) concepts. User also scrutinises some of these but does not engage in tutorial. Seems to be reflective due to their selection of particular concepts and subsequent examination of evidence.

P14
Graph Region: 1
Tutorial Task: Mostly Correct
Learner Model Exploration: Yes
Signs of Reflection: Seems Reflective
Commentary: User is quite engaged with tutorial and does actions correctly. Explores using functionality they have learnt about throughout tutorial. They infer about a large number of concepts and also scrutinise them.

P15
Graph Region: 3
Tutorial Task: Mostly Correct
Learner Model Exploration: No
Signs of Reflection: Non-Reflective
Commentary: User works through tutorial with minimal exploration and has no problems. No comments to qualitative questions.
P16

**Graph Region:** 5  
**Tutorial Task:** Skipped  
**Learner Model Exploration:** Yes  
**Signs of Reflection:** Seems Less Reflective  
**Commentary:** User plays with model by clicking and changing depth before starting tutorial. Clicks and scrutinises concepts but does not engage in tutorial after task 1. Possibly reflective as they do examine the evidence for concepts they click on, despite non-compliance at tutorial tasks.

P17

**Graph Region:** 3  
**Tutorial Task:** Mostly Correct  
**Learner Model Exploration:** Yes  
**Signs of Reflection:** Seems Reflective  
**Commentary:** Explores features in SIV as they do questions but does not provide answers to them despite doing the correct actions. User actively scrutinises concepts they click on, an indication they were curious about them.

P18

**Graph Region:** 5  
**Tutorial Task:** Mostly Correct, Some Skipped  
**Learner Model Exploration:** Yes  
**Signs of Reflection:** Seems Reflective  
**Commentary:** Plays with depth change and scrutinises some clicked concepts before doing tutorial. Abandons tutorial after task 4 and plays with model more, depth change and then click and scrutinise concepts. Their actions seem possibly reflective as they went out of their way to explore evidence for concepts after skipping tutorial tasks.

P19

**Graph Region:** 4  
**Tutorial Task:** Mostly Incorrect  
**Learner Model Exploration:** No  
**Signs of Reflection:** Seems Reflective  
**Commentary:** Tried to engage with tutorial but made a lot of mistakes (suspect language barrier with this participant). Seemed generally confused but managed to complete some parts of tutorial. Interaction and possibilities for reflection seem hampered by language barrier.

P20

**Graph Region:** 2
Tutorial Task: Mostly Incorrect
Learner Model Exploration: Yes
Signs of Reflection: Seems Reflective
Commentary: Does a lot of playing and exploration initially and works through questions incorrectly. Clicks a lot of concepts but does not scrutinise (possibly because user realises there is little or no evidence). This seems somewhat reflective, but possibly hampered by the fact there was little evidence in the learner model for this user.

P21
Graph Region: 3
Tutorial Task: Mostly Correct
Learner Model Exploration: Yes
Signs of Reflection: Seems Reflective
Commentary: Plays with depth change and clicks on a few concepts before doing tutorial. Does tutorial steps relatively closely. Explores a lot (clicking and scrutinising concepts) for task 5, spends a lot of time clicking on concepts but not scrutinising at the end of the tutorial.

P22
Graph Region: 5
Tutorial Task: Mostly Correct
Learner Model Exploration: Yes
Signs of Reflection: Non-Reflective
Commentary: Works through tutorial tasks and spends time exploring early on as they do it. Logs show a lot of clicking but no examination of evidence. There was minimal exploration during last few questions. Does not seem reflective, they were just doing the tutorial tasks.

P23
Graph Region: 2
Tutorial Task: Mostly Correct
Learner Model Exploration: No
Signs of Reflection: Non-Reflective
Commentary: This user explores but does not scrutinise much. Tutorial done pretty much correctly but no comment left on tutorial task 9. Does not seem reflective, possibly more interested in just doing tutorial questions.

