



Terms and Conditions of Use of Digitised Theses from Trinity College Library Dublin

Copyright statement

All material supplied by Trinity College Library is protected by copyright (under the Copyright and Related Rights Act, 2000 as amended) and other relevant Intellectual Property Rights. By accessing and using a Digitised Thesis from Trinity College Library you acknowledge that all Intellectual Property Rights in any Works supplied are the sole and exclusive property of the copyright and/or other IPR holder. Specific copyright holders may not be explicitly identified. Use of materials from other sources within a thesis should not be construed as a claim over them.

A non-exclusive, non-transferable licence is hereby granted to those using or reproducing, in whole or in part, the material for valid purposes, providing the copyright owners are acknowledged using the normal conventions. Where specific permission to use material is required, this is identified and such permission must be sought from the copyright holder or agency cited.

Liability statement

By using a Digitised Thesis, I accept that Trinity College Dublin bears no legal responsibility for the accuracy, legality or comprehensiveness of materials contained within the thesis, and that Trinity College Dublin accepts no liability for indirect, consequential, or incidental, damages or losses arising from use of the thesis for whatever reason. Information located in a thesis may be subject to specific use constraints, details of which may not be explicitly described. It is the responsibility of potential and actual users to be aware of such constraints and to abide by them. By making use of material from a digitised thesis, you accept these copyright and disclaimer provisions. Where it is brought to the attention of Trinity College Library that there may be a breach of copyright or other restraint, it is the policy to withdraw or take down access to a thesis while the issue is being resolved.

Access Agreement

By using a Digitised Thesis from Trinity College Library you are bound by the following Terms & Conditions. Please read them carefully.

I have read and I understand the following statement: All material supplied via a Digitised Thesis from Trinity College Library is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of a thesis is not permitted, except that material may be duplicated by you for your research use or for educational purposes in electronic or print form providing the copyright owners are acknowledged using the normal conventions. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone. This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

Authoring Adaptive Soft Skill Simulations

PhD Thesis

Conor Gaffney

Supervised by

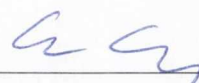
Vincent Wade



Thesis 10072

Declaration

I, the undersigned, declare that this work has not been previously submitted as an exercise for a degree at this or any other University, and that, unless otherwise stated, it is entirely my own work.

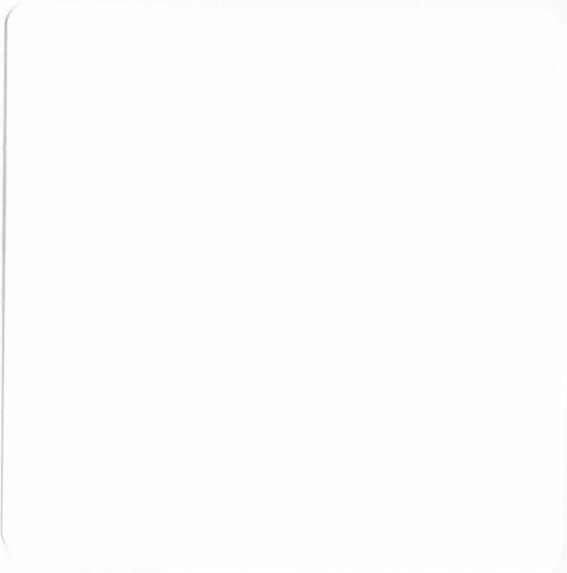


Conor Gaffney

Date: 12/4/13

Permission to lend or copy

I, the undersigned, agree to deposit this thesis in the University’s open access institutional repository or allow the library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.



Acknowledgements

There are a number of people who have helped me on my PhD journey. They have all, in some way, made this thesis and research possible.

Firstly, I would like to thank my supervisor Vincent Wade whose experience, knowledge and guidance has made this work possible; his patience and consideration have been greatly appreciated.

I am extremely grateful to my family and friends for their support and assistance during the course of this research. In particular I would like thank Felicity Wurlod for her understanding and kindness which has both encouraged and motivated me to the very end. I would also like to thank my mother Marie and my father Niall for their unconditional love and for teaching me the importance of education; they have both always encouraged me whatever my goal or interest has been.

I would like to express my gratitude to the members of the Knowledge and Data Engineering Group (KDEG) in Trinity College Dublin; they have always assisted in the smallest query, question and problem that I have had over the years. In particular I would like to thank Declan Dagger, for his insight, expertise and encouragement. Much of this research has also been assisted by members of the Department of Psychiatry in Trinity Medical School, in particular Michael Gill, Brian Fitzmaurice, Katie Armstrong and Cathy Rogers.

Finally, I would like to thank Enterprise Ireland for funding the ADAPT project and continuing to support innovate Irish research.

Go raibh maith agaibh.

Learning is experience. Everything else is just information. — Albert Einstein

Abstract

This thesis examines the personalisation of online training simulations which are a key modern approach in computer aided education. More specifically it focuses on the difficulties involved in authoring personalised training simulations. The composition of such systems is very difficult which has hampered their wide spread adoption. At present adaptive training simulations can only be authored by programmers working closely with subject matter experts. One of the key ways for adaptive simulations to increase their popularity in online eLearning is to reduce the effort and technical skills required by authors in their development. This thesis argues that personalised online simulations need to be composed by subject matter experts, relatively easily and quickly. This thesis details the twin challenges in composing content for both educational simulations and personalisation. It also describes ACTSim, a new and unique composition tool that supports the rapid development of personalised training simulations. In particular ACTSim focuses on situational simulations for inter personal dialogue - so called soft skills. This thesis describes the design development and implementation of ACTSim. It also presents a series of evaluations of the composition tool conducted with real world subject matter experts.

Table of Contents

List of Diagrams and Tablesxiii

List of Abbreviationsxv

1 Introduction 1

1.1 Motivation..... 1

1.2 Research Question 4

1.3 Objectives..... 5

1.4 Contribution to the State of the Art 6

1.5 Technical Approach 9

1.5.1 Overview 9

1.5.2 Educational Underpinnings 11

1.5.3 State of the Art Survey 11

1.5.4 Design and Implementation 11

1.5.5 Evaluation Approach..... 12

1.6 Summary 13

2 Educational Underpinnings of Simulation and Adaptivity Based Learning 14

2.1 Educational Theory..... 14

2.1.1 Constructivism 15

2.1.2 Situated Learning 15

2.1.3 Cognitivism 16

2.2 Simulation Based Learning 16

2.2.1 Categories of Educational Simulations 18

2.2.2 Identification of Key Properties in Educational Simulations..... 20

2.2.3 Consequences of Key Properties within Soft Skill Simulations 27

2.2.4 Educational Theory of Computer Based Educational Simulations..... 29

2.2.5 Application of Educational Theory to Soft Skill Simulations 31

2.3 Adaptivity Based Learning 32

2.3.1 Types of Adaptive Educational Systems 33

2.3.2 Survey of Adaptive Educational Hypermedia 35

2.3.3 Analysis of Adaptive Educational Hypermedia 42

2.3.4 Educational Theory of Adaptive Hypermedia Systems 44

2.3.5 Application of Educational Theory to Next Generation of AEH Systems 45

2.4 Adaptive Educational Simulations..... 45

2.5 Summary 51

3 State Of The Art Survey of Authoring Tools for Adaptive Educational Simulations 53

3.1 Soft Skill Simulation Authoring Tools 53

3.1.1 VISIOOn Composition Tool 54

3.1.2	Experience Builder	57
3.1.3	Captivate 5	62
3.1.4	SimWriter	66
3.1.5	Soft Skill Authoring Tool Critique	70
3.2	Survey of Adaptive Courseware Authoring Tools	76
3.2.1	MOT	77
3.2.2	ACCT	80
3.2.3	Adaptive Courseware Authoring Tool Critique	82
3.3	Summary	83
4	Design	86
4.1	Key Requirements	86
4.1.1	Dialogue Representation	87
4.1.2	Adaptivity Application and Representation	88
4.1.3	Authoring Process	90
4.1.4	Pedagogical Framework	90
4.1.5	Navigational Aids	92
4.1.6	Model Verification	93
4.2	Key Design Elements	94
4.2.1	Authoring Tool Overview and Description	94
4.2.2	Dialogue Representation	94
4.2.3	Adaptivity Application and Representation	97
4.2.4	Authoring Process	107
4.2.5	Pedagogical Framework	112
4.2.6	Navigational Aids	113
4.2.7	Model Verification	116
4.3	Phases of Development	117
4.4	Summary	119
5	Implementation	120
5.1	ACTSim Architecture	120
5.2	Components Architecture View of ACTSim	121
5.2.1	Canvas	122
5.2.2	Dialogue Element	123
5.2.3	Trigger Components	125
5.3	Technological Architecture View	129
5.3.1	GMF Development	129
5.3.2	ACTSim GMF Development	131
5.3.3	ACTSim Graphical Editor	140
5.3.4	ACTSim Authoring Tool	143
5.3.5	Further Functionality Implementation	145

5.4	ACTSim Simulation Title Model	167
5.5	Authored Simulations.....	174
5.6	Summary	177
6	Evaluation.....	178
6.1	Evaluation Overview.....	178
6.1.1	Evaluation Goals	178
6.1.2	Research and Evaluation Process	179
6.1.3	Real World Evaluations	181
6.1.4	Users	181
6.1.5	Soft Skill Simulation Evaluations	184
6.2	Evaluation Types and Timelines	184
6.2.1	Predictive Evaluations.....	185
6.2.2	Interpretive Evaluations	187
6.2.3	Usability Evaluations	189
6.2.4	Development Approach	192
6.2.5	Reflections at the End of Each Phase of Development.....	192
6.3	Phase One of Development.....	193
6.3.1	Phase One Predictive Evaluation	193
6.3.2	Phase One Interpretive Evaluations	196
6.3.3	Phase One Usability Evaluation.....	204
6.3.4	End of Phase One Reflections.....	212
6.4	Phase Two of Development	213
6.4.1	Phase Two Predictive Evaluation	214
6.4.2	Phase Two Interpretive Evaluations	216
6.4.3	Phase Two Usability Evaluation	220
6.4.4	End of Phase Two Reflections	238
6.5	Comparison of ACTSim to Related Research	239
6.6	Summary	243
7	Conclusions	244
7.1	Achievements.....	244
7.2	Objectives.....	246
7.3	Contribution to the State of the Art.....	249
7.4	Future Work	251
7.4.1	Authoring Tool Improvements.....	251
7.4.2	Future Research.....	252
	Bibliography.....	255
	Appendices	282
	Appendix I – Soft Skill Simulations.....	282

Appendix II – Simulation Learning Verbs	285
Appendix III – Annotated Java	286
Dialogue Element.....	286
Assessment Trigger.....	287
Feedback Trigger	288
Reflection Trigger.....	289
Appendix IV – Component Interfaces.....	290
Learning Outcomes Nouns	290
Roles.....	290
Subjects.....	291
Categories.....	291
Appendix V – Usability Definition	292
Appendix VI – Ethical Approval.....	293
Overview.....	293
Ethical Approval Application	293
Appendix VII – Evaluation User Groups	297
Appendix VIII – Storyboards	299
Appendix IX – Composition Tool Sub-Dialogue Feature Comparison	314
Appendix X – First Usability Evaluation	316
Task.....	316
Questionnaire.....	317
Appendix XI – Customer Care Directed Discussion.....	323
Appendix XII – Customer Care Short Questionnaire.....	325
Appendix XIII – Second Usability Evaluation.....	328
Design Document.....	328
Pre-Questionnaire	373
Test.....	375
Post-Questionnaire	377
Error Classification	396
Appendix XIV – Evaluation Results - Model Validation.....	406

List of Diagrams and Tables

Figure 1.1: Technical Approach10

Figure 2.1: Hypothesised Relationship of Fidelity and Learning (Dagger, Rogers, et al. 2007)23

Figure 3.1: VISIOOn Authoring Tool55

Figure 3.2: Partial Screen Shot of the Experience Builder Authoring Tool58

Figure 3.3: Captivate 5 Authoring Tool64

Figure 3.4: SimWriter Authoring Tool67

Figure 3.5: My Online Teacher Authoring Tool79

Figure 3.6: ACCT Authoring Tool81

Figure 4.1: Example of Dialogue Representation96

Figure 4.2: Example of Adapted Dialogue98

Figure 4.3: Example of Dialogue Representation with Triggers100

Figure 4.4: Example of Adapted Dialogue101

Figure 4.5: Constructive Alignment108

Figure 4.6: Pedagogically Informed Process for Composing Adaptive Dialectic Simulations111

Figure 4.7: Phases of Development and Design Elements (Mac Issac 1995)118

Figure 5.1: Overview of Soft Skill Simulation Development and Delivery121

Figure 5.2: ACTSim Architecture122

Figure 5.3: GMF Overview130

Figure 5.4: Annotated Java for Canvas Component132

Figure 5.5: ACTSim Domain Model133

Figure 5.6: Diagram of ACTSim Domain Model134

Figure 5.7: ACTSim Graphical Definition Model136

Figure 5.8: ACTSim Tooling Definition Model137

Figure 5.9: ACTSim Mapping Model139

Figure 5.10: ACTSim Generator Model140

Figure 5.11: ACTSim Authoring Tool144

Figure 5.12: Scenario Properties Interface148

Figure 5.13: Learning Verbs Interface149

Figure 5.14: Learning Outcomes Interface150

Figure 5.15: Dialogue Element Interface152

Figure 5.16: Assessment Trigger Properties Interface154

Figure 5.17: Feedback Trigger Properties Interface156

Figure 5.18: Reflection Trigger Properties Interfac156

Figure 5.19: Subjects Multiple Tagging Interface158

Figure 5.20: Script Function Interface159

Figure 5.21: Highlight Function Interface161

Figure 5.22: Search Function Interface163

Figure 5.23: Validate Function Interface164

Figure 5.24: Full Script Function165

Figure 5.25: Authoring Process Interface166

Figure 5.26: Segment of ACTSim Title File (Beginning of File)168

Figure 5.27: Segment of ACTSim Title File (Dialogue Element Properties Data)169

Figure 5.28: Segment of ACTSim Title File (Dialogue Element Graphical Data)170

Figure 5.29: Segment of ACTSim Title File (Connection Graphical Data171

Figure 5.30: Segment of Transformed ACTSim Title File (Beginning of File)173

Figure 5.31: Segment of Transformed ACTSim Title File (Dialogue Element)174

Figure 5.32: Overview of Simulation Model with Magnified Section and Highlighted Dialogue Element.....	175
Figure 5.33: Simulation Screen Capture: Response	176
Figure 5.34: Simulation Screen Capture: Statement and Response Transcript with 4 Learner Options	177
Figure 6.1: Phases of Development, Design Elements and Action Based Phase Sub-Cycles	180
Figure 6.2: Evaluation Timeline	185
Figure 6.3: Evaluation Timeline – Predictive Evaluations Highlighted	187
Figure 6.4: Evaluation Timeline – Interpretive Evaluations Highlighted.....	188
Figure 6.5: Evaluation Timeline – Usability Evaluations Highlighted.....	191
Figure 6.6: First Cognitive Walkthrough Storyboard – Mockup of Initial Design of ACTSim Authoring Tool	195
Figure 6.7: First Cognitive Walkthrough Storyboard - Mockup of Use of the Zoom Function with Map	196
Figure 6.8: Comparison of Node Editing Box Storyboard and Early Screenshot of ACTSim Dialogue (Bubble) Property Interface.....	199
Figure 6.9: Comparison of Mockup Storyboard (Figure 6.7) and Screenshot of Early ACTSim Prototype (Gaffney, Dagger et al. 2007)	200
Figure 6.10: Phase One ACTSim Prototype Domain Model.....	201
Figure 6.11: Screenshot of Early ACTSim Prototype (Gaffney, Dagger et al. 2007)	203
Figure 6.12: Storyboard Mockup of Dialogue Modelling Process	204
Figure 6.13: Correlated Mean Data of Efficiency and Usability of Dialogue Representation	208
Figure 6.14: Correlated Mean View of Efficiency of Dialogue Model Authoring Process ..	210
Figure 6.15: Correlated Mean View of User Satisfaction of Dialogue Model Authoring Process	211
Figure 6.16: Correlated Mean View of Efficiency of the Navigational Aids	212
Figure 6.17: Second Cognitive Walkthrough Storyboard - Soft Skill Simulation Roles Interface	216
Figure 6.18: Second Cognitive Walkthrough Storyboard - Assessment Trigger Properties Interface	216
Figure 6.19: Comparison of Screenshot of ACTSim Prototype from Phase One to Screenshot of ACTSim Prototype from Phase Two.....	219
Figure 6.20: Comparison of Roles Interface Storyboard (Figure 6.8) and Roles Interface Implementation (Appendix IV – Component Interfaces - Roles).....	219
Figure 6.21: Dialogue Representation Effectiveness Test Results	227
Figure 6.22: Dialogue Representation Effectiveness Questionnaire Results.....	228
Figure 6.23: Adaptivity Effectiveness Test Results.....	229
Figure 6.24: Adaptivity Effectiveness Questionnaire Results	229
Figure 6.25: Mean Time and Error Count for Operations Related to Adaptivity	232
Figure 6.26: Adaptivity User Satisfaction Questionnaire Results	233
Figure 6.27: Authoring Process Usability Questionnaire Results	234
Figure 6.28: Pedagogical Framework Usability Questionnaire Results	235
Table 2.1: Sample of Personalization Dimensions (Wade2009).....	39
Table 3.1: Authoring Tools Surveyed.....	75
Table 6.1: Evaluation Participants.....	183

List of Abbreviations

ACCT - Adaptive Course Construction Toolkit
ADAPT - Adaptive Plug-in for Run-time Composition of Personalised eLearning and Adaptive Simulations
AEH - Adaptive Educational Hypermedia
AH - Adaptive Hypermedia
AHA - Adaptive Hypermedia Architecture
AHAM - Adaptive Hypermedia Application Model
AHS - Adaptive Hypermedia System
AICC - Aviation Industry CBT/eLearning Committee
ASP - Active Server Pages
CAF - Common Adaptation Format
CAPSL - Centre for Learning Technology
ELEKTRA - Enhanced Learning Experience and Knowledge TRAnsfer
EMF - Eclipse Modelling Framework
FBI - Federal Bureau of Investigation
GEF - Graphical Editing Framework
GMF - Graphical Modelling Framework
GRAPPLE - Generic Responsive Adaptive Personalised Learning Environment
HCI - Human Computer Interaction
HCT - Hematocrit
HTML - HyperText Markup Language
IDE - Integrated Development Environment
ISO - International Organization for Standardization
ITS - Intelligent Tutoring Systems
LAG - Layers of Adaptation Granularity
LAOS - Layered WWW AHS Authoring Model and their corresponding Algebraic Operators
LLC - Limited Liability Company
MOT - My Online Teacher
NDLR - National Digital Learning Repository
PC - Personal Computer
RCP - Rich Client Platform
SCORM - Sharable Content Object Reference Model
SWT - Standard Widget Toolkit
URI - Uniform Resource Identifier
WWW - World Wide Web
WYSWYG - What You See is What You Get
XSLT - Extensible Stylesheet Language Transformations

1 Introduction

1.1 Motivation

Online educational simulations can be very effective and efficient method of instruction (Zunzarren and Rodriguez-Sedano 2011; Margo, Swarz, et al. 2010; Alessi and Trollip 2001). They bridge the gap between theory and application by providing a safe and convenient environment for learners to practice their skills (Uibu and Kikas 2012; Alessi and Trollip 2001; Widdison, Aikenhead et al. 1998). The learner is not a passive subject within their learning experience but instead is engaged by the simulation and becomes an active participant in their own educational process (Tanner, Stewart et al. 2012; Barab, Hay et al. 2000; Conrick 1998).

Computer based simulations are not only a productive form of learning, they are also much more cost effective than real world simulations which require expensive equipment or have large recurring overheads (Vázquez-Salceda, Ceccaroni et al. 2009). Online simulations are however still expensive to develop compared to other eLearning solutions (Ramachandran, Sincoff et al. 2011; Alden 1998). They can be very complex and time consuming to author and are typically a “*one-off*” project developed by a team of programmers.

The focus of this research was to potentially reduce the complexity involved in authoring educational simulations* and allow them to be implemented by non-technical subject matter experts. Furthermore, the authoring process developed also allows the composition of simulations which are adaptive. Adaptive eLearning, in the traditional sense, has shown it can reduce development cost (Nkambou, Bordeau et al. 2011; Heffernan, Turner, et al. 2006; Conlan, Wade et al. 2002) and encourage more engagement as personalisation allows for more repeated use. Adaptivity in simulation based education potentially offers similar benefits. However, a key challenge in realising adaptive simulation based learning is the complexity and time needed to produce them.

There are many different types of training simulations. This research has focused on a particular category of educational simulation known as situational simulations (Hertzner and

* Authoring of educational simulations can include the composition/design of training simulations and simulation experiences – these are referred to interchangeably in this thesis.

Pannese 2011; Alessi and Trollip 2001). Typically this category of simulations focuses on soft skills. Soft skill simulations are generally used to teach skills based on interpersonal relationships which involve the learner taking on a role within the simulation. The learner is taught through the process of interacting with simulated people and scenarios. Soft skill simulations are typically used to teach communication skills such as customer care, interpersonal management, sales and negotiation.

The traditional real world approach to teaching these types of skills typically involves hiring actors to play the roles within the simulations. This approach is very expensive as human actors are associated with high costs for training, delivery and management (Reams and Bashford 2011; Odhayani and Ratnapalan 2011; Hubal, Kizakevich et al. 2000). Furthermore, hired actors only allow the learners to practice their skills a very limited number of times during a course. Alternative approaches incorporate the learners themselves play the roles within the simulations in order to try and teach one another. This can be a very ineffective approach as the learners will have little or no previous experience regarding the situations being simulated. The learners are ill equipped to teach something that they themselves are attempting to learn and training the learners to play these roles would also require a great deal of effort and resources.

Research has shown that, although initially resource intensive to develop, electronic soft skill simulations can be considerably less expensive than hiring actors. For example, The Virtual Standardized Patient (Bergin and Fors 2003), a hand crafted soft skill simulation system, required less than 5% of its programme code to be modified to create new cases. These types of online electronic simulations also have the advantage of being easily accessed by the students. Learners can practice their skills over multiple iterations at their own convenience in a safe environment (Aebersold and Tschannen 2012; Pilkington and Parker-Jones 1996).

There are, however, a number of reasons why electronic simulations have not yet been widely adopted. Simulations are complex and difficult to compose (Rocha, Araujo, et al. 2010; Van Joolingen and De Jong 2003); they take considerable time to develop compared to other eLearning solutions (Alessi and Trollip 2001); content is not typically reusable (Rocha, Araujo, et al. 2010; Fish and Harter 1991); and deployment is often limited as they are difficult to integrate (Rocha, Araujo, et al. 2010; Moreno-Ger, Blesius et al. 2008). In

addition to these shortcomings, educational simulations also suffer from another key impediment. With only a very few exceptions, computer based simulations do not offer any kind of adaptivity and rarely customise to a learners contextual needs (so called personalisation).