P24
Graph Region: 4
Tutorial Task: Mostly Correct, Some Skipped
Learner Model Exploration: Yes
Signs of Reflection: Non-Reflective
Commentary: Clicks lots of different concepts during first few questions but does not scrutinise. User works through the last half of the tutorial with minimal exploration.

P25
Graph Region: 2
Tutorial Task: Mostly Correct
Learner Model Exploration: No
Signs of Reflection: Non-Reflective
Commentary: Works through tutorial quickly and relatively correct. Minimal exploration outside of what was required. Their actions do not seem reflective; they only seem engaged in exploring for the purpose of answering the tutorial questions.

P26
Graph Region: 2
Tutorial Task: All Correct
Learner Model Exploration: Yes
Signs of Reflection: Seems Less Reflective
Commentary: User works through tutorial correctly, user model is primarily red. They do a lot of clicking but minimal examination of evidence.

P27
Graph Region: 4
Tutorial Task: Mostly Correct
Learner Model Exploration: No
Signs of Reflection: Seems Less Reflective
Commentary: Works through tutorial quickly with minimal exploration but mostly correct actions. Could be reflective, but they seem more engaged with completing the tutorial tasks.

P28
Graph Region: 5
Tutorial Task: Mostly Incorrect, Some Skipped
Learner Model Exploration: Yes
Signs of Reflection: Non-Reflective
Commentary: Works through first few questions (incorrectly). Abandons tutorial to explore and scrutinise evidence. Non-compliance with tutorial task indicates a lack of engagement.

P29
Graph Region: 4
Tutorial Task: Mostly Incorrect
Learner Model Exploration: No
**Signs of Reflection:** Seems Reflective  
**Commentary:**  
User does not seem engaged with the tutorial. Learner model is primarily red and user believes it doesn't necessarily represent their knowledge. User does not explore the evidence for the red concepts. Their actions seem somewhat reflective as they comment that the learner model does not reflect their knowledge, indicating they provided some thought into what they think they should have learnt or known.

**P30**  
**Graph Region:** 4  
**Tutorial Task:** Mostly Incorrect  
**Learner Model Exploration:** Yes  
**Signs of Reflection:** Seems Reflective  
**Commentary:** Starts by playing with navigation and clicks on a lot of concepts and changes learner model view. They work through the tutorial tasks but many answers are incorrect. Does some exploration and inference, mainly clicking but does not explore the evidence. They seem somewhat reflective but the lack of scrutinising and lack of correctness on questions suggests they are not really engaged.

**P31**  
**Graph Region:** 1  
**Tutorial Task:** Mostly Correct  
**Learner Model Exploration:** No  
**Signs of Reflection:** Seems Reflective  
**Commentary:** Starts by clicking and scrutinising several concepts at the start. Works through tutorial relatively closely and provides answers that are mainly correct. User examines a lot of concepts with no evidence and claims the score of 0 is incorrect for them (need for inference?). User works through rest of tutorial very quickly with minimal exploration. Seems to indicate user was in a hurry to finish tutorial and didn’t engage in the latter parts of the tutorial tasks.

**P32**  
**Graph Region:** 2  
**Tutorial Task:** All Correct  
**Learner Model Exploration:** No  
**Signs of Reflection:** Non-Reflective  
**Commentary:** Works through tutorial quickly with minimal exploration but actions and answers done correctly. User does not engage in much exploration outside of the tasks. This does not seem reflective as user seems only engaged in answering the tutorial tasks.

**P33**  
**Graph Region:** 2  
**Tutorial Task:** Mostly Correct
Learner Model Exploration: Yes
Signs of Reflection: Non-Reflective
Commentary: Early on plays with depth change and clicks on concepts but does not scrutinise. When they are up to task 5, they start to scrutinise as well and also plays with depth change. Infers about a lot of concepts but does not scrutinise them. Click and infer over a range of green and red concepts, no real pattern to clicking or inference. Does not seem reflective, they seem to be toying with the interface.

P34
Graph Region: 4
Tutorial Task: Mostly Correct
Learner Model Exploration: No
Signs of Reflection: Non-Reflective
Commentary: Works through tutorial quickly with relatively correct answers. They do not provide any qualitative feedback. When asked about concepts that were red, user listed concepts not in learner model. Actions do not seem reflective as user appears to be concerned about finishing the tutorial tasks quickly.

P35
Graph Region: 5
Tutorial Task: Skipped
Learner Model Exploration: No
Signs of Reflection: Non-Reflective
Commentary: User did one question of tutorial (changing the view) and then left. Essentially did not use the system.

P36
Graph Region: 2
Tutorial Task: Mostly Correct
Learner Model Exploration: Yes
Signs of Reflection: Non-Reflective
Commentary: User does a lot of clicking on mostly green concepts at several points during tutorial but does not scrutinise. Even at tutorial task 8, does a lot of clicking before settling on a concept to infer about (while selecting concepts that are red). They seem engaged with the tutorial tasks but their actions do not seem reflective.

P37
Graph Region: 5
Tutorial Task: Mostly Correct
Learner Model Exploration: No
Signs of Reflection: Non-Reflective
Commentary: Goes through tutorial without doing anything first, and then starts again. Minimal exploration of learner model, relatively correct actions to questions.
### D.5 Interactions during stuvac

Table D.3: Summary of participant interactions during stuvac

<table>
<thead>
<tr>
<th>Action Types</th>
<th>Examine</th>
<th>Infer</th>
<th>Search</th>
<th>Select</th>
<th>Deselect</th>
<th>Depth</th>
<th>C-me</th>
<th>C-vs</th>
<th>C-ave</th>
<th>Click</th>
</tr>
</thead>
<tbody>
<tr>
<td>P14</td>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>27</td>
<td>7</td>
<td>7</td>
<td>6</td>
<td>57</td>
</tr>
<tr>
<td>S01</td>
<td>23</td>
<td>25</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>51</td>
</tr>
<tr>
<td>S02</td>
<td>13</td>
<td>17</td>
<td>3</td>
<td>7</td>
<td>3</td>
<td>9</td>
<td>2</td>
<td>1</td>
<td></td>
<td>42</td>
</tr>
<tr>
<td>S03</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>57</td>
</tr>
<tr>
<td>P24</td>
<td>15</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>63</td>
</tr>
<tr>
<td>S04</td>
<td>11</td>
<td>1</td>
<td>13</td>
<td>3</td>
<td>1</td>
<td>7</td>
<td>2</td>
<td>1</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>S05</td>
<td>9</td>
<td>3</td>
<td>7</td>
<td>1</td>
<td>10</td>
<td>43</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P04</td>
<td>7</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>49</td>
<td></td>
</tr>
<tr>
<td>S06</td>
<td>21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>29</td>
</tr>
<tr>
<td>P06</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>P03</td>
<td>15</td>
<td>2</td>
<td>1</td>
<td>7</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>S07</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td>S08</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>11</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>17</td>
</tr>
<tr>
<td>P01</td>
<td>9</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td>13</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>S09</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S10</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>2</td>
<td>8</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S11</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>17</td>
</tr>
<tr>
<td>S12</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>S13</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td>1</td>
<td></td>
<td></td>
<td>17</td>
</tr>
<tr>
<td>S14</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>S15</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>S16</td>
<td>6</td>
<td></td>
<td>6</td>
<td>2</td>
<td></td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S17</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>S18</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>18</td>
</tr>
<tr>
<td>S19</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S20</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>S21</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S22</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>S23</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S24</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S25-S26</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S27-S32</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S33-S48</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The table includes all participants who accessed UI-SIV during stuvac. There are six return visitors (those who did the tutorial in Week 13) and 48 participants visiting for the first time. The table is ranked by number of interactions. The columns correspond to the following actions:

**Examine**: examine or scrutinise the currently focused concept

**Infer**: use inference on the currently focused concept

**Select**: select the currently focused concept

**Deselect**: deselect the currently focused concept

**Depth**: changing the depth (amount of concepts visible)

**C-me**: change to view of own learner model

**C-vs**: change to view of own learner model compared to class average

**C-ave**: change to view of class average

**Click**: click on a concept
D.6 UI-SIV and UIDP website log analysis during stuvac

The section contains annotated log file analyses for participants S01, S02, P03, P14 and P24. These are summarised in the case study descriptions in Chapter 8.