The approach of personalisation has been widely accepted in the area of eLearning courseware where it has been very successful at improving the educational effectiveness of electronic courses (Brusilovsky 2012; Mustafa and Sharif 2011; Conlan, Dagger et al. 2002; Brusilovsky 1996). However, it has yet to be widely adopted in the area of educational simulations and for the most part simulations still adhere to “*one size fits all*”. A key challenge in mainstreaming adaptive eLearning is the difficulty and complexity in design and implementation associated with authoring (Nurjanah and Davis 2012; Cristea and Stewart 2006).

There are several reasons why personalisation has proven to be an effective approach to eLearning solutions. Primarily, the learning experiences focus on the individual learning requirements and competencies of the learner (Walkington and Maull 2011; Calvi and Bra 1997) so the learning experience becomes more suited to the learners needs. The content also becomes more contextual so the learner is taught more efficiently (Brusilovsky 2012; Brusilovsky 2003). Finally the learner finds the experience much more engaging (Dagger 2006; Shang, Shi et al. 2001).

As well as potentially improving the educational effectiveness of the simulations, adaptivity can also create a more compelling experience. Computer based educational simulations often become repetitive and do not progress as the learner develops (Sweat 2006; Pagulayan, Keeker et al. 2003). This is an issue which particularly effects soft skill simulations and is alleviated with the introduction of adaptivity.

There are a relatively small number of educational simulations that do offer limited adaptivity. These simulations generally base adaptivity on the very approximate estimation of a users experience or expertise (Lopez and Bidarra 2011; Noble 2002); for example novice, intermediate and expert. While this can be considered an improvement over traditional non-adaptive educational simulations, there is a need for adaptivity which is much more

sophisticated. A key aspect of this research is to provide adaptivity which is not limited to past learner experience but also considers other factors. When considering soft skill simulations this adaptivity could, for example, be based on the role of the learner, learning outcomes the learner is to achieve or subjects areas that they are to cover.

Simulations that do incorporate adaptivity tend to do so in a rigid and inflexible manner. Adaptivity is often hard coded into the simulation, making it difficult to incorporate within composition tools (Drossos, Giotopoulos et al. 2005; Connolly, Johnson et al. 1998). To allow the effective composition of adaptive simulations a new approach was required. This approach, suggested in this thesis, was to separate adaptivity from the simulation content. This allows adaptivity to be incorporated into the composition of the simulations. Furthermore, this separation allows adaptivity to become extensible and the content to be more reusable.

There are, to date, no authoring tools[†] that allow subject matter experts to compose adaptive soft skill simulations. The goal of this research is to research and develop a composition tool that allows non-technical authors to rapidly and easily develop soft skill simulations that can adapt to the needs of individual learners.

1.2 Research Question

This research addresses the question of what are the appropriate adaptive models, mechanisms and authoring processes for adaptive soft skill simulations. In particular it examines how adaptive soft skill simulations can be authored by non-technical designers using a model driven approach. The research defines an authoring process which includes a pedagogical framework to encourage proper pedagogical practice. The process is also iterative, which allows the composition of the adaptive models upon which the soft skill simulations operate to be more effective. The research also examines how designers with a non-technical background are capable of authoring the models required for adaptive soft skill simulations. Formally put:

[†] The terms 'authoring tools' and 'composition tools' are interchangeable in the domains of training simulations and eLearning. As such, both are used interchangeably throughout this thesis.

‘What are the models, mechanisms and authoring processes required for adaptive soft skill simulation[‡] and how can adaptive features for soft skill simulations be authored by non-technical designers?’

In the area of soft skill simulations an important part of the simulation systems are the models which describe their behaviour. Typically, soft skill simulations employ branching models based on dialogue to describe the logic upon which they operate (Emmendorfer 2009); these dialogue models are sometimes realised during composition as graphs. Non-adaptive simulation dialogue models are static and deliver an identical learning experience to each learner. Soft skill simulations also typically employ learning events, interventions and reflective prompts during their execution which are also non-adaptive.

A key challenge of this research will be to allow non-technical subject matter experts to compose simulations so both the dialogue models and learning events can adapt to the needs of individual learners. Part of this challenge will also be identifying the control mechanisms that might be used to define how this personalisation can be applied. Within the area of Adaptive Hypermedia (Brusilovsky 2012) ‘*adaptive dimensions*’ (Harrigan, Kravcik et al. 2009) are sometimes used to control and define personalisation. Such learner and content descriptors, suitable to soft skill simulations, might offer an approach to realising adaptivity in this research.

1.3 Objectives

In answering the proposed research question this thesis presents the design, implementation and evaluation of a composition tool for authoring personalised soft skill simulations. The goals of which are:

1. Analyse the state of the art of soft skill simulation authoring tools. Identify the key gaps in the current support for adaptive soft skill simulations. Derive the requirements for the adaptive models, authoring process and authoring systems.
2. Refine and design the models that are required for composing adaptive soft skill simulations. In particular, models which will support adaptive flow of the training

[‡] Soft skill simulations are training simulations which take place in the context of social situations

simulation, models that will allow the delivery of adaptive interventions and finally models that will support adaptive dimensions that are specific to soft skill simulations.

3. Research and develop an authoring process for composing adaptive soft skill simulations which include the educational underpinnings of simulation based learning.
4. Research and develop a tool to enable non-technical authors to create the relevant models. The tool must also support the processes which allow the development of this very ad-hoc form of learning. The tool should also include the support for the authoring of the associated processes and mechanisms which are the underpinnings for simulation based learning; in particular the inclusion of adaptivity.
5. Validate and evaluate the adaptive models, the authoring process and the simulation composition system developed. In particular, to evaluate approaches for applying adaptivity to dialogue models and triggers which incorporate relevant adaptive dimensions. Employ user based trials with soft skill subject matter experts for all evaluations.

1.4 Contribution to the State of the Art

There are two distinct contributions to the state of the art of authoring personalised simulations attributed to this work. The main contribution to the state of the art is a composition tool that offers and supports *personalised* authoring of soft skill simulations for non-technical subject matter experts. Hitherto, adaptivity in soft skill simulations has only really been possible through handcrafting and programming. This research supports personalisation in soft skill simulations by three different means; i.e. two methods of applying adaptivity to soft skill simulations and a new and unique approach to adaptive dimensions so as they are suitable for soft skill simulations.

The first method of applying adaptivity is a novel approach and is related to the flow and content of the simulation. The few existing educational simulations that incorporate personalisation embed the adaptivity in the actual content. For example, Microsoft's most recent flight simulator, called Microsoft Flight (Microsoft Flight 2012), includes "*customized flight controls to match [a users] skill level*". Similarly, there is no separation of adaptivity

and content in the interview training simulator developed for the FBI (Olsen, Sellers et al. 1999). The approach proposed in this thesis separates the content and the adaptivity with the author initially developing the models upon which the simulations operate and later introducing the adaptive elements. By applying adaptivity across the models only after they have been completed it is insured that content and adaptivity remain separate. This separation allows the adaptivity to become extensible and the content to be more reusable.

The second approach to applying adaptivity for non-technical subject matter experts is through a concept known as '*adaptive triggers*'. These are triggered learning events that occur in the simulation and can be specified for educational strategies or events which are included in the simulation. Educational simulations not only need to present a realistic interpretation of the world, they also need to adhere to standard educational strategies. These strategies include assessing the learner's progress, presenting the learner with feedback and allowing the learner to reflect on the content of the simulation. While these principals may be employed with some educational simulations, this thesis illustrates a new and unique approach whereby the firing of these events can be adaptive. Furthermore, the content which is contained in these events is also personalised to the learner. Learners can be presented with feedback that is based on the decisions they have made during the course of the simulation.

The unique approach to adaptive dimensions is the third aspect of personalisation in which this research contributes to the state of the art. Typically, educational adaptivity and personalisation dimensions have been defined with generic pedagogical characteristics and learner attributes. This research realises adaptive dimension which are related to a specific sub-domain of educational simulations, i.e. situational or soft skill simulations. This research defines four adaptive dimensions based on the learner (role), the dialogue (related subjects), the types of interaction (categories) and educational aspects (learning outcomes).

The second contribution to the state of the art described in this thesis is a rigorous authoring process for composing adaptive educational simulations. Many composition tools incorporate an authoring process but most are very limited and offer little, if any, real support to the author. The authoring process presented in this thesis is a comprehensive framework, which fully supports the author and is based upon well-defined eLearning and simulation composition best practices. The process not only includes steps which allow the author to

fully define adaptivity across the content of the simulation but also defines a pedagogical framework to insure the simulations that are composed are educationally viable.

The contribution of this research to the state of the art is evident in the author's publications. The following works present the peer reviewed and published foundation for this thesis:

- Gaffney, C., D. Dagger, V. Wade (2010). "*Authoring and Delivering Personalised Simulations — an Innovative Approach to Adaptive eLearning for Soft Skills.*" Journal of Universal Computer Science: Special Issue on Authoring of Adaptive and Adaptable Hypermedia, Vol. 16 Issue 19. Available online at http://www.jucs.org/jucs_16_19/authoring_and_delivering_personalised/. Last Checked: 22 September 2011. J.UCS.
- Gaffney, C., D. Dagger, V. Wade (2008). "*Evaluation of ACTSim: a Composition Tool for Authoring Adaptive Soft Skill Simulation.*" 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Hannover, Germany, July 2008. Springer.
- Gaffney, C., D. Dagger, V. Wade (2008). "*A State of the Art Survey of Soft Skill Simulation Authoring Tools.*" 19th ACM Conference on Hypertext and Hypermedia, Pittsburgh, USA, June 2008. ACM.
- Gaffney, C., D. Dagger, V. Wade (2007). "*Supporting personalised simulations: a pedagogic support framework for modelling and composing adaptive dialectic simulations.*" Summer Computer Simulation Conference, San Diego, USA, July 2007. Available online at <http://dl.acm.org/citation.cfm?id=1358093&bnc=1>. Last Checked: 22 September 2011. ACM.
- Gaffney, C., D. Dagger, V. Wade (2007). "*Towards a Composition Tool for Personalised Simulations.*" A3H: Fifth International Workshop on Authoring of Adaptive and Adaptable Hypermedia, 11th International Conference on User Modeling, Corfu, Greece, June 2007. Springer.

The research described in this thesis has also been recognised as being innovative by the Irish Software Association as the author was awarded their national student award in 2008.

1.5 Technical Approach

Developing an authoring process and composition tool used to compose adaptive soft skill simulations is difficult and complex. This section describes the technical approach employed within this research. This section initially describes an overview of the technical approach which is followed by a description of the different components described in the overview.

1.5.1 Overview

The first step of this research was to examine the educational underpinnings in personalised simulations. This was followed by a state of the art survey which examined both soft skill simulation authoring tools and adaptivity based eLearning composition tools. The remainder of the technical approach is divided into two phases, as presented in Figure 1.1, both of which follow similar patterns. Both are phases composed of a cycle of design, implementation and evaluation which were iterated through many times. The final iteration of each cycle concluded with a specific type of evaluation known as a usability evaluation; explained in further detail below.

The technical approach was divided into two phases so as to accomplish two separate objectives. The objective of the first phase of development was to create an authoring tool that allowed the development of basic soft skill simulations i.e. non-adaptive soft skill simulations. The main objective of the second phase of development was to extend the authoring tool so as to allow adaptive simulations to be composed. This two phase approach was employed so as to allow the complexities of the composing soft skill simulations to be initially considered. Once these requirements were satisfied and thoroughly evaluated, features employed to capture adaptivity could be added in the second phase of development. This approach also enabled the design of adaptivity to be made in a more informed manner. The second phase of development was also used to extend further the basic functionality of the composition tool but its primary focus was the incorporation of adaptive features.

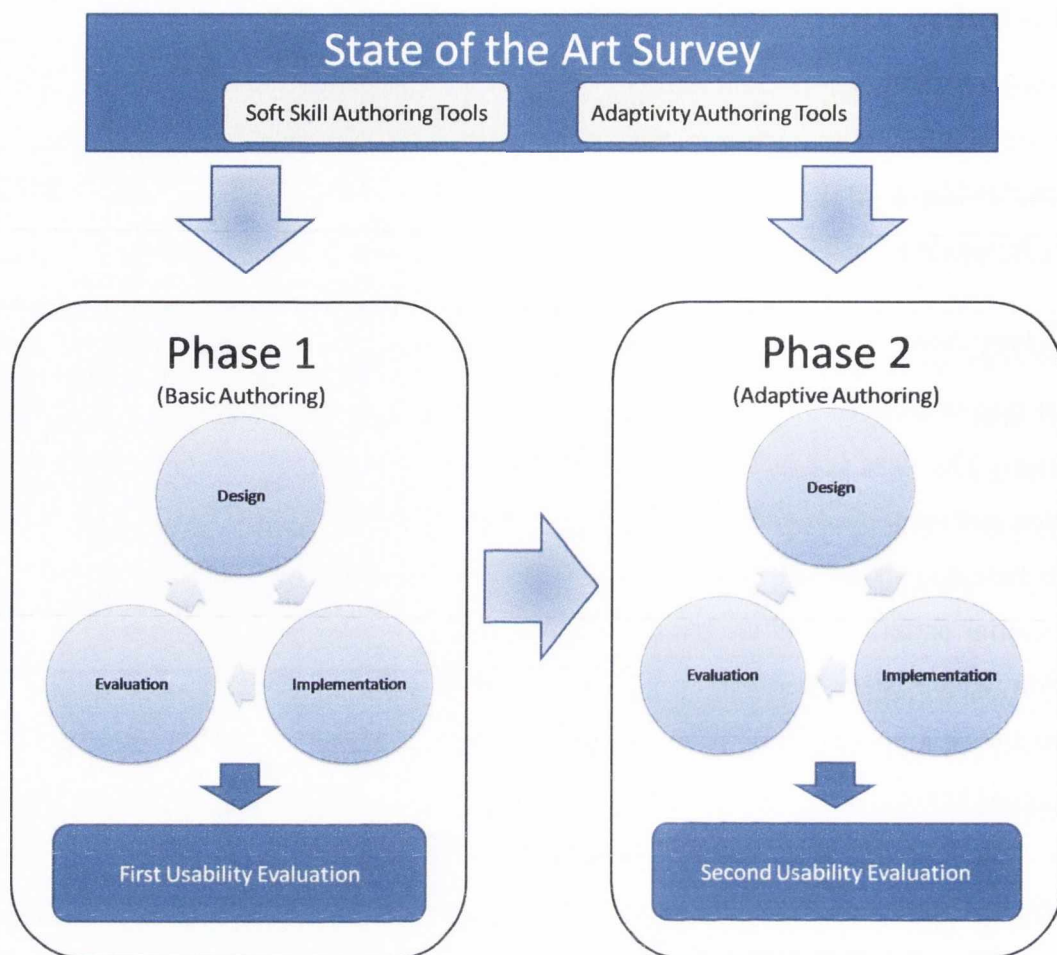


Figure 1.1: Technical Approach

Throughout the development process, in both phase one and phase two, an action research approach (McNiff and Whitehead 2011; Kember 2000) was employed. Action based research incorporates many cycles of plan/reflection, action and evaluation (Chou 2010; O'Shea and Reddy 2007; Hendricks 2006) which is realised here in terms of design, implementation and evaluation, as presented in both phases in Figure 1.1. Action research also requires working closely with end users; soft skill simulation authors within this research. During the course of this research user centric evaluations were completed in several different areas including psychiatry, customer care, exam stress and healthcare. This allowed the composition tool to be evaluated in an authentic real world environment.

The following sub-sections briefly describe the different components of the technical approach. These include educational underpinnings, the state of the art survey, design and implementation, and finally the evaluation approach.

1.5.2 Educational Underpinnings

The first stage of this research was to investigate the educational underpinnings involved in the personalised simulations. This included a detailed investigation of the effectiveness of personalisation in education as well as soft skill simulations. While there has been a great deal of separate research in both of these areas there has been very little with the two combined. The findings of this investigation are described in chapter two; educational underpinnings of simulation and adaptivity based learning.

1.5.3 State of the Art Survey

The second stage was to closely examine the state of the art survey of authoring tools used to compose adaptive soft skill simulations. However, after an extensive search to identify a representative sample of authoring tools used to compose adaptive soft skill simulations, the author concluded that no such tools currently exist. Instead the survey was divided into two; the first survey examined a representative sample of soft skill simulation authoring tools; the second survey a representative sample of authoring tools used to compose adaptive eLearning courses. The composition tools selected for rigorous analysis were chosen as they demonstrated methodologies or features that were particularly innovative. The details of the state of the art survey can be found in chapter three.

1.5.4 Design and Implementation

Once the state of the art survey was complete the first phase of development was begun. The goal of the first phase was to build and evaluate an authoring tool used to compose basic (non-adaptive) soft skill simulations. This was accomplished by iterating through a series of design, implementation and evaluation cycles; as presented in Figure 1.1. The initial iteration of this development cycle incorporated methodologies and requirements that were extrapolated from the analysis of the state of the art survey.

Once the composition tool had reached a mature stage of development in phase one, a complete usability evaluation (Preece 1994) was conducted; further details of this evaluation approach can be found in the following sub-section. The usability evaluation was used to objectively examine if the prototype addressed each of the phase one requirements; i.e. it allowed basic soft skill simulations to be authored. With phase one of development complete, phase two could begin. The main objective of phase two was to incorporate features and methodologies into the existing tool so as to allow adaptive soft skill simulations to be

composed. Phase two was also used to incorporate additional functionality and to re-evaluate aspects of the prototype initially implemented in phase one of development. Phase two followed a similar pattern to phase one, as presented in Figure 1.1. The major difference between phase one and phase two was that the initial iteration through the design, implementation and evaluation cycle needed to consider the prototype produced in phase one along with the requirements extrapolated from the analysis of the state of the art survey. Similarly to phase one, once the composition tool had reached a mature level in phase two, it was comprehensively evaluated with the second usability evaluation. A more complete description of the evaluation approach is outlined below. Design and implementation are described in detail in chapters four and five respectively.

1.5.5 Evaluation Approach

Throughout development, the composition tool was rigorously evaluated. Evaluations were incorporated as part of an action based research approach with results of each evaluation feeding back into further iterations of design and implementation. Evaluations were case study based and user centric, employing users (authors) from authentic real world environments.

During the course of development several different types of evaluation were employed. Suitable evaluation types were selected based on the different stages of development. Evaluation types included predictive and interpretive (Preece 1994), as well as the aforementioned usability evaluations. Each of these evaluation types are briefly outlined below:

- **Predictive Evaluations:** Predictive evaluations are typically used at an early stage of development or to examine new features being introduced to an existing system. Specifically, cognitive walkthroughs (Rieman and Franzke 1995), a type of predictive evaluation, were used in this research to evaluate the composition tool; within cognitive walkthroughs, a mockup of an interface (storyboard) is created and a verbal step-by-step evaluation of the feature(s) is executed. The phase one cognitive walkthrough incorporated the requirements dictated by the state of the art survey. The cognitive walkthrough in phase two incorporated a facsimile of the phase one prototype with additional features incorporated (including adaptivity). These new features were also extrapolated from an analysis of the state of the art survey. The

results of both these evaluations were incorporated into the following development cycles.

- **Interpretive Evaluations:** Interpretive evaluations are used at intermediate stages of development. They can be prepared in a relatively short period allowing rapid development of prototypes. Specifically, contextual interviews (Holtzblatt and Jones 1993), semi-structured interviews and questionnaires were employed within this research.
- **Usability Evaluations:** Usability evaluations are a much more detailed examination of a system than the evaluations previously outlined. They do, however, take considerable time to prepare and execute. During this research usability evaluations were completed at the end of both phases of development to thoroughly investigate the composition tool prototypes had achieved their objectives and realised the systems requirements.

After each evaluation, time was taken by the author to reflect on the results and feedback of the evaluation. Reflection allowed the next iteration of design to be informatively considered. The reflective points during the two phases were minor and less explicit than the two reflection points after each phase of development which were more comprehensive and explicit. A complete description of the evaluation approach, different types of evaluation, evaluation results and reflections can be found in chapter six.

1.6 Summary

This chapter has introduced and outlined the research contained within this thesis. It has described the research motivation and associated research question. It has also presented the objectives and contribution to the state of the art. Finally, it has outlined the technical approach employed during the course of this research. The following chapter described in detail the educational underpinnings associated with this research.

2 Educational Underpinnings of Simulation and Adaptivity Based Learning

The educational underpinnings involved in simulation and adaptivity based learning are varied and complex. In order to fully describe the different aspects involved in their pedagogical principals this chapter is divided into four primary sections. Section 2.1, describes general educational theory and particularly focuses on constructivism, situated learning and cognitivism. The next section, 2.2, outlines simulation based learning and identifies their different categories, key elements, impacts and types of educational theory with which they are associated. These same areas are identified and discussed with respect to adaptivity based learning in the section 2.3. The final section, 2.4, examines adaptive educational simulations, the blending of simulation and adaptivity based learning.

2.1 Educational Theory

This section describes several categories of learning theory; each is described in general terms and is not specified or applied to a particular area of pedagogy. Later in this chapter these theories are re-examined with respect to simulation based learning and adaptivity based learning in sections 2.2 and 2.3 respectively.

It is possible to identify three main categories of learning theories: behaviourism, cognitivism and constructivism (Schuman 1996). Each of these theories takes a different view of how learning occurs and as such, attempt to assist in understanding the complex processes involved. Learning theories provide an explanation of learning which allows behaviour to be explained and predicted (Dorinn, Demmin et al. 1990). Learning theories also provide a conceptual framework for interpretation that can both complement and contrast each other depending on the context and environment in which they occur.

It is beyond the scope of this research to present a complete account of all learning theories. This section details the theories, and sub-theories, that are most relevant to both simulation and personalised based learning. The following sections briefly introduce constructivism, situated learning and cognitivism. In subsequent sections in this chapter these theories are discussed in relation to computer based educational simulations and adaptive hypermedia systems.

2.1.1 Constructivism

There are two main conceptions of learning which rival each other, the objectivist approach and the constructivist approach (Stewart 2001). The objectivist approach includes behavioural and cognitive theories (Mergel 1998) and has, until relatively recently, dominated the field of education (Varsidais 2000). The objectivist theories are a more traditional approach to learning where knowledge is transferred from educators, directly or through technologies, and acquired by learners (Jonassen 1999). Learning in objectivist theories are centred about directly changing and observing the behaviour of the learner (Mergel 1998).

The constructivist approach to learning argues that learners generate knowledge and meaning from their own experiences (Yakimovicz and Murphy 1995; Piaget 1967). Knowledge is *“individually constructed and socially constructed by learners based on their interpretations of experiences in the world”* (Jonassen 1999), instruction should consist of experiences that facilitate this approach. Constructivism views learning as an active process in which learning occurs based on experience (Merrill 1991). This is very much opposed to objectivist theories which view learning to be a much more passive experience. Within constructivist learning a learners interaction with their environment obviously becomes one of the most important elements of the learning experience (Vygotsky 1978; Dewey 1938).