D.6.1 Log Analysis: S01

Session started.

Tutorial interactions shown in section D.3 along with calculated Interaction score.

Accesses UI-SIV again (probably closed window after tutorial).

Participant selects and also scrutinises (examines evidence page) for many concepts. These were coloured red in their UI-SIV learner model. They also make use of inference.

Session started.

New-Session (Sat Jun 18 17:22:17 2005)

(did tutorial)

New-Session (Sat Jun 18 17:34:20 2005)

Scrutinise (5:35:35 P.M.)

usability (05:35:45 PM)

user centred design (05:36:17 PM)

Scrutinise (5:36:23 P.M.)

Infer-user centred design (05:36:29 PM)

Scrutinise (5:36:40 P.M.)

intuitive (05:36:45 PM)

short term memory load (05:36:48 PM)

Scrutinise (5:36:52 P.M.)

Infer-short term memory load (05:37:00 PM)

Scrutinise (5:37:06 P.M.)

desktop metaphor (05:37:10 PM)

Scrutinise (5:37:15 P.M.)

Infer-desktop metaphor (05:37:16 PM)

- metaphor (05:38:21 PM)

Scrutinise (5:38:27 P.M.)

Infer-metaphor (05:38:29 PM)

Scrutinise (5:38:33 P.M.)

situation of concern (05:38:39 PM)

heuristic evaluation (05:38:41 PM)

extended cognitive walkthrough (05:38:43 PM)

Infer-extended cognitive walkthrough (05:38:45 PM)

Scrutinise (5:38:49 P.M.)

conceptual design model (05:38:54 PM)

Infer-conceptual design model (05:38:56 PM)

usability (05:38:57 PM)

Infer-usability (05:38:59 PM)

system image (05:39:01 PM)

Infer-system image (05:39:02 PM)

system response (05:39:04 PM)

Inspects a number of concepts relating to Ergonomics and Extended Cognitive Walkthrough (e.g. design coherence, imposed mental model, intended mental model). They use the infer function on each of these concepts.
Week/03###Sat Jun 18 17:58:33 2005###1119081513.46

Week/09###Sat Jun 18 18:08:40 2005###1119082120.18

Ergonomics+1###Sat Jun 18 18:09:09 2005

Ergonomics+19###Sat Jun 18 19:26:22 2005

printable+Ergonomics###Sat Jun 18 19:27:31 2005

Week/10###Sat Jun 18 19:28:02 2005

Week/13###Sat Jun 18 19:30:10 2005

ExCognitive+1###Sat Jun 18 19:30:37 2005

printable+ExCognitive###Sat Jun 18 19:31:07 2005

UI-SIV interactions for this session end.

Returns to content pages on website.

Accesses the weekly pages listing lab work and online lecture topics.

Accesses topic for Week 9 - Ergonomics, attending the whole online lecture topic (slides 1 to 19) over the course of approx. 1hr 15min. Accesses printable version of the slides.

Continues going through rest of weekly pages listing lab work and online lecture topics.

Accesses the printable version of the topic Extended Cognitive Walkthrough.

Session Ends.
**D.6.2 Log Analysis: S02**

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(did tutorial)</td>
<td>Tutorial interactions shown in section D.3 along with calculated Interaction score.</td>
</tr>
<tr>
<td>text principles (12:28:31 PM)</td>
<td>Examined various concepts, relating to Screen Design and User Centered Design learning topics. Also examines inferred values for the concepts.</td>
</tr>
<tr>
<td>colour (12:28:34 PM)</td>
<td></td>
</tr>
<tr>
<td>cognitive modeling (12:28:36 PM)</td>
<td></td>
</tr>
<tr>
<td>text principles (12:28:38 PM)</td>
<td></td>
</tr>
<tr>
<td>layout complexity (12:28:47 PM)</td>
<td></td>
</tr>
<tr>
<td>usability inspection (12:28:48 PM)</td>
<td></td>
</tr>
<tr>
<td>requirements analysis (12:28:57 PM)</td>
<td></td>
</tr>
<tr>
<td>SCRUTINISE (12:29:06 P.M.)</td>
<td></td>
</tr>
<tr>
<td>INFER-requirements analysis (12:29:14 PM)</td>
<td></td>
</tr>
<tr>
<td>SCRUTINISE (12:29:19 P.M.)</td>
<td></td>
</tr>
<tr>
<td>user interface critique (12:29:40 PM)</td>
<td></td>
</tr>
<tr>
<td>INFER-user interface critique (12:29:41 PM)</td>
<td></td>
</tr>
<tr>
<td>INFER-user interface critique (12:29:42 PM)</td>
<td></td>
</tr>
<tr>
<td>SCRUTINISE (12:29:49 P.M.)</td>
<td></td>
</tr>
<tr>
<td>prompt (12:29:53 PM)</td>
<td></td>
</tr>
<tr>
<td>INFER-prompt (12:29:56 PM)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Various actions with the website accessing printable versions of learning topics for programming aspects and Heuristic Evaluation.</td>
</tr>
<tr>
<td>UserCentredDesign+1###Sun Jun 19 15:19:03 2005</td>
<td>Attends the User Centred Design learning topic, does now finish but accesses printable versions of the lecture.</td>
</tr>
<tr>
<td>printable+UserCentredDesign###Sun Jun 19 15:19:05 2005</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Returned after in the day.</td>
</tr>
<tr>
<td>UserCentredDesign+6###Sun Jun 19 15:20:03 2005</td>
<td></td>
</tr>
<tr>
<td>SCRUTINISE (3:58:23 P.M.)</td>
<td>Searches for “catwoe”, this concept was not taught in the course (it was taught in previous years).</td>
</tr>
<tr>
<td>usability (03:58:29 PM)</td>
<td></td>
</tr>
<tr>
<td>SEARCH-catwoe (03:58:38 PM)</td>
<td></td>
</tr>
<tr>
<td>highlighting text (03:58:45 PM)</td>
<td></td>
</tr>
<tr>
<td>the total interface (03:58:47 PM)</td>
<td></td>
</tr>
<tr>
<td>prototyping (03:58:52 PM)</td>
<td></td>
</tr>
<tr>
<td>gestalt principles (03:58:57 PM)</td>
<td></td>
</tr>
<tr>
<td>user intuition (03:59:10 PM)</td>
<td></td>
</tr>
<tr>
<td>user interface design (03:59:13 PM)</td>
<td></td>
</tr>
<tr>
<td>SCRUTINISE (3:59:25 P.M.)</td>
<td>Examines concepts relating to user centred design.</td>
</tr>
</tbody>
</table>
design again.

Accesses learning topics for Screen Design, accessing the printable versions of the lectures.

Attends the online lecture on Screen Design: Text Principles.