2.1.2 Situated Learning

Situated learning is a derivative of constructivist theory (Land and Hannafin 2000); its fundamental concept is that all learning should occur in an appropriate context which increases the educational effectiveness of educational experience (Alessi and Trollip 2001), learning should take place in the same context in which it is to be applied (Lave and Wenger 1991; Suchman 1987). Knowledge and the relevance of that knowledge to a learner can become depreciated if learning is removed from its context (Duffy and Cunningham 1996). Many studies have shown that learning situated in real-world contexts have delivered positive results in several different aspects of learning, such as motivation, engagement and educational effectiveness (Anderson, Reder et al. 1996; Duffy and Cunningham 1996). Situated learning does not employ a linear view of instruction such as that which is incorporated in more traditional approaches like constructivism. Learning instead occurs naturally as a consequence of the learner recognising the practicality of the knowledge that is

being conveyed to them (Hung and Chen 2002). Situated learning involves interactivity and a practice-based approach to learning which acts as a bridge between theory and execution (Lunce 2006).

Process of thought, or cognition, is a pivotal part of situated learning (Lunce 2006). This is the process of learning to use an instrument or artefact in a real-life situation to accomplish a real-world objective (Henning 1998). As a learner engages in situated learning activities, they improve their cognition and meta-cognition (Land and Hannafin 2000). The knowledge a learner constructs can be more easily transferred into the real-world as the learner can associate it with cues that are already familiar (Moore, Burtan et al. 1996).

2.1.3 Cognitivism

The objectivist approach to learning includes both behavioural and cognitive theories. Cognitive theory is similar to behaviourism in that both theories consider learning has occurred if it can be identified as a change in behaviour of the learner (Mergel 1998). Furthermore, both theories suppose that learning occurs when it is transferred from an educator to a learner. The major difference between the two theories is how they consider the learning process. Behaviourism theory does not concern itself with how learning occurs; it simply accepts that learning takes place through behavioural patterns being repeated (Mergel 1998). This theory is very much opposed to cognitivism which in contrast strongly focuses on *how* a learner builds their knowledge and understanding (Ashcraft 2006). Cognitivism views the mental processes that occur during learning to be the main focus of study. It attempts to model these processes in such a way that they can be identified and understood.

There are two key assumptions contained within cognitivism; firstly that a learner's memory system is an active organizer of information; secondly that a learners prior knowledge plays an important role in learning (Ashcraft 2006). Cognitivism views learning as an active process in which learners are involved in the learning process.

2.2 Simulation Based Learning

Simulations have been widely used and utilised in areas as diverse as business and industry (Markulis and Strang 1985) to science and technology (Casti 1996). There are many interpretations of what a simulation is and while there is no de facto definition they are generally accepted to be a model of a real world environment, normally with the facility for a

user to interact with that environment (Dalgarno 2001). More precise definitions are dependent on what a simulation is modelling and the context upon which it occurs. For example, mathematical simulations might be defined as logical models which are not amenable to conventional analytic or numeric solution and are used to provide insight (McHaney 1991; Neelamkavil 1987). In the domain of physics a simulation might be defined as models of the real world that are observed and used to understand the behaviour of a model and predict its future behaviour. An Astronomy simulation would be defined with Newtonian rules and a Zoological definition would employ physiological references. Within the domain of Computer Science simulations can be defined as a program that contains a model of a system or a process (De Jong and Van Joolingen 1998) that attempt to mimic a real or imaginary environment or system (Langone, Clees et al. 2005).

The purpose of simulations in education is generally accepted to allow a learner to engage in problem solving, hypothesis testing, experiential learning, schema construction, and development of mental models (Duffy and Cunningham 1996; Winn and Synder 1996). While the goal of an educational simulation is apparent there is much confusion and disagreement about how an educational simulation should be defined (Alessi and Trollip 2001).

There are several approaches employed in defining educational simulations. Generally they each highlight one aspect of the educational simulation more so than other definitions. For example, one such definition highlights the active nature of the learner within the simulation stating that educational simulations are interactive learning environments in which a model simulates characteristics of a system that are dependent on actions made by the learner (Huisman and Vries 1991). Another definition refines this definition so that it should also include some form of guidance (Kirschner and Huisman 1998) which focuses on the feedback during the simulation. A further definition highlights learner observation and manipulation of time in the simulation by stating that an educational simulation is designed to teach someone about a system by observing the result of actions or decisions through feedback generated by the simulation in real, accelerated or slowed time (Langone, Clees et al. 2005).

Educational simulations can also be defined with a pedagogical view point; they are designed to both enhance higher-order problem-solving skills and to teach content. Simulations should

allow learners to explore and manipulate variables and then obtain results from the various manipulations. Those results provide feedback to their thinking and learning processes (Akpan 2007). Reigeluth and Schwartz define an educational simulation by describing its components. It is described in terms of its three major design aspects: the scenario, the underlying model, and the instructional overlay (Reigeluth and Schwartz 1989). De Jong and van Joolingen consider the model upon which the simulations operate to be paramount. They define an educational simulation stating that the learner should manipulate the characteristics of the model underlying the simulation. That the learners' basic actions are changing values of input variables and they learn by observing the resulting changes in values of output variables (De Jong and Van Joolingen 1998).

These definitions are however rather restrictive when considering the many different types of educational simulations that exist. While some of the important elements of an educational simulation are captured in these definitions they do not provide an all encompassing description. Furthermore, some of these definitions would not include emerging technologies that should be classed as educational simulations such as, virtual reality (Psocka 1996), case-based scenarios (Jarz, Kainz et al. 1997) and certain types of microworlds (Rieber 1996; Brehmer and Dorner 1993).

Alessi and Trollip (2001) offer a much more complete definition of educational simulations (Dempsey 2009; Lunce 2006). They state that *"an educational simulation can be defined as a model of some phenomenon or activity that users learn about through interaction with the simulation"*. This is a much more encompassing view of educational simulations. While it still describes the more traditional idea of a simulation it also allows for more contemporary applications that should be classed as educational simulations. Furthermore, Alessi and Trollips definition excludes systems that might fit certain criteria of educational simulations but should not be classed as such; these include *"movies, animations, and many types of games"*.

2.2.1 Categories of Educational Simulations

Categorisation of educational simulations is another area of contention with many different approaches and interpretations. De Jong and van Joolingen (1998) divide simulations based on their underlying model. They state that educational simulations can be divided into two categories, simulations containing a conceptual model and those that are based on an

operational model. Conceptual models operate on principals, concepts and facts related to the system that is being simulated. Operational models use cognitive and non-cognitive operations which are applied to create a simulated system.

Reigeluth and Schwartz (1989) build on some earlier work by Alessi and Trollip (1985) to categorise educational simulations as procedural, process and casual. Procedural simulations teach a learner to perform a sequence of steps or decisions; such as adding fractions or flying an airplane. Process simulations are used to teach *naturally* occurring phenomena composed of a specific sequence of events; such as the sequential steps associated with photosynthesis. Casual simulations are based on cause-effect relationships between two or more changes; such as the law of supply and demand.

Aldrich (2005) identifies four categories or genres of educational simulations; branching stories, interactive spreadsheets, game-based models and virtual labs/products. Branching story simulations allow the learner to make multiple-choice decisions along on-going sequence of events in a formulated narrative. Interactive spreadsheet simulations are based on abstract business school issues such as supply chain management, product lifecycle and accounting. Game-based simulations include elements of “*fun*”; they often employ elements of fantasy in order to engage the learner. Virtual lab or product simulations focus on simulating equipment, they either examine learning about the system being simulated or the procedures incorporated in the environment.

There are also many other approaches employed in categorising educational simulations (Gibbons, Fairweather et al. 1997; Towne 1995; Goodyear, Njoo et al. 1991). However, Alessi and Trollip (Alessi and Trollip 2001) again offer the most comprehensive high level categorisation of educational simulations (Lunce 2006). They state that there are four categories of educational simulations: physical; iterative; procedural; and situational. The first two categories of simulations, physical and iterative, are “*teach about something*” simulations. Physical simulations allow the learner to continuously manipulate objects in order to observe results, teaching the learner about these objects. A typical example of this category of simulation is Future Lab: Circuits for Physical Science (SuperKids 2010) which teaches children how to build and understand electrical circuits in a virtual environment. Iterative simulations teach by allowing the learner to initialise the simulation with parameters

and observe the resultant output. Catlab (Catlab 2011), which teaches introductory genetics, is a typical example of an iterative simulation.

The second two categories of simulations, procedural and situational, are “*teach how to do something*” simulations. The focus of these categories is the way in which the learner interacts with the simulated objects. Procedural simulations teach a learner processes such as those involved in interacting with machinery or software. It is the way that learner interacts with the objects that is important. Microsoft Flight (Microsoft Flight 2012) which teaches a learner the processes involved in flying an airplane is a well know example of a procedural simulation. The final category of educational simulations is situational. Situational simulations, or soft skill simulations as they are more commonly known within industry, are used to teach skills based on interpersonal relationships. They are set in a social situation where the learner takes on a particular role and is taught through interaction with a simulated person or people. Examples of soft skill simulations include SkillSoft Simulations (SkillSoft 2012) and Raptivity (Raptivity 2012), both of which can be used to teach management and leadership skills. An additional more detailed description of soft skill simulations can be found in Appendix I – Soft Skill Simulations.

2.2.2 Identification of Key Properties in Educational Simulations

An analysis of educational simulations, and associated literature, was conducted to identify their key properties. The six key properties identified by the author are present in most educational simulations. This section describes each of the six key properties and justifies their inclusion and significance. The properties include the activity of the learner, motivation inferred by the simulations, simulation fidelity, engagement of the learner, feedback and the underlying model of the simulation.

Learner Activity

Simulations require the learner to be active during their learning experience. The learner must participate (Lunce 2006) rather than sit by passively as with more traditional approaches to learning. A learner may interact with a simulation in a variety of ways; they are not necessarily restricted to using a keyboard and mouse as methods of input/modality. A simulation can employ realistic consoles, a joystick or microphone (Alessi and Trollip 2001). As educational simulations become more sophisticated and these devices become more

common, they will inevitably replace the keyboard and mouse as more realistic input devices. Microphones are already widely available on computer consoles and tablet computers (Sharples and Beale 2003; Kay and Goldberg 1977), such as the iPad (Apple 2011), allow more realistic interaction with virtual console displays.

The types of actions the learner can employ within the simulation can be categorised. The learner can make a choice, manipulate an object, react to an event or collect information (Alessi and Trollip 2001). Generally the actions are only approximations of the real world as they are usually not the point of the simulation. The choice parameters that can be manipulated are however important within the simulation. By limiting the parameters that the learner can change and emphasising particular actions and controls a simulation guides learner thinking (Adams, Reid et al. 2008).

Simulations require a learner to interact with their learning environment which is both appealing and effective (Alessi and Trollip 2001). The type of actions employed in a simulation can have a varying degree of realism. It is not necessarily always preferable for the actions used in a simulation to be as accurate as possible. The actions employed will be dependent on the type of simulation and the instructional strategy. The simulation might be based on virtual reality or a microworld which requires the learner to be taught through investigation and discovery. This type of simulation would require realistic interactivity which allows the learner to explore the simulated world. Another type of simulation might focus on a particular software interface or piece of machinery. The learner would not be required to move about in such a simulation and it would need to only replicate the real world actions. This could mean initially restricting some of the learners' actions or offering them multiple choice options throughout the simulation. This approach allows the interactivity to be directed and employs a more guided pedagogical strategy.

Motivation Inferred by Simulations

Although a certain amount of motivation needs to be designed into the simulation, learners are generally interested in a simulation as they are participating in the scenario rather than reading about events (Lepper and Chabay 1985). Motivation is commonly associated with the overall objective of the simulation. Providing clear goals to the learner is therefore very important for the motivation (Adams, Reid et al. 2008). The theory of motivation (Keller and

Suzuki 1988) indicates that the relevance of the lesson to the learner also needs to be clear. The learner is generally provided with objectives in the introductory section where a great deal of emphases is put on explaining the purpose of the simulation. In order to further encourage motivation it is also often useful to clarify the educational purpose of the simulation and to give some idea of the activity to come (Alessi and Trollip 2001).

The role of motivation has been recognised as being very important within the area of education (Pintrich 2000). Learners are more motivated by active participation in a situation rather than passive observation (Lepper and Chabay 1985). They feel that simulations are more relevant to what they are learning than the use of classes or books (Alessi and Trollip 2001). Material that is not taught in the correct context, such as that encouraged by simulations, can adversely affect a learners motivation (Henning 1998).

While providing the learner with clear objectives is the most concise way to ensure motivation there are also several other methods which are far more subtle. For instance, the type and frequency of a simulations interactivity can effect a learners motivation. Learners need to have a sense of control when engaging with a simulation (Malone and Lepper 1987). If a simulation does not allow a learner to constantly and meaningfully interact with their environment they can quickly lose interest. Another method of insuring a learner stays motivated is by adjusting the educational strategy. This can be accomplished by employing gaming techniques (Lepper and Chabay 1985) or an element of fantasy (Malone and Lepper 1987) into the simulation.

Simulation Fidelity

Fidelity refers to the accuracy with which the simulation models a real world system or phenomenon (Lunce 2006; Alessi and Trollip 2001), it is the level of realism that a simulation presents the learner (Feinstein and Cannon 2002). A simulation with a high level of fidelity is considered to be realistic while a simulation with low level is considered to be unrealistic. Fidelity is not a single factor of an entire simulation but one that applies to many different aspects of a simulation, including graphical representation, interactivity and system reaction (Alessi and Trollip 2001). A simulations level of fidelity is closely related to its educational effectiveness.

There have been many studies of simulations completed (Blaiwes and Regan 1986; Martin and Waag 1978; Miller 1978) which examine the relationship between fidelity and how it affects a simulations educational effectiveness. These studies have found that a high level of fidelity does not necessarily mean that a simulation will be educationally more effective (Feinstein and Cannon 2002). It has been discovered that simulations with lower fidelity can in fact be more educationally effective (Alessi 1988; Dwyer 1974; Miller 1974). Some studies go as far as to state that high fidelity can actually reduce the educational effectiveness of simulations as the learner can become overwhelmed (Martin and Waag 1978).

Alessi and Trollip (Alessi and Trollip 2001) believe that the relationship between fidelity and educational effectiveness is actually far more complex than simply stating that high or low levels of fidelity are most effective. The fidelity of a simulation and its educational effectiveness is in fact dependent on the learners' previous experience. A novice learner who has not had much experience is more effectively educated with a simulation that is lower in fidelity, as the learner progresses through the simulation they would require a simulation with a higher fidelity. The theory hypothesises that in the initial stages of learning, a learner can be confused and overwhelmed with a simulation that is very realistic. Whereas a more experienced learner requires the challenge presented with a realistic simulation and would be educated more effectively.

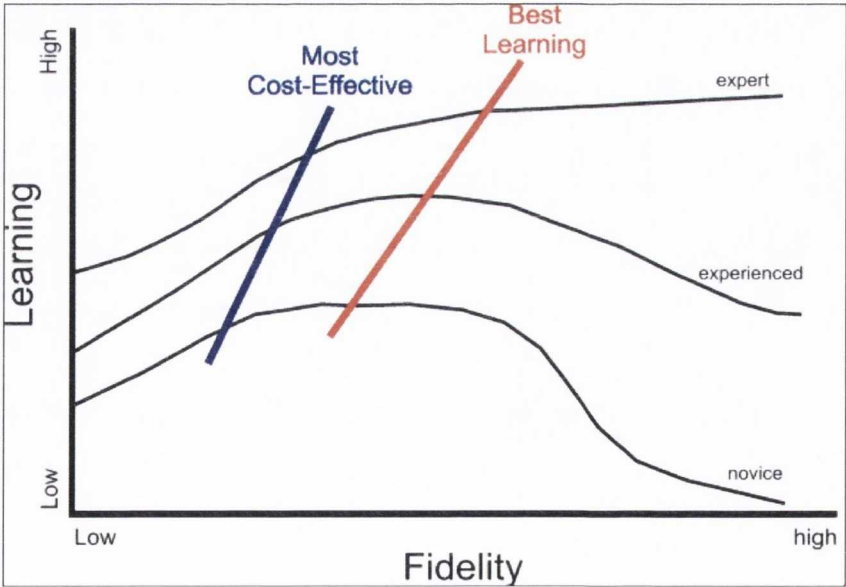


Figure 2.1: Hypothesised Relationship of Fidelity and Learning (Dagger, Rogers, et al. 2007)

Figure 2.1 (Dagger, Rogers, et al. 2007; Alessi 1988) illustrates the hypothesised relationship between fidelity and learning. The line of best learning, near the maximum point of each curve, illustrates the point of best learning for the different levels that learners are at: novice, experienced and expert. The most cost effective line illustrates the point for each learner type where there is only a small amount of gain in educational effectiveness compared to a great increase in expenses.

Engaging the Learner

Learners are engaged in simulations through thoughtful exploration by posing questions and seeking answers (Adams, Reid et al. 2008). Curiosity is created by balancing the challenge of the simulation with an optimal level of informational complexity (Malone 1981). A simulation also needs to obtain a certain level of believability for the learner to find it engaging (Adams, Reid et al. 2008). This means that the simulations need to be interesting so learners enjoy using them, if they look boring or intimidating learners are not drawn to them (Adams, Reid et al. 2008). As found with the concept of motivation, providing clear objectives to the learner also makes the simulation more engaging.

It has been well documented in many studies that learner engagement is very important to the educational effectiveness of any learning environment (Herrington, Oliver et al. 2003). When a learner is engaged in a learning process their levels of retention and the effectiveness of learning are increased (Kearsley and Shneiderman 1998). Learners should be engaged through interactions with others and with tasks that are meaningful and worthwhile (Kearsley and Shneiderman 1998).

Most state-of-the-art technologies do not ensure that learners are engaged in their learning experience (Lim 2004). In order to support meaningful learning the environment which engages a learner needs to be interactive, provide purpose (Jonassen, Dyer et al. 1997) and context (Henning 1998). Educational simulations, by their very nature, are highly interactive (Alessi and Trollip 2001). They also provide a particular goal or purpose for the learner which must be designed to be attainable to encourage self-esteem and engagement (Malone 1981). Simulations are also set in a model of the real-world which not only engages learners but allows knowledge to be created in the correct context (Henning 1998).

Feedback of the Simulation

Educational simulations commonly employ feedback and system reaction (Alessi and Trollip 2001). There are two dimensions to be considered when examining delivery of feedback, the first is the *method* of delivery and second the *timing* of delivery. The method of delivery of feedback can be categorised as either natural or artificial. Natural feedback means that the feedback or system reaction is true to that of the real world. Artificial feedback consists of helpful directions or advice that would not normally appear in the real world. For example, a flight simulator natural feedback might be an alarm on the console that the airplane is low on fuel. Artificial feedback might also include information on where the closest airport is and direct a user on an optimal route.

The second dimension to be considered with feedback is timing which can be categorised as immediate (delivered in real-time) or delayed (delivered at a later stage of the simulation or once the simulation has ended). Taking the flight simulator again as an example, the aeroplane might be coming into land without its undercarriage down. While immediate feedback would alarm the learner as they are landing, delayed feedback would not occur until after the simulation has ended. Delayed feedback is often used to explain to a learner where they have made errors; in the case of the example crashing because the undercarriage was not down.

The different approaches employed in delivering feedback to the learner each have an effect on the learners' educational experience. Artificial feedback can be very helpful to the learner during the simulation. It is usually obvious and understandable but it is not the kind of feedback the learner would receive in the real world so does affect the fidelity of the simulation. Natural feedback creates a more genuine facsimile of the real world which creates a greater challenge and a much more interesting learning experience. Immediate feedback can prevent the learner from making errors and increase learning efficiency but can distract the learner from their activity in the simulation. Delayed feedback allows the learner to be corrected and informed about their performance but does not break the continuity of the simulation. Delayed feedback can be considered similar to the notion of reflective learning. Reflective learning is the process of internally examining and exploring an issue of concern (Atkins and Murphy 1993; Boyd and Fales 1983). It has been viewed that reflective learning "*is central to understanding the experiential learning process*" (Boyd and Fales 1983).

Different types of feedback should be incorporated at different stages of an educational simulation. A good simulation should begin with feedback that is helpful, corrective and immediate. As a learner progresses through the simulation the feedback should become less artificial and more natural. This approach consolidates the relationship between fidelity and learning as feedback greatly affects fidelity. Immediate artificial feedback creates a low fidelity simulation while natural or delayed feedback creates a high fidelity simulation. The final approach is to employ no feedback in the simulation; this would be suited to a simulation that is used in testing the learner and would probably be the final application of a simulation before the learner engages with the real world.

Underlying Model of a Simulation

The underlying model is the representation of the system or phenomenon that is being simulated (Lunce 2006; Reigeluth and Schwartz 1989). The model is formed with a computer program which depicts the physical entity; the procedure; the situation the learner is a part of; or the process that the program mimics (Lunce 2006). There are three types of underlying model: continuous; discrete; and logical (Lunce 2006; Alessi and Trollip 2001). Continuous models represent a system with an infinite number of states and are implemented using calculus. Discrete models represent systems with quantitatively discrete states and are typically based on probability and statistics. Logical models are represented by a set of *if-then* rules in a program. These are the most common type of underlying model in educational simulations. The models represent systems such as those used in the operation of machinery, many social interactions and decision making in running a business (Alessi and Trollip 2001).

The underlying model upon which a simulation operates is typically not exposed to a learner; the learner will interact with the simulation but will not be presented with the underlying model directly. Depending on the type of underlying model a simulation is based on, the learner may not even need to understand or conceptualise it. For instance, logical model simulations are based on *if-then* rules and while the learner needs to understand the consequences of their actions within the simulation they are not required to learn this set of rules. Logical models can also be very abstract compared to their realisation in the simulation and their exposure would only confuse the learner. Discrete model simulations allow the learner to be taught about patterns and distributions which are based on alternating discrete

numbers that are generated by algorithms in the underlying model. These algorithms can be very intricate and complex but the learner does not need to understand how the models are implemented, they need only concern themselves with the results in the simulation. Continuous models usually support simulations in the area of physical and social sciences. Typically, the goal of these simulations is to illustrate the application of formulae. With this type of underlying model it is important for the learner to have a clear understanding of the formulae upon which the model is based.

2.2.3 Consequences of Key Properties within Soft Skill Simulations

There are several means by which a learner can be **active** within a soft skill simulation. Learners can interact with a simulated person through multiple choice (mouse or touch screen), text input (keyboard) or voice input (microphone). There are advantages and disadvantages to these different approaches. Voice input allows the learner free and easy input. However, the underlying models associated with such a system are typically very difficult to compose as they require a deep understanding of natural language and the construction of complex rules. Furthermore, voice recognition technology is often inaccurate. Text input can overcome this inaccuracy but is slow and it also typically involves difficult to author underlying models. A multiple choice input is quick and easy to use but limits a learner's options. Multiple choice does however allow a learner to experience correct paths and decisions; with voice or text input a learner may never select the correct interaction with a simulated person.

The activity of a learner in a soft skill simulation is typically very high. Soft skill simulations usually follow turn taking that occurs naturally within a dialogue; a learner will input their dialogue selection, the simulated person responds and the learner is again required to interact. Such a high level of participation, with a great deal of control incorporated, results in a learner being immersed and **motivated** throughout a simulation. Motivation is also designed into a soft skill simulation by providing a learner with an overall objective at the beginning of a soft skill simulation in an introductory section.

There are several ways which a soft skill simulation can be realised and delivered; directly affecting their **fidelity**. The responses of a simulated person can be presented and implemented with any combination of text, images, audio, virtual reality avatars or video. Simple text and images are however not particularly engaging. Computer synthesised audio

can work well in some situations (e.g. simulated telephone dialogues) but it is typically associated with very complex underlying models. Virtual reality avatars similarly require complex underlying models and also require a great deal of resources to be graphically designed and implemented. Furthermore, virtual reality avatars presently lack the sophistication to capture the subtleties and nuances of a person that are required to create a really believable simulated person. While the level of fidelity within a simulation is related to its effectiveness of as a learning tool, it should also be noted that if a simulation is not believable it is not engaging (Malone 1981).

Soft skill simulations **engage** a learner by initially setting clear objectives which incorporates a high level of activity while also delivering a believable level of fidelity. Soft skill simulation must also be challenging for a learner to keep them interested but not so difficult that learners become frustrated. For example, with multiple choice simulations the correct options cannot be too obvious. Similarly, voice and text input soft skill simulations must be sophisticated enough to comprehend and respond to a learners dialogue.

Soft skill simulations can implement **feedback** naturally or artificially, deliver feedback immediately or with a delay. However, feedback delivered immediately and artificially does affect the fidelity of an educational simulation. This is particularly relevant to soft skill simulations as immediate feedback, such as reflective interventions or direct coaching, can potentially break the flow of a dialogue. While the series of turn taking that occurs in a dialogue allow for such immediate feedback, if the interventions occur too often, require a great deal of time to consider or are cognitively exhausting they can adversely affect a simulation.

There are a number of ways the **underlying model** of a soft skill simulation can be designed and implemented. Discrete models are employed the most as they are relatively easy to create. They are typically realised as branching graphs or trees. A learner is allowed to select from multiple choice options based on dialogue components that would most probably be selected in real life. This type of underlying model is best suited to very refined dialogues that are taking place with a specific goal, outcome or underlying process involved. Logical models are typically employed with soft skill simulations that allow voice or text input.

Logical models are much more difficult to create as the rules are difficult to define, some employ complex AI while others are dependent on very large banks of dialogue components.

2.2.4 Educational Theory of Computer Based Educational Simulations

The educational benefits of simulations have been well documented in literature (Forinash and Wisman 2001). They can incorporate a wide range of strategies such as role playing, case-based scenarios, microworlds and virtual reality (Alessi and Trollip 2001). Educational simulations are more effective than other modalities at communicating complex information and skills. When correctly incorporated into a learning curriculum simulations can be a very powerful learning aid. Simulations have been shown to provide instruction which results in an improved performance in a real world setting (Leemkuil, De Jong et al. 2003).

There are many factors that feed into the effectiveness of educational simulations. For instance, the learner actively participates in their learning experience and they can be guided to achieve specific learning goals (Gibbons, Fairweather et al. 1997). The learner is also highly motivated which is very important to the educational effectiveness of the simulation and ensures its completion (Pintrich and Schunk 1996). By incorporating fidelity that is suited to the experience of the learner a simulation can be more effective than immediately placing the learner in the real world (Alessi 1988). By engaging a learner a simulation creates an environment where the learner is actively working to understand the content of the course (Adams, Reid et al. 2008). Feedback provided to the learner during a simulation can also be very beneficial to their learning experience (Granland, Bergland et al. 2000).

While there are several factors that feed into the educational effectiveness of simulations and many different learning strategies can be employed in an educational simulation, they are all based on two underlying principals of education, constructivism and situated learning. Earlier in this chapter, section 2.1, both of these theories were described in general terms; the following two sections discuss these theories as they relate to educational simulations.

Constructivism

The constructivist approach to learning believes that learning is formed by the learner through interaction and practice. A learner's activity with their environment is integral to their learning process. Simulations are an ideal method of implementing the constructivist approach to learning. They are highly interactive, engaging and self motivating. In the past

there did not exist technologies which easily allowed any other approach to teaching other than objectivist as learners were taught by directly transferring knowledge. Providing the facilities and infrastructure to learners to allow them to learn through interaction was simply not viable. With the development of new technologies, such as virtual reality, computer games and in particular simulations, there now exists the means to teach learners with a constructivist approach to learning (Dede 1996; Eastmond and Ziegahn 1996). The role of computers in technology enhanced learning environments switches from a transmitter of knowledge (objectivist approach) to a facilitator of interaction (constructivist approach) (Varsidais 2000).

Situated Learning

Situated learning theorises state that, for learning to be effective, it should be situated in a realistic setting (Merrill 1991). Educational simulations provide both the context and interactivity that are required by situational learning (Hung and Chen 2002). Although they can be simplifications of the real world (Alessi and Trollip 2001) they remain contextually correct with a high level of interactivity which provides effective education. Simulations facilitate situated learning by providing contextually rich learning environments (Gruender and Tobin 1991) which allow a learner to practice real-world skills while focusing on the essential elements of a real problem or system analysis (Heinich, Molenda et al. 1999).

Appropriation and Implementation of Educational Simulations

While simulations are educationally very effective they are not necessarily the most optimal primary solution in every situation. Alessi and Trollip argue that educational simulations are more effective than other modalities *only* in particular domains (Alessi and Trollip 2001). They believe that certain domains or content that is very removed from the real-world and which is abstract or theoretical is typically more effectively taught through other modalities of instruction. There may for example be little benefit to including simulations in a theoretical mathematics course. However it could be argued that in some theoretical subjects' simulations may be effective as a secondary or supplementary mode of instruction which complements the primary method of teaching. In such situations simulations could have the ability to bring to life or animate abstract theory or allow learners to complete related activities.

Simulations are often incorporated in situations where it is desirable for the learner to immerse themselves in a problematic situation and experiment with different approaches within that environment (Heinich, Molenda et al. 1999). This approach to learning can take considerable time compared to other modalities of education and can also, if not correctly implemented, prove to be ineffective. For simulations to be educationally effective the learner must be provided with appropriate scaffolding, coaching (Duffy and Cunningham 1996) and feedback (Leemkuil, De Jong et al. 2003). Simulations that do not incorporate these elements as part of their learning experience can provide the learner with little gain (Min 2001; Heinich, Molenda et al. 1999).

Finally, it must be noted that for simulations to be effective they should be employed in conjunction with other forms of education. Learners should initially be taught theory with an objectivistic approach and then be allowed to construct their own interpretations and knowledge through the use of contextually rich interactive environments. While objectivism and constructivism are viewed to be opposing theories that are mutually exclusive and incompatible they are in fact complementary to one another (Jonassen 1999).

2.2.5 Application of Educational Theory to Soft Skill Simulations

Soft skill simulations take advantage of many technological advancements such as high fidelity graphics, easily accessible video, voice recognition and fast bandwidth. These advancements allow realistic dialogues to be conducted by learners with simulated people. Soft skill simulations allow learners to interact and practice dialogue so as to generate knowledge and meaning from their experience (constructivism). This learning experience is enhanced by the fact that soft skill simulations, by their very nature, create the appropriate context and setting of an immersive real world situation (situated learning).

Soft skill simulations are a very appropriate means of delivering effective educational experiences. However, they are in themselves not an exclusive component. They are instead, a stepping stone that connects traditional approaches to learning with real world situations. Soft skill simulations can be viewed as an integral component of a blended learning strategy that has in the past been limited by technological deficiencies and high production costs.

2.3 Adaptivity Based Learning

Personalisation and adaptivity have long been successfully incorporated in educational instruction (Sampson and Karagiannidis 2002; Conlan, Wade, et al. 2002; Corno and Snow 1986). This has in the past taken place in a “*one-on-one*” environment with one instructor teaching a single learner (Reiser 1987). Such a situation is accommodating to personalisation as the instructor can easily adapt to the needs of one learner. While “*one-on-one*” instruction is a very effective method of teaching (Graesser, Person et al. 1995), it is very impractical due to the cost and logistics of its implementation. The more traditional approach to education is to employ an instructor that teaches a large group of learners at once. This is a far more practical approach but results in less effective learning as the instructor cannot teach each learner individually, instead they must taught as a group.

Lectures and classrooms are traditional teaching methods which have acknowledged limitations (Davis and White 2005). The development of new technologies has introduced many positive improvements to learning. ELearning can be interactive, media-rich with just in time delivery available in user-centric environments (Sampson and Karagiannidis 2002). These technologies however have, until recent years, followed the traditional ‘*one size fits all*’ approach to learning (Brusilovsky 2003). What these technologies also allow for is the introduction of personalisation to large groups of learners without the high cost or impractical logistics. Personalised eLearning (Brusilovsky 2001; Brusilovsky 1998; Eklund and Brusilovsky 1998) allows systems to adapt to the individual needs of the learner and improve the educational effectiveness of each learner’s experience. Adaptive educational systems can take into consideration many aspects of a learner’s requirements such as their context (Milosavljevic 1997), culture (Sampson and Karagiannidis 2002), learning goals (Kaplan, Fenwick et al. 1993) or prior knowledge (Sheinberg 2001). Personalised eLearning has been shown in many studies to be highly effective at increasing students’ performance (Sampson and Karagiannidis 2002) and has been identified as the key element of next generation eLearning systems (Brusilovsky 2004).

A learner’s educational environment can be personalised in several ways, by adapting not only the content that is delivered to the learner but by also personalising activities, presentation and collaboration. To ensure that environments are adapted in a manner that is focused on education they must be built upon sound pedagogic strategies (Wade and Power

1998; Wade and Power 1998a; Dagger and Wade 2005). The educational effectiveness of a learning experience can be improved by personalisation by making the content more relevant (Kuo and Chen 2001); increasing the learners motivation and engagement (Cordova and Lepper 1996) and giving the learner a sense of empowerment (Bajraktarevic, Hall et al. 2003). These factors combine to create a tailored learning experience which seeks to maximize the potential of each learner (Dagger 2006).

2.3.1 Types of Adaptive Educational Systems

There are three types of systems that provide personalised education; intelligent tutoring systems (Cumming, Okamoto et al. 1999; Ottmann and Tomek 1998); adaptive hypermedia systems (Brusilovsky 1999; Tochtermann, Westbomke et al. 1999; Brusilovsky and De Bra 1998; Brusilovsky, Kobsa et al. 1998); and intelligent pedagogical agents (Conati and Zhao 2004; Kinshuk 2004; Johnson 2003; Reeves and Nass 1996). This section briefly describes each of these approaches to adaptive education.

Intelligent Tutoring Systems

The term Intelligent Tutoring Systems (ITS) describes a diverse number of computer based educational systems. An ITS can be laboratory instructors, problem solving mentors, coaches and consultants (Shute and Psotka 1995; Sleeman and Brown 1982). Their common feature is that, unlike traditional tutoring systems, they can adapt to the need of learner. Typically, an ITS consists of a student model, domain model, tutoring model and interaction model (Sampson and Karagiannidis 2002).

The student model stores data that is specific to each learner and is relevant to the learning environment. This could include prior knowledge, learning objectives or cultural context. The student model may also follow a learner's performance, progress and navigation as they advance through the learning environment. The domain model stores all the relevant content of the educational course. Unlike standard tutoring systems the domain model of an ITS is dynamic and also incorporates a set of rules which allows the system to '*reason*'. Tutor models describe the pedagogy(s) of the learning environment, it is the model that is used to examine each individual learner model and infer a particular learning style which is applied across the domain model to produce a tailored learning experience. The interaction model controls interactivity and presentation of the educational material. It manages the screen layouts and dialogue that occurs in the learning process. These descriptions offer only an

outline of the different components of an ITS. A great deal of research has been conducted on the development of different architectures and refinement of these models to offer innovative effective personalisation.

Adaptive Educational Web

A hypermedia network is comprised of components known as hyperdocuments. Each hyperdocument contains some form of multimedia content and acts as a node which is connected (linked) to other hyperdocuments (De Bra and Calvi 1998; Brusilovsky, Schwarz et al. 1996a). This structure allows a learner to “*jump*” from the multimedia contained in one hyperdocument to the multimedia contained in other hyperdocuments in a non-linear manner. The number of hyperdocuments and links within a hypermedia model can be very large. Navigating such a model can be difficult due to the large number and diversity of the navigational paths that are available (De Bra and Calvi 1998). Such is the complexity involved in traversing such large complex models that a learner may suffer from “*cognitive overload*” and become ‘*Lost in Hyperspace*’ (Conklin 1987). In order to alleviate the cognitive and spatial overload of users adaptivity was introduced to simplify the hypermedia model. Adaptive Hypermedia (AH) accomplishes this by adapting the structure and appearance of the hypermedia model for each learner based on their requirements (Conlan 2005; Ohene-Djan, Gorle, et al. 2003). These requirements are determined from information that is stored about each learner in their own unique user model.

Adaptive Educational Hypermedia (AEH) incorporates the architectures, methods and techniques that are employed in AH and applies it to an educational context in order to create personalised eLearning. Adaptivity is implemented by restructuring an educational hypermedia model with a learner’s user model, similarly to an ITS, and particular pedagogy strategies which are also contained within the system.

Intelligent Pedagogical Agents

Intelligent agents have many different functions with a wide variety of applications. They can, for example, be used as informational managers, information seekers, planning agents, user representatives, etc. Intelligent agents have numerous definitions and there is little agreement of what exactly an agent is. An intelligent agent’s ability varies significantly

depending on its purpose and its environment. In order to describe an agent its abilities are usually the centre of its characterisation. Intelligent agents are typically described using four major concepts which define their behaviour; autonomy; responsiveness; pro-activeness; and sociability.

Intelligent agents act by themselves or with other agents in order to achieve a particular goal. In the domain of education the goal of an intelligent agent includes: instructing a learner; correlating teaching material; and providing assistance in navigating complex learning environments. The agents can be personal assistants, user guides, alternative help systems, dynamic distributes system architectures, human-system mediators, and so forth (Aroyo and Kommers 1999). Modern day educational environments are becoming increasingly complex and dynamic; their infrastructures have become too difficult for a learner to manage independently. Intelligent agents allow material to be delivered more efficiently to learners but more importantly as effective personalised learning experiences. Intelligent pedagogical agents also deal with other issues such as security, providing services online and offline and mobility (hand held devices). Advances in technology have also provided a new type of intelligent computer tutors, animated pedagogical agents. Animated pedagogical agents adapt to their dynamic environments and provide opportunistic instruction. The learner interacts with an agent as an animated persona which allows the learner to engage in face-to-face dialogues which incorporate facial expressions and gestures. Animated pedagogical agents have proven to be very educationally effective as well as increasing the learners' motivation and attention. They attract learners and can demonstrate physical tasks.

2.3.2 Survey of Adaptive Educational Hypermedia

While there are several different approaches to realising adaptive learning this research focuses on Adaptive Hypermedia. Adaptive Hypermedia systems offer a crossover of the three approaches to personalisation previously described. AEH systems can be composed using ITS whereby the principals of adaptive hypermedia are encapsulated by the ITS in the authoring process. Furthermore, it is possible to develop AEH systems that make use of intelligent agents to deliver aspects of adaptivity within the learning environment. The intelligent agents could, for example, be used to gather content, organise content or provide adaptive selection.

Evolution of Adaptive Educational Hypermedia Systems

Although only a relatively recent development, AEH systems have already undergone several revolutions. Currently AEH systems are in their third generation of development. The first generation of adaptive educational hypermedia systems were developed in the early 1990's. These were monolithic systems that were typically not distributive in nature (Carver, Hill et al. 1999). They were rather simple in their approach to personalisation and employed stereotype-based models to determine adaptivity. The intelligence or rules used to apply adaptivity was also embedded into the content of the systems (Wade 2009). This resulted in systems that were application specific and courseware that was difficult to update and reuse. First generation systems were also limited by their method of composition. Authoring of early AEH systems was conducted by programmers working with instruction from domain experts. This approach was difficult, costly and time consuming. Examples of second generation AEH systems include AHA! v1.0 (De Bra and Calvi 1998), ELM-ART (Brusilovsky, Schwarz et al. 1996) and InterBook (Brusilovsky, Schwarz et al. 1996a).

The second generation of adaptive educational hypermedia systems improved on existing systems in several ways. The most widely implemented improvement to AEH systems was to employ a multi-model approach (Conlan 2005) to defining and developing the different components of the system. Typically, these were specified as the domain model, learner model and adaptation model (Surjono and Maltby 2003). The domain model was employed to describe the content of the educational course and how each component of the content relates to one another. The learner model details the learner's properties and preferences. The adaptation model describes the rules of adaptivity that can take place in the course. A personalised course is delivered with all three of these models working in cooperation. The second generation of AEH systems also eased the complexity of composition allowing development to be authored with less effort than that of programming the details of the course and adaptivity. Instead authoring tools were developed that allowed authors who possessed a background in adaptive hypermedia to more easily create adaptive courses. While these authoring tools did not allow non-technical domain experts to compose AEH systems they substantially reduced the cost and complexity of composing AEH systems. Examples of third generation AEH systems include APeLs (Conlan and Wade 2004) and AHAM (De Bra, Houben et al. 1999).

Third generation adaptive educational hypermedia systems have further improved on the innovations introduced in the previous generation. Third generation AEH systems are portal based and employ a federation of models. The type of adaptivity that they allow is far more multi-dimensional. Furthermore, third generation AEH systems allow adaptivity to occur at a much finer level of granularity (Carver, Hill et al. 1999). Both the dimensions and granularity of adaptivity are discussed in the following section. The third generation of adaptive educational hypermedia systems also adapt much more than simple hypertext as was predominantly the case with previous generations (Carver, Hill et al. 1999). The most innovative AEH systems adapt courseware that includes video, audio, tasks, etc. which creates a much more authentic hypermedia environment than previous generations. The authoring tools used to compose AEH systems have also greatly improved. Their use is no longer limited to experts in authoring adaptive systems but can now be accessed by non-technical domain experts. These tools allow authors who previously had no experience with developing adaptive eLearning courses to compose personalised experiences for their learners without the assistance of programmers or adaptivity experts. Examples of third generation AEH systems include Knowledge Sea (Brusilovsky and Rizzo 2002) and AHA! v4.0 (De Bra, Stash et al. 2007)

Dimensions of Personalisation

The dimensions of personalisation describe the aspects of the users' model that can be utilised in order to adapt the learning environment to create a personalised learning experience. The dimensions of personalisation have evolved in adaptive educational hypermedia systems as the systems themselves have advanced. Initially only a single characteristic was considered when adapting a course (Carver, Hill et al. 1999). For example, in first generation AEH systems prior knowledge was predominantly employed for personalisation (Surjono and Maltby 2003). Other learner characteristics employed by systems included cognitive styles (Mitchell, Chen et al. 2005) or a learners' navigational path (De Bra and Calvi 1998).

In second generation AEH systems the number of dimensions employed expanded but most systems still only processed a small number of learner characteristics (Carro 2002) and often in a not particularly sophisticated manner (Armitage and Bowerman 2005). Learners were typically grouped in a homogenous manner in order to describe their characteristics. This is

contrary to a realistic view of how learning occurs in the real world and might be adapted. Actual learners are not simple and one-dimensional but are much more complex, multi-dimensional and multi-faceted (Carver, Hill et al. 1999). The learning process in an AEH environment is complex and may be influenced by many different dimensions of a learner (Surjono and Maltby 2003). A learner's dimensions may be described with multiple user models or employ other sources of information about the learner that become available (Wade 2009). Third generation AEH systems have implemented such a multi-dimensional approach (Wade 2009; Surjono and Maltby 2003). These systems create a personalised learning experience that is influenced by many characteristics in order to generate an accurate adaptation (Surjono and Maltby 2003). The characteristics, or dimensions, that are employed as axes of personalisation have expanded to include a great number of a learners profile properties, a sample of which can be found in Table 2.1.

Selecting the dimension upon which adaptivity is based becomes very important to the operation of the system. Not all characteristics will necessarily improve a learners experience in the learning process (Surjono and Maltby 2003). Selecting the incorrect dimensions will not benefit the learner educationally and simply result in a model that becomes unnecessarily complicated (Carver, Hill et al. 1999). Selection of dimensions on which to adapt will be very much dependent on the nature of the learning environment. For example, the dimensions employed to adapt an eLearning game or simulation might be very different to the dimensions employed to adapt a more traditional eLearning course. The importance of each learner's adaptive dimension may also be very different. Over time this importance may also vary with a learner as they progresses through their learning environment (Carver, Hill et al. 1999). Third generation AEH systems also allow adaptive dimensions to have a much finer granularity than found in previous generations (Carver, Hill et al. 1999). An adaptive dimension can have a range of scales with many intermediate values in order to more accurately model learners.

Table 2.1: Sample of Personalisation Dimensions (Wade 2009)

User Age	Interaction (with display)
User Disability	Group membership
Subject Pre-requisites	Group activity
Display Device	Deadline/Event
User Role	Challenge
User Motivation	Difficulty
User Language	Relevance
Preferred Modality	Progress
User Prior Knowledge	Operating conditions
User Competencies	Speed of device
User Experience/History	Volume of information
User Objectives	Social Footprints
User Emotion	Cost
User Preferences	Complexity
Relationship to other subjects	Activity
User Interests	Availability
Time	User Goals
Performance	User Behaviour
Level of Control	Process (rules)

The major difficulty with employing multi-dimensional adaptivity is gathering the data for each learner model. This has typically been accomplished in less sophisticated systems by asking the learners to fill in a questionnaire, complete an achievement test or observing the learner as they progress through adaptive course (Surjono and Maltby 2003; Carver, Hill et al. 1999). Systems that incorporate multi-dimensional characteristics require a great deal more input than earlier generations of AEH instances (Carver, Hill et al. 1999). This can be complicated further when considering each dimension may have a large range of values and different levels of importance.

Impacts of Adaptive Educational Hypermedia

There are a number of indicative studies that show adaptivity improves the impact of educational hypermedia. While the case is not overwhelming, there are three areas in which personalisation has indicated improvement; these areas include effectiveness, efficiency and user satisfaction. Adapting different dimensions in different learning environments can affect each of these aspects in a different way. Adapting dimensions may also affect more than one of these aspects. The following sections describe each aspect of educational hypermedia.