Accesses the printable version for the Ergonomics topic, and then attends slides 1 to 6.
### D.6.3 Log Analysis: P03

<table>
<thead>
<tr>
<th>Event Type</th>
<th>Time</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCRUTINISE</td>
<td>11:54:09 A.M.</td>
<td>Changed to the different comparative views of the learner model before changing back to their own values.</td>
</tr>
<tr>
<td>usability</td>
<td>11:54:29 AM</td>
<td>Participant scrutinised various concepts that were red in their learner model, and examined the evidence.</td>
</tr>
<tr>
<td>time</td>
<td>11:56:23 AM</td>
<td>Slides accessed (UI-SIV was active at this time, the participant accessed these online slides via the evidence display in SIV).</td>
</tr>
<tr>
<td>SCRUTINISE</td>
<td>11:56:33 A.M.</td>
<td>participant scrutinised various concepts that were red in their learner model, and examined the evidence.</td>
</tr>
<tr>
<td>performance measure</td>
<td>11:56:40 AM</td>
<td>participant scrutinised various concepts that were red in their learner model, and examined the evidence.</td>
</tr>
<tr>
<td>SCRUTINISE</td>
<td>11:56:50 A.M.</td>
<td>participant scrutinised various concepts that were red in their learner model, and examined the evidence.</td>
</tr>
<tr>
<td>CHANGETOMEVS AVERAGE</td>
<td>11:57:02 A.M.</td>
<td>changed to the different comparative views of the learner model before changing back to their own values.</td>
</tr>
<tr>
<td>CHANGETO AVERAGE</td>
<td>11:57:45 A.M.</td>
<td>changed to the different comparative views of the learner model before changing back to their own values.</td>
</tr>
<tr>
<td>DEPTHCHANGE-1</td>
<td>11:57:53 AM</td>
<td>changed to the different comparative views of the learner model before changing back to their own values.</td>
</tr>
<tr>
<td>DEPTHCHANGE-5</td>
<td>11:57:55 AM</td>
<td>changed to the different comparative views of the learner model before changing back to their own values.</td>
</tr>
<tr>
<td>CHANGETOMEVS AVERAGE</td>
<td>11:57:58 A.M.</td>
<td>changed to the different comparative views of the learner model before changing back to their own values.</td>
</tr>
<tr>
<td>CHANGETO AVERAGE</td>
<td>11:58:03 A.M.</td>
<td>changed to the different comparative views of the learner model before changing back to their own values.</td>
</tr>
<tr>
<td>TUTORIAL0</td>
<td>11:58:11 A.M.</td>
<td>changed to the different comparative views of the learner model before changing back to their own values.</td>
</tr>
<tr>
<td>SELECT-performance</td>
<td>11:58:19 AM</td>
<td>changed to the different comparative views of the learner model before changing back to their own values.</td>
</tr>
<tr>
<td>experience</td>
<td>11:58:28 AM</td>
<td>changed to the different comparative views of the learner model before changing back to their own values.</td>
</tr>
<tr>
<td>SCRUTINISE</td>
<td>11:58:32 A.M.</td>
<td>changed to the different comparative views of the learner model before changing back to their own values.</td>
</tr>
<tr>
<td>performance measure</td>
<td>11:58:39 AM</td>
<td>changed to the different comparative views of the learner model before changing back to their own values.</td>
</tr>
<tr>
<td>SCRUTINISE</td>
<td>11:58:42 A.M.</td>
<td>changed to the different comparative views of the learner model before changing back to their own values.</td>
</tr>
<tr>
<td>speed</td>
<td>12:04:24 PM</td>
<td>changed to the different comparative views of the learner model before changing back to their own values.</td>
</tr>
<tr>
<td>SCRUTINISE</td>
<td>12:04:27 P.M.</td>
<td>changed to the different comparative views of the learner model before changing back to their own values.</td>
</tr>
</tbody>
</table>

**Note:**
The participant leaves UI-SIV, and accesses printable version of Heuristic Evaluation learning topic before ending session.
D.6.4 Log Analysis: P14