It has been indicated in numerous studies that adaptive educational hypermedia can be more **effective** than educational hypermedia systems that do not include personalisation (Park and

Lee 2004; Brusilovsky 2001a). Learners can process information more effectively if it is presented to them in a manner that is easy for them to follow and understand. A course could, for example, be adapted on a learners' intellectual ability. A learner with a low intellectual ability can process information more effectively with a more structured and less complex approach to instruction. Compare this to a learner with a high intellectual ability that would process more effectively with a less structured and more complex approach to instruction (Snow and Lohman 1984).

The speed at which learners can be taught can also be greatly improved by adaptivity (Brusilovsky 2001). Learners can process content much more **efficiently** if it is presented to them in a personalised manner. Adapting a course on a learners' prior knowledge is an excellent example of how personalisation can increase a learners' efficiency. A learner with a low level of prior knowledge will require a lengthy introduction to the basic concepts of a subject. Compare this to a learner that has a very high level of prior knowledge who can bypass the basic concepts and proceed to more complex content. The higher the level of prior knowledge the less instructional support is required which creates a much more efficient learning experience (De Bra, Eklund et al. 1999).

Furthermore, research currently being conducted has indicated that personalisation may improve the efficiency of courses by presenting content in a manner which is more suited to individual learners. This could be based on using different examples for individual learners or employing different vocabularies or contexts; topics are presented in particular ways that allow them to be absorbed more readily.

Personalisation also has a significant impact on a learner's **user satisfaction**. This describes how much a learner enjoyed and was engaged by their learning experience. A learner's user satisfaction has a direct effect on a learner's interest and motivation. A course that is not adapted to a learner may not hold their attention and reduce the effectiveness and efficiency of the learning experience. User satisfaction is effected by several aspects of a course and adaptivity can be used with many different learners' dimensions in order for it to improve. For example, it has been argued that by adapting a course based on a learner's learning style, the course can greatly affect a learner's self-perceived competence (McClelland 1965). If a learner is taught using a learning style that is not suited to them they may find the course

difficult to follow and believe that it is their own competence that is at fault. This would in turn create anxiety and effect a learner's motivation and enjoyment of the course.

It should be noted however that there is some contention when it comes to the validity of employing learning styles. While some have argued that they are an effective means of grouping learners others argue whether or not these groupings are actually important. The assertion is that people adopt different learning styles depending on the domain/subject/experience level. Some studies have shown that categorising learners based on their learning styles has shown no significant effect (Brown 2007).

As previously stated there is no single dimension that affects each of these impacts on adaptive education. Different educational situations need to employ appropriate dimensions at different times in order to create a more effective, efficient and enjoyable learning experience. There can also be a great deal of crossover with dimensions affecting several areas of impact. For example, by adapting a course based on a learner's cognitive style it may improve not only a learner's efficiency but also their enjoyment of the course. The challenge with employing such multi-dimensional approach is twofold. Firstly, selecting and balancing dimension so as to optimise the personalisation for each situation. The second challenge is creating tools and methodologies that easily and efficiently allow multi-dimensional adaptive courses to be authored.

Techniques for Applying Personalisation

A number of different techniques have been developed in order to apply and implement personalisation in adaptive hypermedia systems. The goal of these techniques is to alter either the structure of the hyperspace or the content contained in the leaf nodes of a hyperspace. These techniques have generally been divided into two categories (Surjono and Maltby 2003; Calvi and Cristea 2002); Adaptive Navigation (Abdullah and Davis 2003); and Adaptive Presentation (Brusilovsky 1996). The following sections describe the principals involved in each of these categories.

Educational hypermedia courses are delivered to learners in the form of a hyperspace. This allows the learner to engage with the content of a course in an exploratory manner jumping from one leaf node of the hyperspace to another through hyperlinks. Navigating the

hyperspace in a non-linear manner can be rather slow and inefficient. In order to improve the efficiency, effectiveness and enjoyment of the course adaptivity can be applied to the navigation of the hyperspace. **Adaptive Navigation** provides guidance to the learner by directing them through routes in order improve their learning experience. This is accomplished by personalising the structure of the hyperspace to the learners needs. There are several mechanisms that are categorised as Adaptive Navigation. Each of these techniques restructures or deploys the hyperspace links in a different manner. Specific mechanisms of Adaptive Navigation include annotation, sorting and link hiding. Adaptive Navigation is the most common category of adaptive technique that is employed in AEH systems due to its effectiveness and relative ease of application.

The second category of personalisation techniques used within educational hypermedia is **Adaptive Presentation**. While Adaptive Navigation focuses on the structure of the hyperspace to create personalised experiences, Adaptive Presentation is applied to customising the content of a course. This type of personalisation adapts the data, media, tasks, etc. that are contained within each of the hyperspaces nodes. The content of these nodes is personalised by customising how and what is presented to the learner based on the values contained in the dimensions of their user model. This is achieved by initially fragmenting the components of a course into discrete units known as Pagelets. Pagelets are then included, excluded or elaborated on within a hyperspace node in order to create a customised page/node for each learner. For example, one learner may require a detailed complex explanation of a piece of content while another learner only needs to be presented with a short simple description. Similarly, two learners may need to be presented with the same information but require it to be explained with different approaches, through different media types for instance. Pagelets can be fragmented at different levels of granularity from interchanging a complete page to a single word that appears in a piece of text. There are several techniques used to achieve adaptive presentation including frame-based adaptivity, stretch text and conditional text.

2.3.3 Analysis of Adaptive Educational Hypermedia

There are several key aspects of adaptive educational hypermedia that are highlighted by their evolution. At the very least AEH systems need to incorporate third generation strategies. For example, they should employ a multi-model approach using user models, content models

and adaptivity models where appropriate. AEH systems should also incorporate more than just text components and should deliver media rich content and immersive experiences.

The other areas discussed in the previous section do however emphasise that AEH systems need to improve in other aspects of their design, delivery and implementation. For example, adaptivity needs to be accessible to those without technical background or knowledge of adaptive hypermedia. This may only be achieved with the use of innovative and accessible authoring tools. Future generations of AEH systems also need to be considerate of what type of adaptive dimensions are being used. For example, the adaptive dimensions relevant and which improve an immersive simulation based learning experience (such as a game or simulation) may not be usefully applied to typical classroom style adaptive course. Techniques employed to apply personalisation may also become more sophisticated. While adaptive navigation and adaptive presentation will continue to be used, more advanced techniques will also become available. These could, for example, be related to interlinked adaptive learning interactions such as reflection points or assessments, sophisticated ‘buddy’ systems or advanced external services.

The challenge for future AEH systems is placing all of these components together in a manner that is accessible to subject matter experts. A balance needs to be found with accessibility and the impact the personalisation has on the end learners. An adaptive dimension or technique may be very effective at creating a better learning experience but might be so complex that it is not easily accessible for a subject matter expert. Conversely, an adaptive model might be easily composed but offer little in the way of educational improvement of a course. Future generations of AEH systems should be developed by non-technical subject matter experts in authoring tools built to compose courses in specific learning environments. Different learning environments could be classroom based, book based, simulations and games. Even within these general classifications different types of composition tool may be required for different learning environment sub-classes. Attempting to capture many types of adaptive dimension, adaptive technique or content modelling for several different learning environments may result in not accommodating any of them adequately. Future generations of AEH systems should be composed by accessible composition tools that are designed for a specific purpose.

2.3.4 Educational Theory of Adaptive Hypermedia Systems

Earlier in this chapter, section 2.1, constructivism and cognitivism were described in general terms; the following two sections discuss these theories as they relate to adaptivity based learning.

Constructivism

The nature of educational hypermedia systems aligns very effectively with the principals of constructivism. Hypermedia systems are non-linear environments that allow learners to actively construct their own knowledge of a domain. Learners are allowed to move from one hyperspace node to another creating their own meaning and linking new ideas. Control is delegated to the learner giving them a sense of empowerment. However, while this approach to education is appealing and effective it does present some impediments. Specifically, the open-endedness of educational hypermedia, and environments of this nature, can be rather overwhelming for a learner. If all the content of a course/domain is accessible and available to a learner without any direction there is substantial risk of the learner becoming lost. A learner can quickly and easily become disorientated while exploring. Furthermore, a learner may never engage in the intended areas of learning or be overloaded with content that is too advanced. In order to address these issues that can occur with hypermedia systems adaptivity can be employed. Personalisation can be used to create educational hypermedia systems that promote constructivism in a much more directed manner. The learner is still constructing their own view of the content, aligned with the constructivist approach to learning, but they are doing so in a more effective and efficient manner.

Cognitivism

The cognitive view of learning differs to constructivism as it focuses on the mental processes that occur with a learner. Cognitivism views the knowledge domain that a learner creates to be consistent and that all learners will eventually form an identical view of a domain. This is opposed to the constructivist view of pedagogy but unlike the behavioural theory of learning cognitivism does consider a learner's dimensional characteristics. Cognitivism recognises that individual learners will process what they are learning in different ways based on their particular circumstances within the learning environment. Cognitivism believes that while all learners will eventually converge to the same view of a domain they will each take their own individual path. Cognitivism takes into account several aspects of how a learner forms their

processes such as how they are taught in different ways (learning styles) or that they may be at different stages of learning (prior experience). In order to assist this theory of learning adaptivity can be employed to create a course that is customised to a learner's cognitive processing needs. Personalisation directs a learners cognitive thinking in order to create environments that are educationally more effective and efficient as they are suited to how a learner learns.

2.3.5 Application of Educational Theory to Next Generation of AEH Systems

As outlined in section 2.3.3, future generations of AEH systems will need to balance ease of authoring with effective adaptivity. Selection of adaptive dimensions is important and should be relevant to the learning environment and adaptive techniques being employed. While adaptivity is based on these attributes it should still be considerate of relevant underlying theory. For example, adaptive dimensions and techniques should also consider how they will guide a learner so as to improve the underlying constructivism. Adaptivity based on constructivism is typically a navigational based technique so dimensions which improve the navigation of a learner should be selected. This type of adaptivity can be viewed as being rather subtle. The design of which within a system should be controlled by a subject matter expert so as to insure that the underlying educational theory is effective.

Adaptivity related to cognitivism is rather more direct. The goal of the cognitivism is to direct all learners to a specific way of thinking. The manner in which this is realised within an AEH system is typically achieved with direct learning interventions which adapt to learners so as to guide them to a particular knowledge goal. These could, for example, be based on coaching interventions or reflective prompts.

2.4 Adaptive Educational Simulations

The personalisation of educational simulations is a very new area of research. There have been very few educational simulations developed which could be described as being truly adaptive. Most systems that characterise themselves as personalised learning simulations view adaptivity as a rather coarse reactionary process that does not consider a learners user model, prior knowledge or previous actions. These systems would understand that adaptivity has occurred in a system if it simply reacts to a learner taking an action or making a decision. All reactions of the system for each learner would be identical irrelevant of their situation, prior experience or any other personal data. This broad view of "*adaptivity*" is very removed

from real personalisation as outlined in AEH systems. Most adaptive educational simulations that do employ a more accurate definition of personalisation execute it in a rather unsophisticated manner. For example, these systems will adapt an environment based on a learner being in a particular group such as skill competency (beginner, intermediate or expert). Little progress has also been made in expanding simulations' adaptivity beyond single learner dimensions such as prior experience and the learning outcomes to be achieved.

This section examines several different aspects of adaptive educational simulations. Initially the educational theory of applying adaptivity to educational simulations is described. This is followed by a short description of some of the existing research that has been completed in this area. Finally, adaptive educational games are described as there are many parallels between them and adaptive educational simulations.

Educational Theory

Examining the educational theories of simulations and adaptive hypermedia the most obvious similarity is that both approaches align with the constructivist theory of learning. Simulations and hypermedia implement educational environments that are exploratory in nature. Both approaches allow the learner to construct their own views of the domain they are being taught. It has been previously highlighted that educational hypermedia systems employ personalisation in order to create a more focused learning experience. Applying these principals of customised constructivism to simulations should be similarly beneficial. The need to employ adaptivity with educational hypermedia is perhaps more immediately obvious than applying it to simulations. Visualising the model of a large hyperspace used to implement an educational environment is more approachable than considering the models used to describe educational simulations. This is perhaps due to the fact that the types of models used to describe simulations are dependent on the category of simulation that has been implemented; physical; iterative; procedural; or situational. Each category of simulation employs very different model types to describe the learning environment that they create. The idea of how adaptivity might work and how it would be beneficial is more difficult to deliberate than that of a hypermedia model.

One approach to visualising simulations models so they are more accessible is to regard them as decision trees. At any particular point within a simulation a learner must make a decision

how to proceed. Each decision a learner makes guides them to another decision point or to an end point of the simulation. Due to their linear nature procedural and situational simulations particularly lend themselves to this interpretation of a simulations underlying model. The decision tree models produced by physical or iterative simulations would be far more complex. There would potentially be a vast number of interconnections between decision points with learner choices affecting the states of different components within the simulation. Iterative simulations may in fact have an infinite number of branches within a decision tree model.

Approaching simulations in this manner emphasises how adaptivity would be beneficial for a learner. A learner needs to be directed and guided through the simulation similarly to how Adaptive Educational Hypermedia assists a learner. Considering simulated environments as a domain that is to be engaged with and explored highlights how adaptivity could be applied and how it would be similarly beneficial to learners. Personalisation can focus a learner in the direction that is most suited to their needs creating a more efficient and effective learning experience.

Applying adaptivity to educational simulations also creates a learning environment that personalises situated learning. Traditional non-adaptive situated learning does not consider a learner's individual characteristics. Learner dimensions such as prior knowledge and possibly more importantly prior experience with a particular skill set are not recognised as important factors in the learning process. The goal of situated learning is often to allow learners to practice skills rather than for them to gain knowledge (as knowledge would be traditionally viewed). Applying personalisation to situated learning results in a learning environment that not only strongly reflects reality, but also adapts to the needs of the learner within that reality. A learner practices the skill they are learning but within an environment that accounts for their level of competency with that skill. Adaptivity applied to a situated learning environment focuses on adapting towards improved interaction with the environment compared to traditional AEH that focuses on adapting towards improving a learner's knowledge.

The cognitive processes involved in learning and practicing a skill will also be very different to the cognitive processes involved in a learner increasing their knowledge. Applying

adaptivity to traditional educational hypermedia environments allows a learner to improve their knowledge more effectively and efficiently. The cognitive processes that occur focus on how the learner builds their view of the world. What these processes do not do is teach the learner how to improve skills or perform tasks. A learner might understand the principals involved in executing a skill or completing a task but the cognition that has occurred does not necessarily teach them the intricacies of interaction. Adaptivity allows a learner's cognition to be focused so as to improve the effectiveness and efficiency of interaction.

Existing Research

As previously stated, research in the area of adaptive educational simulations is very limited. There have been very few systems developed that deliver truly personalised educational simulations. There have been even fewer research projects that examine the theoretical educational underpinnings of such systems. This section briefly describes three projects that have produced adaptive educational simulations.

The first system presented was developed in the Department of Medical Physics in the University of Patras, Greece. It was designed to teach radiology skills using procedural simulations that incorporate adaptive hypermedia (Costaridou, Panayiotakis et al. 1996). This system uses hypermedia to model the building blocks of an interactive procedural radiology environment that can be dynamically restructured based on the characteristics of the learner. The system was however not formally tested by clinical personnel although their initial findings were very positive. The designers of the system felt that it would offer advantages in speed, efficiency and enhance the simulations interactive environment.

Another project that has combined principals from adaptive hypermedia and educational simulations is GRAPPLE (GRAPPLE 2011) - Generic Responsive Adaptive Personalized Learning Environment. This project, now complete, involved several different European institutions. GRAPPLE had a number of different goals including delivering a suite of authoring tools that could be used to develop adaptive procedural simulations. More specifically the objective of the GRAPPLE project was to deliver to learners *“a technology-enhanced learning (TEL) environment that guides them through a life-long learning experience, automatically adapting to personal preferences, prior knowledge, skills and competences, learning goals and the personal or social context in which the learning takes*

place” (GRAPPLE 2011). There were several key innovations in this project including the development of a generic adaptive learning environment (GALE), a generic user model framework (GUMF) and an authoring tool (GAT). The authoring tool, which “*enable[s] teachers and authors to create, maintain and store adaptive [procedural] simulations*”, was viewed to be “*flexible*” and “*intuitive*” (Power, Dagger et al. 2010). The GRAPPLE project, which included over a dozen European partners, was very successful. The project passed all of its reviews and “*opened up a new way of learning and support for life-long learning*” (GRAPPLE 2011).

DiscoverNet (Belkada, Cristea et al. 2001) is another system that has been developed as an adaptive simulation-based learning environment. It however does not employ adaptive hypermedia but instead uses knowledge based neural networks. This is a hybrid of intelligent systems which exploit features of knowledge-based systems and neural network methodologies (Towell, Shavlik et al. 1990; Fu 1989). DiscoverNet uses its neural networks to adapt its user interface to the current learner based on their skill level. This system has yet to be extensively evaluated but initial testing of its prototype have been completed and returned positive results.

Existing research in adaptive educational simulation is obviously limited and does not present any conclusive results. Although the systems described above indicate some positive initial findings they have not been extensively evaluated. These projects have not yet shown how adaptivity improves the effectiveness and efficiency of simulation based learning environments. Other research, such as that completed by Leutner (Leutner 1993), which has specifically investigated the benefits of adaptivity in simulation based learning environments have evaluated only very crude personalisation. Furthermore, their evaluations seem to have mostly examined the educational benefits of simulations in general and not the effects of adaptivity.

The DiscoverNet and radiology systems that were developed were also very specialised. They have both been designed for very specific purposes with adaptivity hard coded into their design. They are not easily extensible or reusable and are also limited by the number of learner dimensions they employ for adaptivity. Unlike these two projects GRAPPLE offers a flexible framework for developing multi-dimensional based personalised simulations.

GRAPPLE has shown that it is possible for adaptivity experts to quickly and easily produce sophisticated adaptive procedural simulations.

Adaptive Educational Games

Adaptive game based learning (Torrente, Moreno-Ger et al. 2009; Prince and Davis 2008; Moreno-Ger, Burgos et al. 2007;) is a relatively new area of research. Similarly to personalised simulations applying adaptivity to educational or serious games is very difficult and costly. One of the most notable challenges within adaptive game based learning is employing personalisation without compromising the game play of the game. This is less of an issue with educational games as the pedagogy, which is obvious and transparent, is the primary focus with game play a secondary consideration. It is however very difficult to implement personalisation in serious games. Adapting a serious games narrative or presentation could potentially break the game; adaptivity must be controlled and restrained in order for the enjoyment of game to be preserved (Peirce, Conlan et al. 2008).

Educational games and simulations have a great deal of similarities. Educational games can be viewed as simulations where each learner competes to win, often against each other or against computer controlled players. Both educational simulations and games are interactive, engaging and self-motivating. They are also both typically time consuming, expensive and difficult to develop. Similarly to educational games one of the most notable challenges with applying adaptivity to simulations is to do so in a manner that retains the fidelity of the simulation. Personalisation should improve the educational effectiveness of the simulation but insure that the simulation remains realistic enough to be an effective learning tool. It is also important to note that educational games often incorporate elements of fantasy or fiction into their narrative and design which would not be seen in simulations.

The educational benefits of applying adaptivity to games are highlighted in a number of studies. For example, the simulation based game Tactical Training & Cultural Training System (Johnson, Wang et al. 2007) that is used in military training programs has shown effective learning outcomes can be achieved through adaptivity. Games developed in the ELEKTRA (ELEKTRA 2008) project have also returned positive results including a higher degree of absorption, an increase in confidence and a lowering of a learners extraneous cognitive load (Peirce, Conlan et al. 2008). Other systems which present positive results

include Prime Club/Climb (Conati and Zhao 2004) and games developed on the <e-Adventure> (Moreno-Ger, Burgos et al. 2008) framework such as the HCT (Hematocrit) Blood Test game (Torrente, Moreno-Ger et al. 2009).

The techniques used to achieve personalisation in adaptive educational games differ to those employed in adaptive hypermedia. While concepts such as Adaptive Presentation exist they must be implemented in a manner that does not diminish the game play of the game such as disrupting the narrative. Several approaches to adaptivity have been developed to achieve personalisation while also preserving game play; for example, adaptivity that is designed at a fine grained level such as Dynamic Difficulty Adjustment (Hunicke and Chapman 2004) which personalises small components of a game without affecting the overall experience. Other systems, such as 80Days (Law and Kickmeier-Rust 2008) adapt at a much higher level through sequencing of story units. This is rather more challenging as the re-sequencing of story units must preserve the narrative of the game through story pacing and by combining linear and modular story forms.

2.5 Summary

This chapter has described the educational underpinnings of simulation and adaptivity based learning related to this research. In order to present such a complex subject, the chapter was decomposed into four sections. The first section introduced the relevant basic theories including constructivism, situated learning and cognitivism. The following section examined simulation based learning and outlined the different categories of educational simulation, key properties of educational simulations and educational theories as related to educational simulations. The third section described adaptivity based learning which included the types of adaptive educational systems, their key elements and impacts, and educational theories as related to adaptive learning. The final section presented a composite of both simulation based learning and adaptive based learning educational underpinnings.

The goal of this chapter was to present an analysis of several aspects related to the educational underpinnings of simulation and adaptivity based learning. There are some key findings highlighted by this analysis that should be incorporated into future generations of adaptive educational simulations. Firstly, to allow greater adoption of adaptive simulations there is a need for intuitive and accessible authoring tools. These authoring tools should allow non-technical subject matter experts to compose and implement adaptive simulations without

the need to write code or create any rules; i.e. adaptive techniques should be accessible. Furthermore, authoring tools should be designed specifically for composing particular types of learning environments and should incorporate relevant and effective learning dimensions.

3 State Of The Art Survey of Authoring Tools for Adaptive Educational Simulations

After conducting an extensive search to identify a representative sample of authoring tools used to compose adaptive soft skill simulation, the author concluded that no such tools currently exist. Therefore the author divided the state of the art review of authoring tools for adaptive educational simulations into two sections. The first section, 3.1, presents a review of a representative sample of soft skill authoring tools. The second section, 3.2, presents a review of a representative sample of current personalisation course authoring tools. The final section of this chapter, 3.3, presents a summary of the chapter. It should be noted that these authoring tools were surveyed at different stages of this research, for the purposes of this thesis all authoring tools were re-examined and any relevant updates are included.

3.1 Soft Skill Simulation Authoring Tools

This section describes four authoring tools that are used to compose online soft skill simulations. The majority of soft skill authoring tools have been developed in the private sector; very few such systems have been produced in academia. As such, three of the four authoring tools reviewed are industry based. These four were selected as they give an excellent account of the different types of authoring tools that are generally available. They each offer a unique approach to authoring. The VISION (Fitzmaurice, Armstrong et al. 2007) authoring tool was chosen for the survey as it offers an interactive and very intuitive display of the dialogue models upon which the simulations operate. Experience Builder (ExperienceBuilder 2012) was examined as part of the survey as it offers a unique approach to the authoring process. Captivate 5 (Captivate 5 2010) is also examined in this survey and is typical of most types of composition tool that can be found. The final application included in the survey is SimWriter (NexLearn 2010); it was selected as it is one of the few authoring tools that employs both a solid authoring process and an intuitive representation of dialogue. While each of the authoring tools examined are capable of composing online soft skill simulations, Captivate 5 can also be used to create other types of simulations such as procedural simulations. For the purposes of this survey, only the authoring tools proficiency in composing soft skill simulations is examined and evaluated.

In order to survey the authoring tools they are examined under carefully selected criteria. These criteria include, as expected, an overall description and evaluation of the systems.

Other criteria used to examine the composition tools involve the approach used to represent the dialogue, the user interface and the composition tools authoring process. These criteria were used as part of the survey as they are vital to competent composition of soft skill simulations, as outlined in a support framework (Gaffney, Dagger et al. 2007a) developed as part of this research. Each of the criteria used to examine the composition tools are detailed below.

Description: The authoring tools general description is detailed which includes its origins, overall goal and information regarding the soft skill simulations that are produced.

Dialogue Representation and User Interface: An integral part of creating a soft skill educational simulation is the authoring of the dialogue upon which the simulation operates. The composition tool needs to allow the author to effectively and accurately link the components of the model to create the dialogue.

Authoring Process: The authoring process describes the procedure that the author follows in order to create the online soft skill simulations.

Evaluation: The final survey criteria describes the evaluation of the authoring tool. It details a critical account of the advantages and disadvantages of using the tool to author online soft skill simulations.

The following sections describe VISIO_n, Experience Builder, Captivate 5 and SimWriter with respect to these criteria. This is followed by a section that compares and contrasts the authoring tools under these criteria.

3.1.1 VISIO_n Composition Tool

VISIO_n (Fitzmaurice, Armstrong et al. 2007), Virtual Interviews for Students Interacting Online, began in 2002 in Trinity College Dublin as a joint project between the Department of Psychiatry and the Knowledge and Data Engineering Group of the Department of Computer Science. Its principal purpose was to create online simulations used to teach psychiatric interviewing techniques. This was achieved by allowing the students to take on the role of a doctor within the simulations and allowing them to interact with simulated patients. Through this interaction the students' communication skills with real patients were greatly improved.

The simulations created were very effective. They were delivered to the learner online as a combination of media files and hyperlinks. Each hyperlink selected by the learner directs them through a different path of the dialogue to another media file and a new set of hyperlinks. The accumulated sequencing of the media files creates a realistic psychiatric interview for the students. Due to the intricate nature of the dialogues within psychiatric interviews the simulations that were created were very large and complex. To accommodate the authoring of such complex simulations by domain experts the VISION composition tool was developed.

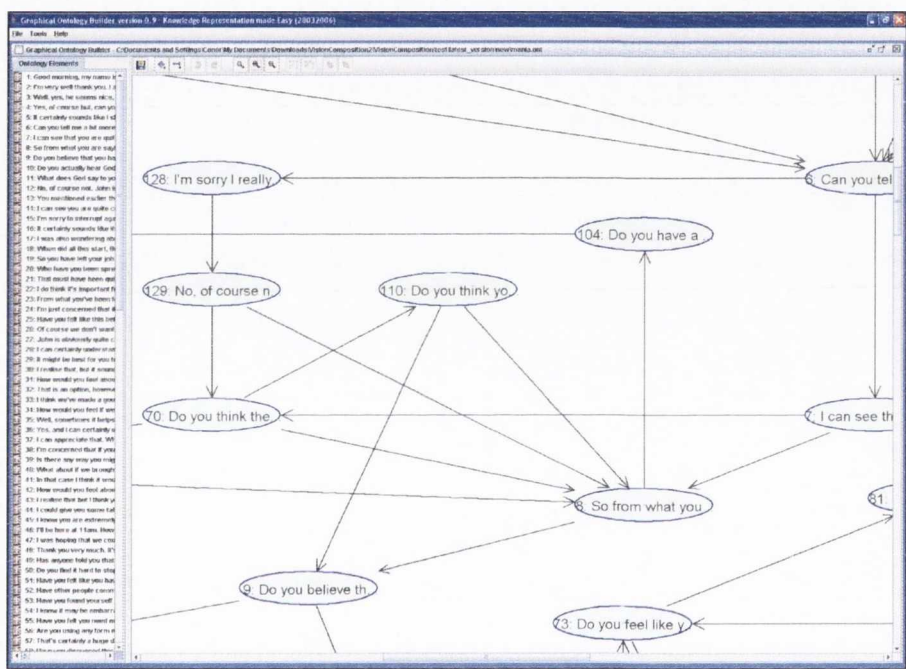


Figure 3.1: VISION Authoring Tool

Dialogue Representation and User Interface

The models upon which the simulations operate are based entirely on the dialogue that would occur in the simulation. To allow the author to compose these models, known as dialogue models, they are separated into their most basic components. Each of these components represents an initial statement and corresponding response that would occur within the dialogue. To illustrate the flow of the simulation the author connects each of these components to create the overall dialogue model. Once the dialogue model is completed, each component of the dialogue could be associated with an appropriate video file.

The components of the dialogue are graphically represented as ellipses in the authoring tool. The different paths, or hyperlinks that the learner can choose within the simulation, are represented as directed connections between the ellipses. To facilitate this representation, the dialogue model is visually expressed as a graph of nodes and connecting edges. Within this representation the author graphically adds, moves and connects the different sections of the dialogue with no restrictions of movement or expression.

The user interface, as seen in Figure 3.1, is composed of three sections. The largest area is the main canvas where the dialogue model is composed using connected dialogue sections to represent all possibilities that can occur within the online simulation. Above this is a tool bar allowing the author to add nodes and edges to the main canvas, zoom in and out, undo/redo and save. To the left of the main canvas there is also a list of all nodes that have been created.

Authoring Process

The authoring process used to create the online simulations is simplistic. The only concern for the author is to create the dialogue models upon which the simulations are to be based using the menu bar and main canvas. To edit the details of each dialogue section the author double clicks the ellipse for a popup menu, which allows the author to fill in the initial statement and response for each section. Once the dialogue model is completed and the all the video files are created the location of the corresponding video file locations can be added to the properties of each dialogue section. When the author has completed the dialogue model they can publish the online simulation. This is accomplished by uploading the dialogue model onto the VISION web server. It can then be accessed by the author for examination and finally made available to the learners.

Discussion

The representation of the dialogue model within the VISION authoring tool would seem to be highly effective. The approach used is similar to very successful dialogue management tools (Luz 2000); systems not used to describe or represent simulation models. Such graphical representation is intuitive and easy for the author to follow. It allows the author to compose the dialogue models quickly and efficiently. The composition tool allows the author a high

degree of freedom to express the dialogue model they are creating. The author is not restricted in the types of connection they can make between the different components which can be created quickly within the main canvas. The author is also not distracted from composing the dialogue with less vital details of the simulation, such as detailing the front end of the simulation. For example background colour, location of hyperlinks, etc.

There are, however, two main deficiencies with composing soft skill educational simulations using the VISIOOn authoring tool. The first of these is the lack of formal process incorporated while creating the online simulations. As the author composes the dialogue model, there is no authoring framework to aid how they structure the components of the model. Without such a process, design is difficult and the models can easily become cluttered and difficult to manage. There is also a danger that the simulations will not achieve their educational objectives as the author does not explicitly state what these are.

The second issue with using the VISIOOn authoring tool is that, with the exception of the zoom functionality, there are no navigational aids provided to the author in the main canvas. With the approach of representing the dialogue model as a graph with nodes and edges, the main canvas can become very congested. This issue is particularly magnified when the author is creating a very large and complex online simulation. Without assistance the author can easily become disorientated within the main canvas.

3.1.2 Experience Builder

Experience Builder LLC (ExperienceBuilder 2012) began as a spinoff company in 2000 from Institute for the Learning Sciences at Northwestern University. Its primary goal was to simplify the way in which soft skill simulations were created and to provide a tool to allow non-technical authors to create their own simulations. Experience Builder now provides authoring tools for the composition of both procedural and soft skills. It is the later authoring tool which is the focus of this survey.

The educational simulations created by Experience Builder are deliverable in HTML and Flash format. The simulations created present the learner with a short video clip and a selection of hyperlinked options within each step. In order for the learner to interact with the simulation they select an option after each video clip which directs them to the next video

clip and alternative set of hyperlinked options. These simulations are primarily used within the business domain, teaching a wide variety of skills from sales to customer service.

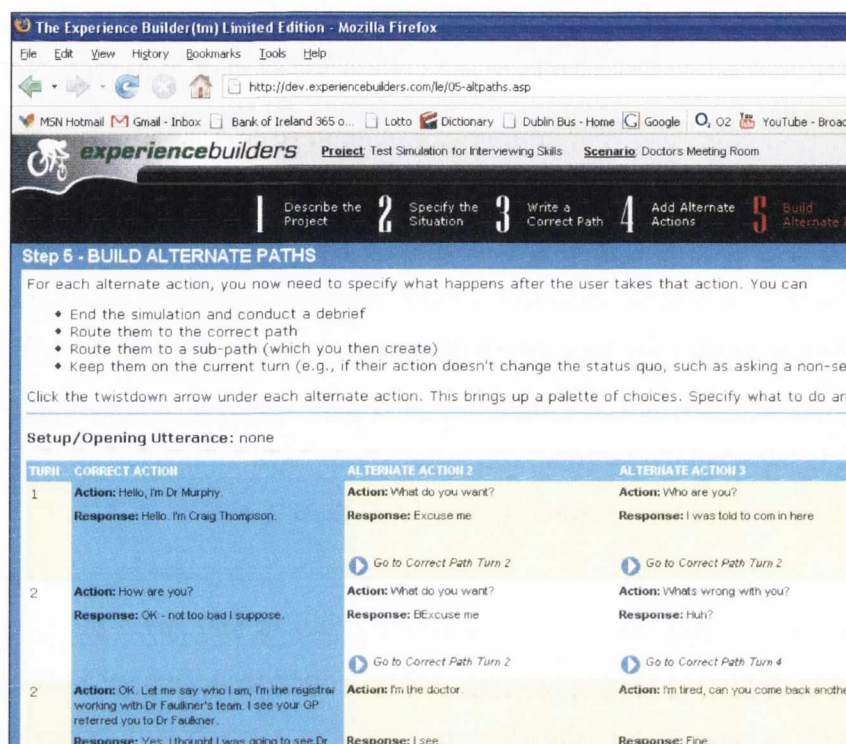


Figure 3.2: Partial Screen Shot of the Experience Builder Authoring Tool

Dialogue Representation and User Interface

Unlike most soft skill authoring tools, Experience Builder is an online application which allows the author to compose educational simulations through a web browser, as seen in Figure 3.2. The details of the simulation are captured using a precise authoring process which consists of eight steps which the user follows.

The simulations are again based upon the dialogue that takes place in the simulation which is developed over three steps of the authoring process. For each step the dialogue is represented in a different manner and is displayed on a separate web page. In the first step the author describes the optimal route the learner would take within the simulation. This is visually represented with text which is decomposed into dialogue components consisting of a 'User Action' and 'Character Response'. The text components are separated using boxes and appear on the left side of the web page. The components appear in the same sequence as they

do when the simulation is in operation. These components are later associated with the suitable media files.

The next step of authoring the dialogue is to create alternative actions. These are the possible choices the learner could take within the simulation which deviate from the optimal path. Similarly to the components of the optimal path, they are also composed of a User Action and Character Response and are represented using text and boxes. Again, media files are associated at a later stage.

The final step of composing the dialogue is to create the alternative paths. These allow the author to connect the alternative actions they created in the previous step to the optimal path. This step creates the different branches which will become the hyperlinked options within the simulation.

The main component of the user interface is a menu bar which is placed across the top of the web page. It displays the eight steps of the authoring process that are used to create the simulations. Each step is given its own ASP web page (Active Server Pages). By clicking on any of the steps the authoring tool will open the associated page. This allows the author to update the details of the authoring process step and save it to the Experience Builder server. All details of the process steps properties are stored and edited using text contained in boxes. The only notable exception being the specification of the links between the different components of the dialogue, these are input using a drop down menu which lists all possible components available.

Authoring Process

As previously mentioned, the authoring process is central to the development of the online educational simulations in this authoring tool. The author follows each step from one to eight to compose their simulations, each of these authoring steps are examined in detail in the following paragraphs.

1. Describe the Project: The author names and describes the simulation they are composing using the text boxes provided. This step also allows the author to create learning objectives within the project which detail the different educational goals the author would like to

achieve with the simulations. By doing this the author provides a sound pedagogical framework upon which the simulations are to be based.

2. Specify the Situation: The author describes the situation in which the simulation is to take place, such as a physical description of the simulation. This step also includes the '*Learner's Goal in the Situation*', a short description which is displayed to the learner. It details what the learner should be attempting to accomplish within the simulation.

3. Write the Correct Path: In order to create the dialogue upon which the simulations operate the author initially describes the '*Correct Path*' of the simulation. This is the optimal and most direct route through the simulation. It is represented using a series of text boxes which detail the User Action and Character Response which are displayed in series one under another.

4. Add Alternative Actions: These are the alternative actions (or options) that the learner can choose from within the simulation. An '*Alternative Action*' is again composed of a User Action and Character Response similar to the components of the Correct Path.

5. Build Alternative Paths: This step allows the author to indicate the hyperlinks that connect the alternative actions created in the previous step to the Correct Path. The author must decide on the outcome of each action created. There are three possibilities available to the author; the Alternative Action can link back into the Correct Path, branch away from the Correct Path to another sequence of dialogue components that the author would then need to create or link to nothing, indicating the simulation had ended.

6. Create HTML Pages: This step of the authoring process allows the user to add additional HTML pages to the simulation. For example, the author can include a web page that introduces the simulation in the beginning to describe the simulation to the learner.

7. Publish and Download: This step allows the author to choose the skin of their simulation. This indicates the look of the soft skill simulations created. It includes the location in the web page of the video clips, background colour, etc. The author is then allowed to publish the

simulation online and is also given the option to download the simulation in HTML or Flash format

8. Perform Optimal Tasks: The purpose of this step is to allow the author to associate media files with appropriate components of the dialogue and complete other administrative operations. These include creating backup of the dialogue model or completing a spell check.

Although the steps are numbered and the author is expected to complete the previous steps before continuing, they often need to make adjustments or add additional information in their simulation. The author is therefore provided with the freedom to iterate through the authoring process multiple times, not necessarily completing every step in each iteration.

Discussion

One of the most unique characteristics of this authoring tool is that it is an online application. This is not only convenient but also allows several authors to collaborate. While having the authoring tool online is a novel approach, it is in fact the detailed authoring process which is the most compelling feature of this composition tool. The process provides the necessary structure to the author to create a complete, educationally sound soft skill simulation. The author is encouraged to detail all aspects of the simulation. The most important aspect of the process is the inclusion of learning outcomes which assists best practice and pedagogical design.

Other important steps included in the authoring process are the scenario and situation descriptions. These give the author a clear overview of the simulation they are creating and which assists them in their composition. The approach of initially developing an optimal path is also very useful as it provides the author an excellent starting point. The author is also not distracted with the details of the front-end of the simulation as composition is centred on the authoring of content. While the authoring tool does allow the author to influence the look of the simulation, content is given the highest priority.

The major disadvantage using the Experience Builder authoring tool is its approach to visualisation and representation of the models upon which the simulations operate. The dialogue created is very difficult to follow, particularly with large simulations. This is mainly

due to the use of text to represent the dialogue. This would be an acceptable approach if visualising a simple sequential conversation but is inadequate if attempting to express all the possibilities that can occur within a dialogue. Another aspect of poor representation is the manner in which different components of the dialogue are linked. A simple text reference without any graphical representation makes it very difficult to follow the flow of the dialogue.

Although specifically designed to be used to author online soft skill simulations the Experience Builder has its limitations. While the process used in creating the simulations is excellent the dialogue representation is confusing and very difficult to use. This is particularly true if creating a large and complex simulation. The interface is most probably restricted as it is an application accessed through a web browser. With the advances of Web 2.0 these limitations could be overcome.

3.1.3 Captivate 5

Adobes Captivate 5 eLearning authoring tool (Captivate 5 2010) was originally released in 2007 and is now in its fifth version. Captivate is aimed at learning professionals, educators and business users. It allows the author to compose software simulations, presentations, quizzes and rich media online simulations. Captivate 5 is very flexible and allows the author to create both procedural and soft skill simulations. The soft skill simulations can be based upon text or media files such as sound or video clips. To allow the author compose the simulations their models are decomposed into smaller components. The author indicates the flow of the simulation by linking these components together. While this approach allows the author to create procedural simulations it is also suited to describing the dialogue that takes place within a soft skill simulation. Captivate 5 was originally reviewed as Captivate 3 as part of this research but subsequent examinations show little change with respect to the development of soft skill simulations. Subsequent developments have continued to focus on refining and developing features which are related to composing general eLearning content and procedural simulations.

Dialogue Representation and User Interface

The Captivate 5 authoring tool decomposes the simulation models into components called 'slides'. Each slide represents a section of the simulation and corresponds to a single step in the simulation. A slide typically consists of text or a media file and a set of hyperlinked

options linking it to other slides. The authoring tool allows the author to edit the presentation of each individual slide with a WYSWYG interface. For example background colour, location of hyperlinks, size of video window, etc. can all be edited. The result of this approach is to combine the dialogue upon which the simulation is based with the visual description of each slide.

The user interface consists of several sections. The first of these is a menu bar which is situated across the top of the application which provides basic functionality. This includes save, undo, preview and publish. The remainder of the screen consists of three tabbed displays which are the '*Storyboard*', '*Edit*' and '*Branching*' displays.

The Storyboard display presents all the slides that have been created in the simulation. These are viewed by the author as thumbnails and present an outline of each of the slides. Within the Storyboard display the author can perform basic tasks with the slides such copy, paste or delete. While in the Storyboard view the author cannot edit the content of slides. To edit a slide the author double-clicks it so it opens in the Edit display. Once in this display the author can add objects to the slide such as media windows, text or hyperlinks. Each of these objects can be moved and resized within the slide. This display also contains another thumbnail view of the projects slides which allows the author to change the slide appearing in the Edit display. The author cannot add or delete slides in the Edit display, this can only be achieved within the Storyboard display.

The final display of the authoring tool is the Branching display, as seen in Figure 3.3, which presents the simulation in a graph like view consisting of nodes and edges. Each slide is represented with a node and is displayed as thumbnail. The nodes are connected with edges which represent the different hyperlinks between slides and show the possible paths within the simulation. The slides cannot be moved in this display, they are fixed in the same order as they were placed by the author in the Storyboard view. To change the position of a slide the author must return to the Storyboard view where the slides can be moved with drag-and-drop functionality. The Branching display does not allow the author to create new hyperlinks. This can only be achieved by editing slide objects within the Edit display. The Branching display does however allow the author to alter existing links between slides. To do this the author initially selects the link they wish to alter. Then, from a property view to the left of the graph

To aid the author in navigating the Branching display a map of the graph is provided. This presents the author with an outline of their position within the display and is particularly useful when the simulation becomes large and complex. Another useful feature in this display is an encapsulation function. This allows the author to group several slides together into a single thumbnail to which reduces the complexity of the graph.

To create a new soft skill simulation the author begins by clicking the '*Record or create a new project*' icon on the opening screen. From a popup window that appears the author chooses '*Scenario Simulation*' from the options provided. The author then uses the '*Project Wizard*' to create a new project. Within the Project Wizard the author names the project and decides the layout of the slides. The author also decides the number of slides the project will initially have. During the composition process the author can add or delete to this number as needed. Once completed the Project Wizard generates the appropriate slides and opens the Storyboard view. The author then edits each slide in the Edit tab by adding the appropriate object using the insert functionality.

Typically the objects added include a media window (media file) or static picture along with text. Hyperlink objects are also included; these allow the author to connect the slides to one another creating the multiple choice options within the online simulation. Connecting of slides is completed in the 'Edit' display using a drop down menu which presents the author with all of slides that have been created. Once all the slides have been created and the connections are completed the author can publish the simulation.

Discussion

The most unique characteristic of the Captivate 5 authoring tool is the Branching display which allows the author to view all the slides that are created along with connections between them. This display is particularly useful as it provides a map and grouping feature to aid the author in navigation. The map shows a miniature version of the main display which uses an orange box to indicate the section of the map that is currently being viewed. The grouping function allows the author to encapsulate multiple thumbnails into a single thumbnail and thus reduces the complexity of the graph. These are particularly useful when the simulation becomes very large and complex.

The Branching display does however have its drawbacks as it is not completely interactive. It does not allow the author to alter the sequence of the slides or graphically make connections. Existing connection routes can be altered using a drop down menu which provides a list of all existing slides but to actually create a new connection the author must return to the Edit display.

Certain aspects of the representation of the connections within the Branching display could also be improved. Any connections that are linked to a previous slide are not shown graphically but instead are indicated using text which refers to the name of the slide they are connected to.

Another usability issue involves the approach of separating the dialogue authoring into three different displays. With a complex simulation the author would find they need to constantly change the display they are using to add, move or edit the slides. This creates a cumbersome and inefficient authoring process.

There is also room for improvement in terms of the representation of the dialogue. While the Edit display performs very well by allowing the author control over the presentation of each slide there is no separation of concerns between the dialogue model and visual details of each slide. With the content and visual display of the simulation blended together the composition becomes focused on the look of the simulation rather than content.

Overall Captivate 5 has the potential of being a very effective authoring tool and employs some useful concepts for creating online soft skill simulations. It is however too focused on the visual display of the simulations rather than the content. While it is possible to build large complex simulations with Captivate 5 the logistics involved make this a very difficult prospect. The authoring tool would be greatly improved with a more intuitive and interactive single display. This could be based exclusively on the 'Branching' display which is already in place.

3.1.4 SimWriter

NexLearns SimWriter (NexLearn 2010) authoring tool is a commercial application used to develop soft skill simulations. SimWriter allows non-technical authors to develop situational simulations without needing to write code or create complex rules. It employs a graph representation of the simulation models and incorporates an easily accessible authoring process. SimWriter allows authors to compose simulations that can include text, audio or video which are interconnected with actions or dialogue links to create a meaningful interactive simulation. The authoring tool also allows authors to customise the design of the front-end of the simulations with a drag and drop interface. Once completed the simulation can be built and tested within the application. The simulations that are produced by SimWriter are Flash based. They are also both SCORM (Bohl, Schellhase et al. 2002) and AICC (2010)compliant.