---Session Started---###Sat Jun 18 10:46:54 2005
...
NEW-SESSION(Sat Jun 18 11:51:46 2005)
SCRUTINISE(11:51:54 A.M.)
usability(11:52:13 AM)
CHANGETOME(11:52:28 A.M.)
CHANGETOMEVS AVERAGE(11:52:34 A.M.)
CHANGETO AVERAGE(11:52:40 A.M.)
...
white space(11:52:56 AM)
CHANGETOME(11:52:58 A.M.)
SCRUTINISE(11:53:06 A.M.)
the total interface(11:54:06 AM)
...
CHANGETOMEVS AVERAGE(12:05:20 P.M.)
...
NEW-SESSION(Sat Jun 18 22:07:52 2005)
SCRUTINISE(10:07:59 P.M.)
usability(10:08:13 PM)
CHANGETOME(10:11:36 P.M.)
...
Apple Macintosh(10:12:27 PM)

Session started.
Participant accesses various weekly pages of the website listing learning topics and lab activities.
Accesses UI-SIV.
Examines various views of themselves vs. class average and the class average.
During this period, they continue to access the learning content on the website, particularly the printable versions of the online lectures.
Ends interaction with UI-SIV.
They return later in the day for another session, similar activity to earlier in the day, accessing printable versions of the lecture notes while navigating their own model.
### D.6.5 Log Analysis: P24

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Event Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sun Jun 19 00:26:09 2005</td>
<td>Session started. Scrutinises a concept related to GOMS. Visits online lecture slide directly from the link provided on the evidence page in UI-SIV.</td>
</tr>
<tr>
<td>12:26:18 A.M.</td>
<td>usability (12:26:30 AM)</td>
</tr>
<tr>
<td>12:28:45 AM</td>
<td>predictive metric (12:28:52 A.M.)</td>
</tr>
<tr>
<td>Sun Jun 19 00:28:58 2005</td>
<td>Attends the online lecture and appears to listen to audio.</td>
</tr>
<tr>
<td>1:00:18 AM</td>
<td>human-computer interaction (1:00:37 AM)</td>
</tr>
<tr>
<td>1:00:43 A.M.</td>
<td>empirical methods (1:00:43 A.M.)</td>
</tr>
<tr>
<td>Sun Jun 19 01:00:55 2005</td>
<td>Returns to UI-SIV. They look out for red concepts. Here, human-computer interaction was red. They select it, and see a related red concept, empirical methods, which they subsequently scrutinise as well.</td>
</tr>
<tr>
<td>Sun Jun 19 01:31:52 2005</td>
<td>Accesses Screen Design and User Centred Design topics briefly, directly from the link in the UI-SIV scrutinise evidence page.</td>
</tr>
<tr>
<td>1:01:16 AM</td>
<td>UserCentredDesign+1###Sun Jun 19 01:14:16 2005</td>
</tr>
<tr>
<td>1:01:43 A.M.</td>
<td>UserCentredDesign+4###Sun Jun 19 01:15:43 2005</td>
</tr>
<tr>
<td>01:21:05 AM</td>
<td>task completion time (01:21:32 A.M.)</td>
</tr>
<tr>
<td>1:21:56 AM</td>
<td>Know Thy User (1:22:00 A.M.)</td>
</tr>
<tr>
<td>Sun Jun 19 01:21:58 2005</td>
<td>Similar to before, user examines a concept in UI-SIV and goes to the corresponding lecture slide.</td>
</tr>
<tr>
<td>1:21:58 AM</td>
<td>ScreenText+9###Sun Jun 19 01:25:48 2005</td>
</tr>
<tr>
<td>01:26:18 AM</td>
<td>discretionary users (01:26:19 AM)</td>
</tr>
<tr>
<td>01:26:20 AM</td>
<td>documentation (01:26:50 A.M.)</td>
</tr>
<tr>
<td>1:26:50 A.M.</td>
<td>heuristic evaluation (1:27:02 AM)</td>
</tr>
<tr>
<td>1:27:07 A.M.</td>
<td>SCRUTINISE (1:27:07 A.M.)</td>
</tr>
</tbody>
</table>
Returns later in the day and examines concepts while browsing printable notes and weekly summary pages on the course website.
References


272