Dialogue Representation and User Interface

SimWriter decomposes the storyline or logic of the simulation into several smaller components to allow the author to compose the simulations. There are three types of components used to author the simulations; '*character responses*'; '*learner decisions*'; and '*directed feedback*'. These are each represented in the '*map*' display of the authoring tool

with each component allocated a distinctive icon, as seen in the main the left panel of Figure 3.4. The author connects each of the components in the map view to indicate the routes or paths that are available to the learner in the simulation. This approach to representation creates an interactive graph in the map view of the composition tool. The character response elements describe the information that is delivered to the learner. This could consist of a character dialogue, background information or other relevant data. The learner decision elements describe points where the learner must decide which route they must choose from the several options that they are presented. The directed feedback elements describe the responses to the learner’s decisions which can include character responses, encouragement or correction. The map area of SimWriter is divided into sequential rows known as ‘stages’. Within the stages an author can employ any combination of the SimWriter design elements to describe their simulation.

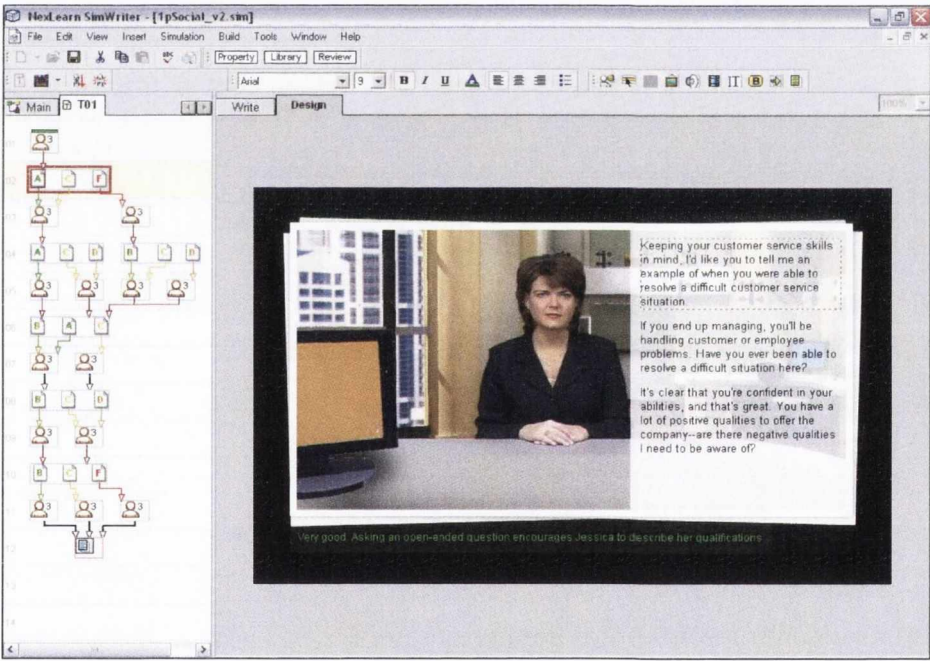


Figure 3.4: SimWriter Authoring Tool

The map display is one of three main areas used to author simulations within the SimWriter authoring tool with the ‘write’ and ‘design’ displays also playing an integral role. The write display allows the author to access, edit and update the properties of the elements contained in the map display. The author can describe the dialogue contained in the simulation as well as the feedback that is to be delivered to the learner along with any other pertinent information (a simulations introduction for example). The write display also allows the

author to use a number of editorial functions. This includes basic functions such as spell check but also includes functions that are more specific to situational simulations such as '*scripting*'. The '*scripting*' function allows the author to view the simulation, particularly the dialogue, in an ordered format so it can be reviewed and edited. The write display also allows the author to add scoring to the decision points within the map. This permits the author to appoint a score to each learner based on the decisions that they have made during the simulation. The final function available to the author allows learning outcomes to be defined and associated with design components contained in the map display. This assists in constructing a model that is educationally effective as it insures authors follow best practice in design and development. Learning outcomes can also be used in reports that can be distributed to the learners.

The third display in the SimWriter authoring tool is the design display, as seen in the main panel of Figure 3.4. It is used to associate text, video or audio with components in the map display based on content that has been developed. The design display also allows the author to customise the front end of the simulation. Authors can add many different design elements to the simulation such as images, buttons, reports or hyperlinks. Elements can be added to the associated map components using drag and drop functionality. The design display also allows the author to build and test simulations in part or completely through.

Authoring Process

The SimWriter authoring process consists of three major phases each associated with the three main displays of the authoring tools interface. The author initially composes the overview of the simulation by adding and linking components in the map display. The author can then describe the properties of the elements in the design phase of the authoring process. The third and final phase allows the author to associate media with components and design the front end of the simulation. Typically an author will move back and forth from the mapping phase to the write phase as they create small segments of the simulation populated with dialogue. The author will only move to the design phase once they have completed the map and assigned the elements with values. The author will typically complete the map and write phases of authoring over many iterations before moving to the design phase. Once the superficial aspects of the simulation have been decided (background colour, component orientation, etc.) video or audio can be recorded and associated with appropriate components.

During the map and write process the author is allowed to associate learning outcomes that they create with simulation components although this is not considered to be an essential step in the authoring process. The author can script sections of the simulation at any point during the authoring process to allow them to assess the flow of the dialogue. The scripting function also allows the author to print manuscripts for the audio/video recording.

Discussion

SimWriter’s representation of the dialogue that occurs within the simulation is conceptually very good. The dialogue models are created using nodes and edges to form a graph in order to describe the underlying logic of the simulation. This approach is simple, descriptive and easy to understand. The display presents a branching representation of the dialogue which is intuitive and allows the author to view and describe all the possibilities that are available to the learner. These models can be used by the system to execute the simulation without the author needing to create rules or program any code.

SimWriter also incorporates a clear and concise authoring process which consists of three accessible and easily understandable phases. The inclusion of steps that allows the author to describe and associate learning outcomes within the simulation is also very useful. This simplistic but effective pedagogical framework assists the author in creating a simulation that is more likely to be educationally effective. This is a feature that most educational simulation authoring tools do not incorporate. The scripting function that can be accessed during the authoring process is also very beneficial. Allowing an author to sanity check (validate) the possible routes that will be available to the learner assists the author in creating a believable learning experience. This is another feature that cannot be found in most soft skill authoring tools.

SimWriter does however suffer from a number of frailties as an authoring tool. The most notable of these is that while the concept of employing a graph in the map view is intuitive, it is poorly designed and executed. The display is restrictive and limits how the simulation model can be structured. The author is restricted by having to place elements in stages (Gaffney, Dagger et al. 2007). This creates a very linear model of the dialogue which is more akin to procedural simulations than that of the expansive models required for soft skill

simulations (Emmendorfer 2009). The map view also employs several different element types in order to model the dialogue. While this is suitable for short simple simulations it is not practical for larger and more complex models which would be more typical of a real world dialogue.

The authoring process, while concise, also has its weaknesses. It is, for example, rather slow as the author must constantly change from the map display to the write display. Rapid development is hampered further due to the work required to develop the design the front end of the simulation. Although SimWriter employs templates to aid the learner, composing the visual elements can very time consuming.

3.1.5 Soft Skill Authoring Tool Critique

This section will compare and critique the different aspects of soft skill authoring. It examines dialogue representation, the authoring process, pedagogical framework, the visual description of the simulation, scalability, navigation and validation.

Dialogue Representation

There are numerous eLearning authoring tools that employ an ontology or workflow based interface for composing and configuring traditional courseware, such as DialogPLUS (Davis and Fill 2007; Bailey, Fill, et al. 2006) and LAMS (Page and Ghiglione 2012). However, soft skill simulations require a rather specialised approach to visualising content. It is clear through the examination of the systems previously described in this chapter that the ability to visualise the structure and flow of a dialogue within the soft skill simulation is critically important. Soft skill simulations require that all possibilities of the dialogue that can occur are described by the author. This description forms the model upon which the simulation operates. The model contains the dialogue components, hyperlink connections between the components, links to all of the content and any other information needed by the simulation. For the author to describe this data they need an interface that is user friendly and intuitive such as that provided by the VISION authoring tool.

While the authoring tools surveyed acknowledge the need for the dialogues to be decomposed into smaller more manageable components they do not recognise the need for a graphical display which sufficiently captures the dialogue. The use of text boxes, such as those used in Experience Builder, is inadequate for this purpose. Text boxes do not offer a

visual interface which is necessary for displaying hyperlinks and routes that could be taken by the learner within the simulation. The author can also only view small portions of the dialogue at any one time. To have a clear understanding of the dialogue the authoring tool should be capable of displaying the whole dialogue and not just small sections.

Although Captivate 5 and SimWriter do provide graphical representation to utilise the dialogue their interfaces are restrictive. For example, the Branching display of Captivate 5 allows the author to view connections within the simulation but all the simulation components (slides) are fixed and connections cannot be moved, deleted or created graphically. Connections can only be created or deleted in the Edit display and the author must use a drop down menu to alter existing connections. SimWriter employs a more interactive view of the simulation logic but is also rather restrictive. Components in the map view must be placed in stages which creates a very narrow dialogue model. This approach allows short simple simulations to be composed but does not scale to permit real world simulations to be authored.

The dialogue representation in the VISION authoring tool does not suffer from the same restrictions. The graph approach used to represent the dialogue in the VISION tool allows the author to quickly create dialogue components, populate them and connect them. This approach also allows the author to maintain an overall view of the dialogue which includes connections between the different components. The only issue raised is the difficulty in navigating the graph produced when authoring large complex simulations.

Authoring Process

The dialogue representation is very important but it is only one aspect of authoring. Soft skill simulation authoring tools must also address the need for a clear and concise authoring process. The authoring process is an outline or blueprint that the author follows to insure the simulations created achieve their goals. It is a step-by-step cycle which the author must follow. It is also an iterative process as an author is unlikely to create an effective online soft skill simulation in their first attempt. They must review and edit the different aspects of their simulation, constantly making adjustments until the simulation meets their needs. Part of the authoring process does include the composition of the dialogue but also includes other steps involved in creating the simulations. For example, describing the scenario or outlining the

situation in which the simulation takes place. Without these steps the author has no abstract view of what it is they are attempting to achieve with the simulations.

The VISIO authoring tool allows the author to compose a dialogue but does not provide any support for the authoring process. Captivate 5 does provide help functionality but this is more of a user manual than a formal authoring process. The authoring process used in SimWriter is conceptually better than most other authoring tools. It divides authoring into three intuitive phases that are easy to understand and follow. However, the phases themselves do not provide enough detail in the support they provide to the author. Experience Builder is one of the few authoring tools that provides the required detail in a manner that is effective and efficient. In fact Experience Builders central approach to creating simulations is based upon the author following this process. This insures the author has a complete understanding of the simulations they are creating and so is more likely to create simulations which achieve their goals.

Pedagogical Framework

The goal of soft skill simulations is to transfer knowledge to the learner. To accomplish this, the simulations must be educationally sound. This can only be ensured by putting in place a pedagogical framework to support the author. A pedagogical framework consists of support structures, standard operating procedures and pedagogical best practice which aid the author while composing their simulations. They are placed across the entire authoring process. Without such a framework, the author risks creating simulations that contain no educational merit.

Experience Builder and SimWriter are the only authoring tools of the four surveyed to include any kind of a pedagogical framework. Experience Builders authoring process allows the author to create and populate learning objectives in its initial step. SimWriter allows the author to create learning points/objectives as they compose their dialogue. When creating a new course or syllabus it is normal practice to identify and describe learning objectives. This should also be the case if creating new educational simulations. A useful addition to simply describing the learning objectives is associating each of the learning objectives with dialogue components as SimWriter allows. This ensures that all learning objectives are addressed and that all the content developed by the author serves an educational purpose. SimWriter

however does not explicitly highlight enough the importance in creating learning outcomes in its authoring process.

Visual Description of the Simulation

The visual description refers to the presentation of the delivered simulations. For example, this would include the location of the hyperlinks, background colour, size of video window, etc. Two of the authoring tools examined, Captivate 5 and SimWriter, place a very high priority on the author's role in detailing the visual description. Captivate 5's approach is to combine the visual description of simulations with the model upon which the simulation will operate. This '*page built*' approach, so called as the author composes each of the simulations pages individually, results in the content and visual description being blended together. Without these being separate concerns there is less opportunity for reuse and any editing of the visual description that may occur would result in each page being edited individually. SimWriter employs an approach that separates the visual description (design) and content. Design is completed in the final phase of authoring and can be rather slow and time consuming. In contrast is the VISION authoring tool which does not allow the author to influence the visual description whatsoever. While the author is not distracted from developing the content of the simulation, they have no control over the visual description of the simulation. The optimal approach may be provided by Experience Builder which separates the dialogue model from the visual description but still allows the author to choose a "*skin*".

Scalability

In order to create very large and complex simulations a soft skill composition tool must allow the author to quickly and easily create the components of the dialogue, add them to the project and indicate where the hyperlinks are to occur. The composition tool also needs to provide a view of the entire dialogue which includes a visual representation of the links that exist between the different dialogue components.

The only authoring tool that accommodates these specifications is the VISION authoring tool. While the other authoring tools surveyed are technically capable of creating large complex simulations the logistics in doing so would make it a very difficult task. Scalability is closely

connected with dialogue representation. Without a visually accessible dialogue, creating a large complex simulation is very difficult. This is due to the cognitive overload that would be suffered by the author. For example, Experience Builder, which uses text to display the dialogue and has no graphical representation of the hyperlinks that occur, would not easily allow large simulations to be composed.

Navigation

Although Captivate 5 is not suited to authoring large complex simulations due to the time needed to create the dialogue components and connect them, it does offer some novel approaches in navigation. Within its Branching display the entire simulation is viewable by the author. One of the navigational aids offered to the author is a map of the simulation model. Within this map is an orange box which indicates the section of the model being displayed. As the author moves position in the display the orange box within the map moves accordingly.

Another interesting feature provided by Captivate 5 is the grouping function. This allows the author to replace several slides with a single slide which represents that group. The group of slides can still be accessed by rolling the mouse over the group slide to reveal the slides that are hidden. By reducing the number of slides visible, Captivate 5 reduces the complexity of the data that the author needs to manage. While the other authoring tools surveyed do offer some navigational aids such as zoom functionality, Captivate 5 is the most innovative.

Validation

The models designed and composed by authors to be used to operate soft skill simulations can become very large and complex. A great deal of burden is placed on authors to correctly compose simulation models; this includes creating and connecting components precisely within models and populating the components with intelligible data. Although model validation is highly thought of within the domain of simulations, often assigned its own track within simulation conferences, very few simulation training authoring tools provide any support to authors in assisting them to create comprehensive system readable models. Of the composition tools surveyed only SimWriter provided any kind of validation functionality. The script operation it incorporates allows authors to view how their simulation might play

out when published. While this is a very useful function there is a great deal of scope to improve validation of models with soft skill simulation authoring tools.

Summary

Each of the four authoring tools surveyed offer a unique approach to composition or functionality that creates a more efficient or effective authoring experience. A summary of each authoring tools performance with respect to the requirements described can be viewed in Table 3.1, a shaded entry indicates that an authoring tool has successfully addressed the related requirement.

Table 3.1: Authoring Tools Surveyed

	VISION	Experience Builder	Captivate 5	SimWriter
Dialogue Representation	Graph based Intuitive Free manipulation	Text based Inadequate	Tree based Restricted interaction	Graph based Linearly restrictive
Authoring Process	Basic model construction	Detailed and iterative Central to composition tool	Basic model construction Display switching inefficient	Clearly defined Addresses different aspects of composition
Pedagogical Framework	None	Allows authors to identify Learning Outcomes	None	Allows authors to identify Learning Outcomes
Visual Description	None	Authors select from canned front-end components	Over elaborate Requires a lot of resources	Final phase of design Slow and time consuming
Scalability	Allows large and complex models to be composed	Not feasible with text representation	Restricted due to cumbersome authoring process and representation	Restricted graph input creates linear model (not suitable)
Navigation	Only zoom functionality, not sufficient for graph representation	None	Incorporates an interactive map display	None
Validation	None	None	None	Author can script the simulation at any point to validate its content

It should be noted that although Captivate 5 allows the authoring of soft skill simulations it can also be used to author procedural simulations. By accommodating the composition of both procedural and soft skill simulations its competence in authoring soft skill simulations

suffers. This is particularly true of the dialogue representation as the dialogue model which is not separated from the visual description of the simulation. A focus on the visual description of the simulation is more appropriate to authoring procedural simulations and does not work well for authoring soft skill simulations.

3.2 Survey of Adaptive Courseware Authoring Tools

This section describes two authoring tools that are used to compose adaptive eLearning courseware. There are many different composition tools that can be used to author such adaptive systems. These two were selected as they offer approaches to applying or representing adaptivity that are particularly relevant for soft skill authoring. These two also offer a good account of the authoring tools that are available and highlight both the strengths and weaknesses that are typical of such systems. The ACCT (Dagger, Wade et al. 2004) authoring tool was chosen for the survey as it allows non-technical domain experts to compose adaptive systems. It is accessible, user friendly and allows authors to rapidly develop their courseware. MOT (Cristea and de Mooij 2003) was examined as part of the survey as it permits the author to not only compose their content but also allows them complete control of personalisation in the system.

In order to survey the authoring tools they are examined under carefully selected criteria. These criteria include an overall description and evaluation of the systems. Other criteria used to examine the composition tools include how adaptivity is applied to the courseware and how it is represented within the authoring tool. These criteria were used as part of the survey as they are vital to competent composition of adaptive systems and are relevant to successfully authoring adaptive soft skill simulations. This survey also describes the systems courseware representation and authoring process as they are required to fully understand how adaptivity is applied and represented. These concepts in themselves are not relevant to composing soft skill simulations but are necessary to provide a complete description of the authoring tools. Each of the criteria used to examine the composition tools are detailed in the following paragraphs.

Description: The authoring tools general description is detailed which includes its origins, overall goal and information regarding the courseware that is produced.

Authoring Process and Courseware Representation: AEH authoring tools typically incorporate an authoring process that an author follows in order to compose adaptive courses. These processes include steps for developing content, pedagogical strategies and applying/describing adaptivity. Courseware representation describes how content is visualised in AEH authoring tools. There have been many different approaches employed for courseware representation. Typically a course's content is decomposed into smaller more manageable components that are interconnected to form a hyperspace.

Adaptivity Application and Representation: This is the most important criterion that AEH authoring tools are examined under in this survey. It is desirable to separate adaptivity from content so adaptivity is not hard coded into a system. Using this multi-model (Conlan 2005) approach allows a separation of concerns meaning content and adaptivity can be easily edited and reused. AEH authoring tools typically allow authors to initially create content which is later personalised based on adaptive rules or techniques. This criterion examines how the AEH authoring tool allows authors to apply adaptivity and how it is graphically represented.

Discussion: The final section of each of the following reviews presents a discussion about each of the adaptive courseware authoring tools being surveyed. It details a critical account of the advantages and disadvantages of using the tool to author adaptive courseware.

3.2.1 MOT

My Online Teacher (MOT) (Cristea and de Mooij 2003) is an academic adaptive hypermedia composition tool developed in the Technical University of Eindhoven TU/e. It is a web based authoring system used for on-line adaptive course production. MOT is a "*general authoring tool*" (MOT 2010) so can be used to compose courses which are compatible with different adaptive engines such as that used in the AHA! system (De Bra, Aerts et al. 2003). To achieve this MOT exports to a generic format called CAF (Common Adaptation Format). MOT is based on two frameworks: an adaptive authoring framework for adaptive hypermedia systems and a framework to integrate in the future with LAG (Layers of Adaptation Granularity) which is a layered view on adaptation functionality. Lessons are composed using a concept map and the system is implemented using AHAM (Adaptive Hypermedia Application Model) (De Bra, Houben et al. 1999), LAOS (Layered WWW AHS Authoring Model and their corresponding Algebraic Operators) (Cristea and de Mooij 2003a) and LAG (Cristea and Calvi 2003) models.

Authoring Process and Courseware Representation

The MOT authoring process is based on the LAOS model for composing adaptive systems. The author initially creates a domain model which consists of all the concepts that are available in the course. In order to relate these concepts the author creates a goals and constraints model. This is overlaid with dimensions that are used to describe the characteristics of the learner model. To describe the adaptivity that is to occur in the system the author must define rules that work with the concepts defined and learner dimensions that are described. Finally, the author defines the presentation format that is to be used in the course.

All content in the composition tool is described by the author in the concept map. The concept map is a hierarchical list that encapsulates and describes all conceptual information that is contained in the course being developed, as seen in the left panel in Figure 3.5. The simple list display allows an author to add and name concepts and sub-concepts. Each concept allows the author to populate associated attributes such as the concepts title, text and associated keywords, as seen in the right panel of Figure 3.5. Once the concept map is completed courses (or lessons) can be extracted based on the goal and constraints that an author defines. By creating different goals and constraint models, or modifying existing ones, an author can create many different lessons from a single concept map. An author can further improve the effectiveness of this process by adding weights to the concept map that they have created. MOT's flexible framework is critical for applying adaptivity. It should be noted that while multiple courses can be created by manually editing the goals and constraint models, further steps are required for the system to become adaptive.

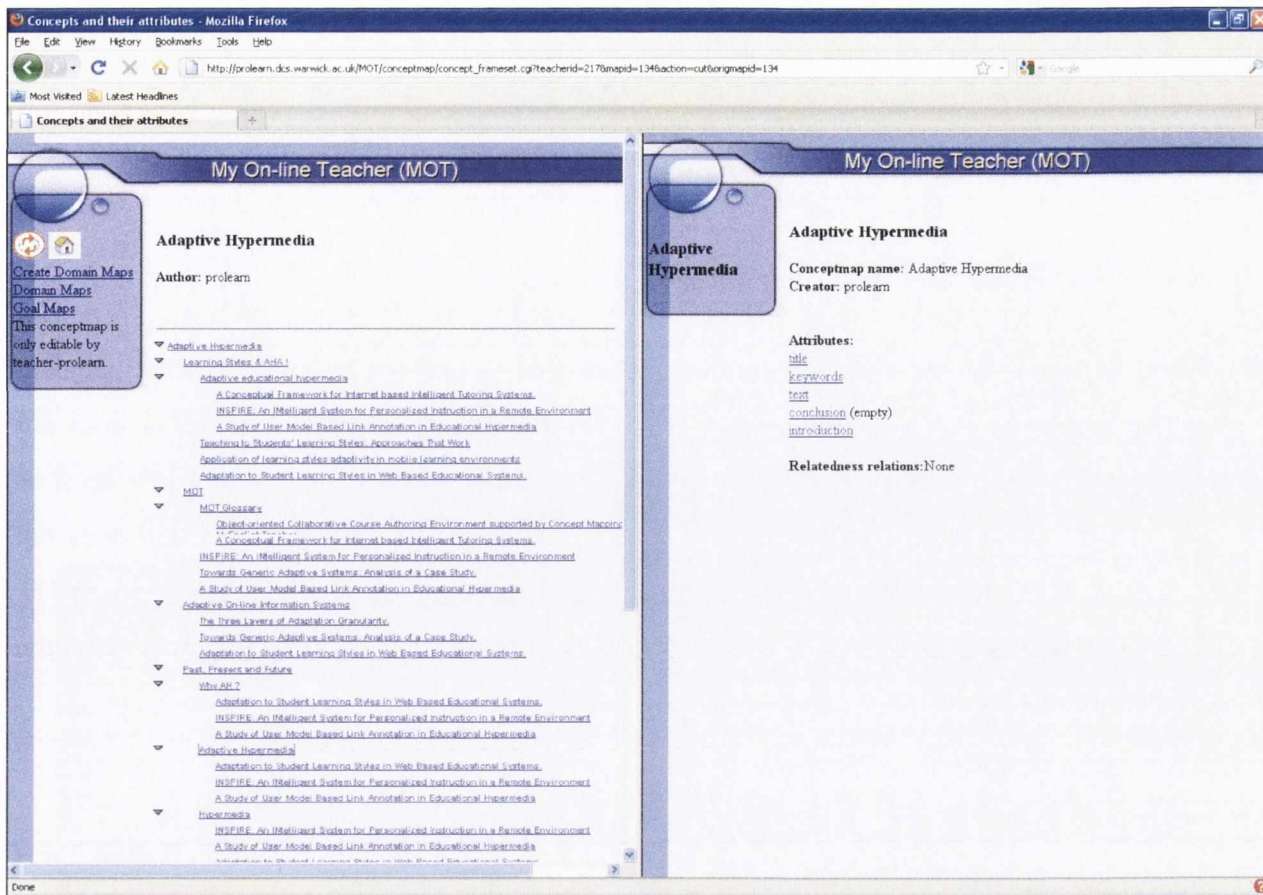


Figure 3.5: My Online Teacher Authoring Tool

Adaptivity Application and Representation

The goals and constraints models combined with weighting allow lessons to be generated from a single concept map. However, in order to adapt the context and structure of a course based on the dimensions of a learner model an author must describe suitable rules. Adaptivity rules are based on the LAG architecture. They are condition-action based and specify the property of a concept or learner model along with logical operators to be tested against an assigned value. Adaptation is calculated using the composed rules and numerical values that are assigned to concept or learner models.

Discussion

MOT employs an effective approach to representing and authoring course content. The hierarchical format of concepts is intuitive and would be easily understood by subject matter experts. Relating different concepts is also an idea that authors would be comfortable with. MOT's most innovative feature is that it allows the authors to create the rules which will adapt and personalise the course. The adaptivity that occurs is also powerful and effective.

MOT however has not been used commercially and is only really accessible by those with a technical background. The difficulty with MOT comes with the conception and implementation of the personalisation. While the manual refinement of the lessons is easy to follow the application of adaptivity is rather technical. An author needs to be familiar with adaptive hypermedia principals to effectively and efficiently apply personalisation. The use of logic rules combined with weighting of relatedness is also not an approach that is easily accessible for non-technical trainers and subject matter experts. Furthermore, MOT does not support any global strategies of personalisation with all adaptivity applied to a low level of granularity. Another criticism of MOT is that is basically a modelling tool which does not directly produce an educational course. Instead, the LAG models that are developed in MOT need to be placed into AHA (De Bra, Aerts et al. 2003) along with other models that the author must write in order to produce an adaptive course.

3.2.2 ACCT

The ACCT (Dagger, Wade et al. 2004), Adaptive Course Construction Toolkit, was developed in Trinity College Dublin and is another academic adaptive hypermedia authoring tool. The ACCT was designed for a non-technical course developer to create adaptive and non-adaptive activity-oriented courses based on sound pedagogical strategies. It incorporates a graph representation of the author's concept space and provides a real-time course test and evaluation environment.

Authoring Process and Courseware Representation

The ACCT authoring tool provides a clear and concise adaptive course construction approach (authoring process). This process supports pedagogical and instructional theories while also employing adaptive techniques. The process is iterative and includes all the steps an author must address to create an adaptive educational hypermedia course. The author initially describes the overall goal of the course along with specific learning outcomes that are to be achieved. The author then defines the learning methods and activity sequences that are to be used. The content of the course is then composed as part of the subject domain. The author can then select which content and services are to be used for each lesson. The final two steps of the authoring process allow the author to assess and evaluate the course before beginning their next iteration of the process until the course is complete.

The ACCT contains a number of key design elements including narrative structures, activities, subject matter concept space, narrative attributes, learning resources and personalised eLearning designs. An author must consider each of these models when composing an adaptive eLearning course. Content of the courses are described in the subject matter concept space, as seen in main panel in Figure 3.6. Concepts (also listed to the left of the composition tool) are represented in the authoring tool as nodes in a graph. Relationships, such as ‘contains’ and ‘prerequisite’ are represented as directed edges that link different nodes. Components can be added to the subject matter space using the graph tools bar found at the top of the composition tool.

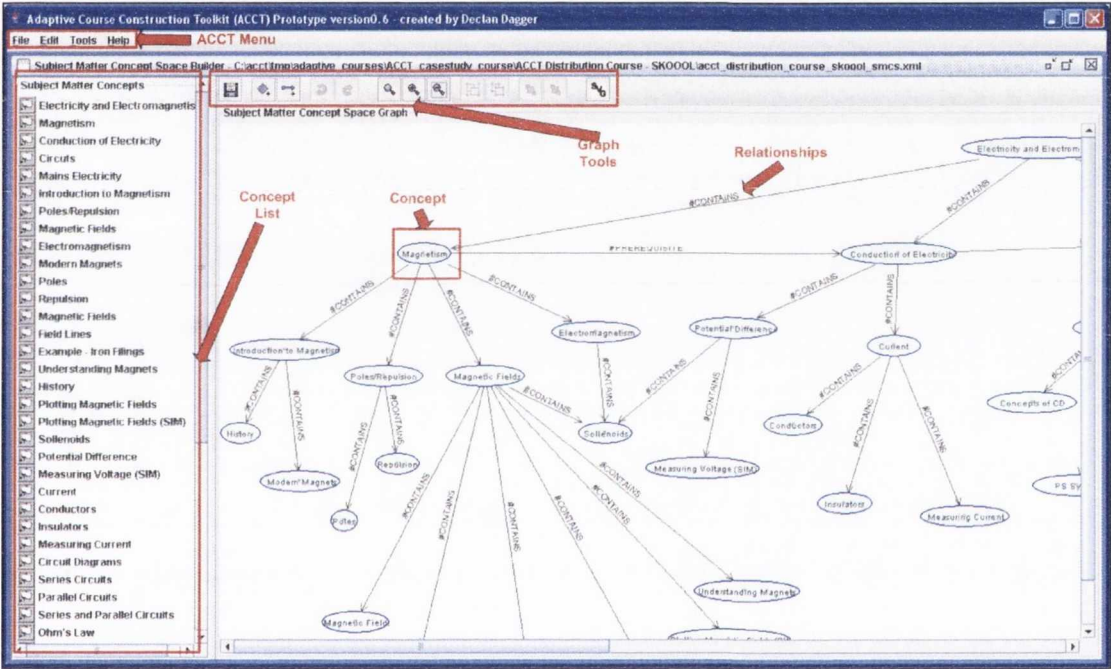


Figure 3.6: ACCT Authoring Tool

Adaptivity Application and Representation

The narrative attributes are used to describe a specific axis of personalisation (learner dimensions) along with its associated adaptive techniques. Adaptive techniques are based on operations that select appropriate content and are related to a number of different conditions. The personalised eLearning design model generated by the authoring tool is the final model that combines all other models and key elements including the narrative.

Discussion

The ACCT strongest feature is the use of a graphical interface which allows the author to easily and quickly develop the course concept space. The author is provided with clear and concise space for composing their course which is easily accessed by non-technical subject matter experts. The ACCT also provides a course developer with such tools as a custom narrative builder, content package assembler and learning resource repository. These features create a tool that allows the author to develop an entire course from beginning to end. ACCT was also successfully trialled and evaluated with secondary school teachers and other domain experts.

The ACCTs weakest feature is that it does not allow the author to create their own rules of adaptivity; instead the personalisation is predefined and fixed. There is also a poor representation of process flow within the concept space. The concept space is also rather lightweight and lacks the robustness of a true domain model.

3.2.3 Adaptive Courseware Authoring Tool Critique

MOT and the ACCT authoring tool are typical of adaptive educational hypermedia composition tools. They reflect both the positive and negative features that can be found in most authoring tools of this nature. AEH authoring tools fall in one of two categories; they either allow the authors to fully control adaptivity but do so in a manner that is not easily accessible (MOT); or they allow adaptive courses to be relatively easily authored but do not permit authors to control the adaptivity (ACCT). The difficulty with employing adaptivity in an authoring tool is that it is challenging to describe, apply and represent. Simply removing the ability to manipulate adaptivity, as the ACCT does, is not the answer. An author needs to be capable of managing and controlling all dimensions of the course including adaptivity.

Course designers need to be given control of adaptivity within the composition tool but this does present some difficult challenges. The principals of adaptivity in themselves are rather complex for domain experts (teachers/trainers) without a background in adaptivity to comprehend. Explaining adaptive navigation or adaptive presentation is not a straightforward task. Applying adaptivity to a course is also a rather problematic activity. The functions involved must be explained to the author as well as their affect on the course. MOT, for example, requires the author to compose formal rules that are not particularly intuitive for those without a technical background. Other authoring tools, such as the GRAPPLE

authoring tool (GRAPPLE 2011), employ a simple programming language which again requires technical training.

Representation of adaptivity within AEH authoring tools is also very limited. MOT, for example, does not use any graphical representation of adaptivity which is instead completely contained within the rules that an author defines. Other systems allow the author to control adaptivity using separate models to the content. This again is a rather technical approach and also requires considerable additional effort on behalf of the author. These approaches to representation also do not allow the author to view how the adaptivity will work with the content they have composed. While it is desirable to separate adaptivity and content it must be implemented in a manner that allows an author to view how the two models will work together so both do not become completely abstracted from one another. Effective adaptivity representation is very reliant on how content is represented. In order to represent adaptivity in a composition tool in a satisfactory manner content representation must also be considered.

3.3 Summary

Chapter three presented the state of the art survey associated with this research. The chapter was split into two major sections; 3.1 and 3.2. The first section, 3.1, described soft skill simulation authoring tools; these included the VISION authoring tool, Experience Builder, Captivate 5 and SimWriter. Each of these systems was examined under the criteria of their Description, Dialogue Representation and User Interface, Authoring Process and overall Evaluation. The final part of the first main section also presented a critique of all the soft skill simulation authoring tools.

The second major section, 3.2, described adaptive courseware authoring tools; these included MOT and the ACCT. They were both examined under the criteria of their Description, Authoring Process and Courseware Representation, Adaptivity Application and Representation and finally their overall Evaluation. The final part of the second main section also presented a critique of all the adaptive courseware authoring tools.

The results of this survey highlight the important aspects of design that are required by a tool which will allow non-technical subject matter experts to compose adaptive soft skill simulations; these include:

- **Dialogue Representation:** such is the importance of how the soft skill simulation models are represented it was used as a criterion under which the soft skill authoring tools were described and evaluated. The VISIO authoring tool exemplified the best approach to this requirement.
- **Adaptivity Application and Representation:** the manner in which the adaptive courseware authoring tools allows adaptivity to be applied is also very important. While neither of the adaptivity courseware tools examined incorporated adaptivity effectively they both employed concepts that, if combined, could be very effective.
- **Authoring Process:** nearly all of the authoring tools surveyed employed some kind of an authoring process which highlights its importance. However, these were for the most part inadequate or far too implicit to be effective. The authoring tools to employ the most effective authoring processes were Experience Builder, SimWriter and the ACCT.
- **Pedagogical Framework:** only two of the tools surveyed employed any kind of pedagogical framework, although the need for such in eLearning design is well documented. The ACCT and Experience Builder allowed authors to describe their simulation with respect to pedagogy although the design was rather poor.
- **Navigational Aids:** while the graph representation is very effective it does require navigational aids to assist the author. This becomes quite clear with the ACCT as authors can easily become lost or disorientated when their models become large and complex. SimWriter was the only tool surveyed to address the concept of navigational aids.
- **Model Verification:** only one of the systems evaluated, SimWriter, employs any kind of model verification. The importance of this requirement becomes evident when using any of the authoring tools, particularly when employing a graph representation such as that used in the ACCT. It becomes very difficult for an author to visually inspect such large complex models if examining each dialogue component individually. Functionality like SimWriter script would allow authors to complete such a task quickly and much more effectively and efficiently.

The only aspect previously mentioned in this chapter not thought to be of value is the incorporation of a simulation visual description (front end of the simulation). Employing such a feature into an authoring tool would distract from the more important model development.

The effort in describing such an aspect of the simulation would not be proportional to its contribution to the end simulation as soft skill simulations typically employ video or sound files. The video/sound components are central to the simulation and the front end becomes rather inconsequential.

The following chapter describes in detail the requirements of an authoring tool that will allow non-technical subject matter experts to compose adaptive soft skill simulations.

4 Design

This chapter describes the design of the ACTSim authoring tool. It discusses the process of design in three sections. The first section, 4.1, presents the key requirements of ACTSim as determined by an analysis of the state of the art survey in the previous chapter. The second section, 4.2, presents the key design elements of ACTSim as determined by the requirements; the design elements show how the requirements are realised within this research. The design elements are also contrasted to current approaches in the Related Work section contained in the Evaluation chapter of this thesis. The third section, 4.3, briefly outlines when the different design elements were introduced into the system in relation to the two phases of development, as described in chapter one. The final section of this chapter, 4.4, presents a summary of the chapter.

The candidate was solely responsible for the design of the ACTSim composition tool described in this chapter. The candidate was also responsible for implementing and building the ACTSim system (as detailed in chapter five) and for conducting all of the evaluations described in this thesis (as detailed in chapter six).

4.1 Key Requirements

To create an effective authoring tool that easily allows non-technical domain experts to compose adaptive soft skill simulations there are a number of requirements that must be addressed. As derived at the end of chapter 3 and presented in previous work (Gaffney, Dagger et al. 2008) these requirements were identified as:

- Dialogue Representation
- Adaptivity Application and Representation
 - Adaptive Dialogue Models
 - Adaptive Triggers
 - Adaptive Dimensions
- Authoring Process
- Pedagogical Framework
- Navigational Aids
- Model Verification

The following sections explore each of these requirements in detail.

4.1.1 Dialogue Representation

Traditional eLearning composition tools are not expressive enough to represent the models upon which simulations operate. Educational simulations require a very new way of thinking about content (Aldrich 2004) which must be accommodated in the tools used for their development. Simulation authoring tools must represent content in a manner which allows the author to capture the intricate relationships that exist within the simulation models. Furthermore, authoring tools need to refine model representation to suit the particular type of simulations that they are being used to develop. Different types of educational simulations require different types of model representation in their composition.

The models upon which soft skill simulations operate are based on the dialogue that occurs within the simulation. Soft skill simulation authoring tools must therefore represent the dialogue in an intuitive and user friendly manner. The following requirements were identified as being essential for successful model representation:

- The model upon which the simulations operate should be based on the dialogue that occurs within the simulation.
- The dialogue should be decomposed into smaller components which are connected with directed arrows which indicate the possible paths available to the learner.
- As the models can become large and intricate only one type of node should be used to express the dialogue to reduce its complexity.
- Each component of dialogue should represent a single statement and response along with metadata describing that statement and response.
- The dialogue model should be represented as a graph of nodes and edges as seen in many soft skill simulation authoring tools and dialogue management systems (Luz 2000).
- Learning outcomes and adaptivity should not be expressed as connected nodes within the main dialogue model as this would increase the models complexity. Although a feature should be incorporated in the composition tool that allows the author to view learning outcomes and adaptivity associated with the model.

4.1.2 Adaptivity Application and Representation

Adaptivity within eLearning authoring tools is typically only accessible to those with a background in adaptive hypermedia and personalisation. Adaptivity within the next generation of systems needs to be designed in such a way that is accessible to domain experts who have no previous experience with personalisation. Adaptivity must be intuitive and easily explainable to authors. Personalisation that is complex and not easily understood may be very effective but will not be accessible to authors and so will be redundant within an authoring tool.

The principals employed in adaptivity should be simple enough that they can be quickly and easily explained to authors but should also be effective at personalising a learner's experience. The challenge is designing an authoring tool that finds a balance between effectiveness and accessibility.

Most traditional adaptive courseware composition tools are unsuccessful at satisfactorily representing adaptivity. Content is typically represented and presented as text within these authoring tools and adaptivity is based on tagging this content. Therefore it becomes difficult for an author to view where and how adaptivity might occur within the adaptive course as they can only view small segments of the course at any one time. Traditional adaptive courseware composition tools also typically require the author to compose complex rules or commands based on process languages. In order for adaptivity to become accessible to non-technical subject matter experts authoring tools need to incorporate a graphical approach to expressing personalisation.

This research has identified two approaches to adaptivity within the domain of soft skill simulations and has defined adaptive dimensions specific to soft skill simulations. The first approach to personalisation is manipulating the dialogue model so it becomes relevant to each individual learner; **Adapting Dialogue Models**. Dialogue is adapted so parts are included or excluded in a learner's simulation. This is accomplished by pruning the model so only particular sections or sub-dialogues become available to the learner. This is an approach based on adaptive navigation principals as seen in traditional courseware adaptive hypermedia systems. It is simple, effective and can be easily explained to non-technical

authors. In order to accommodate this type of adaptivity within the composition tool the following requirements were identified:

- The author should be allowed to tag dialogue nodes within the dialogue model for personalisation with values which will describe the nodes and indicate if they are relevant to a learner based on the learner's user model.
- Tagging should be based on properties (**Adaptive Dimensions**) that are appropriate to soft skill simulations. These include, but are not limited to, the role of learner, learning outcomes the learner is to achieve, related subjects that describe the dialogue which are relevant to the learner and categories of interaction that available to the learner in the simulation.
- The author should not be required to create any rules or write any code for adaptivity to occur. Personalisation of the model should be based on the types of tagging that the author has implemented.
- The model should generally not present a visual indication of tagging to the author to reduce the complexity of the model. However the authoring tool should incorporate a feature that allows the author to view tagging within the model which they can turn on and off.

The second approach to personalisation identified within this research was to adapt the learner's learning experience based on educational principals that are triggered within the simulation (**Adaptive Triggers**). These triggers include assessment, feedback and reflection (Quinton 2006). In a traditional approach triggers would occur at the same time within a simulation for each learner and deliver identical content. The approach employed in this research personalises the triggers so they only fire within a simulation if relevant to the learner. Furthermore, the content delivered in these triggers can be personalised to the learner based on their route through the simulation or their user model. In order to accommodate this type of personalisation within an authoring tool the following requirements were identified:

- Triggers should be represented as nodes in the dialogue model that can only be placed *within* the dialogue nodes to indicate their occurrence in the simulation. This approach insures triggers do not increase the complexity of the dialogue model.

- It should be possible to relate triggers to a learner based on dimensions (**Adaptive Dimensions**) that are appropriate to the soft skill simulations and the learner. For example the role of the learner or the learning outcomes they have already achieved within the simulation.

4.1.3 Authoring Process

ELearning authoring tools, particularly those used to compose educational simulations, require a clear and concise authoring process. The authoring process is an outline or blueprint that an author follows while developing eLearning courseware. The composition of adaptive soft skill simulations must include several aspects in its authoring process which cover both simulation development and adaptivity. A competent authoring process should include the following:

- The authoring process should be an iterative step-by-step cycle which focuses on composing the dialogue models upon which the simulations operate. All other steps within the process should support the development of these models.
- The authoring process should include a pedagogical framework which aligns the authoring process with educational strategies and principals.
- There should be no specific limit on the number of Dialogue Elements that can be created within a simulation. Typical implementations could contain between 100 to 200 dialogue nodes. The author should be allowed to create and connect these nodes rapidly within the authoring process.
- The authoring process should include a strategy for developing and connecting dialogue nodes within a graph interface.

4.1.4 Pedagogical Framework

As mentioned in the previous section a composition tool and authoring should include a pedagogical framework. The author defines the pedagogical framework as being a set of broad principles that guides the quality design of the soft skill simulation. The framework should be implemented at two levels to adequately support an author and fully describe a simulation's pedagogy. The first level of the framework is the pedagogical approach. This is an abstract description of a simulation's pedagogy. It details the overall educational approach that is to be used within the simulation. The pedagogical approach is closely related to the style of dialogue that a simulation is to accommodate. For example, the dialogue may be

highly interactive or have a more informative approach; it may be supportive or confrontational. Different types of soft skill simulations will require different dialogue approaches depending on their overall objective.

The second level of the pedagogical framework allows authors to define the learning outcomes of a simulation. This is a much more concrete approach to defining the pedagogy than the pedagogical approach. Learning outcomes describe specific educational goals that are to be achieved within the simulation. Identifying learning outcomes encourages authors to have specific pedagogical objectives which assist in the design of the simulation. Although the system does not interpret the learning outcomes they support good practice in educational design.

This should be supported further by allowing an author to explicitly relate the dialogue model with learning outcomes that they define. This assists the author in composing a model that is educationally sound while also acting as documentation of the developed model. The concept of learning outcomes should be familiar to most domain experts. However, authors may not have a great deal of experience formally defining learning outcomes. Learning outcomes should be designed to be easily definable and simple to understand. An authoring tool should address the requirements of a pedagogical framework by including the following:

- The authoring tool should allow an author to define the pedagogical approach of the simulation. This is an abstract view of the educational simulation that captures the style of learning that runs throughout the dialogue. The pedagogical approach should be described using an open text format which can be edited and updated at any time during the authoring process.
- An author should be capable of creating learning outcomes which describe the pedagogical goals of a simulation. Learning outcomes should consist of a '*learning verb*' and a '*learning noun*'. This approach insures that Learning outcomes remain short, concise and descriptive.
- An author should be allowed to associate one or more learning outcome that they have created with the model they have created.

4.1.5 Navigational Aids

While the graph based approach employed to represent the dialogue within the simulation is intuitive it can become difficult to manage when it becomes very large. An authoring tool needs to address this by including aids to assist the author. Some of these navigational aids are incorporated in the systems surveyed in the state of art review such as the map and zoom functionality. Other functionality was identified during the development and evaluation of this researches artefact. To assist an author in navigating their dialogue model the following functionality should be included:

- **Map:** The author should be allowed to navigate their dialogue model using a map. The map should be positioned so as not to interfere with the view of their dialogue model. It should display the entire dialogue model as well as highlighting the area being currently viewed in the main viewing window. The highlighted area of the map should also be interactive allowing the author to change the view in the main dialogue window by dragging the highlighted area within the map.
- **Zoom:** The author should be allowed to zoom in and out of the main dialogue window to view the dialogue model at different sizes. The function should be accessible by selecting in the main viewing window where the zoom is to be applied. The zoom function should also be accessible by selecting predetermined sizes based on percentage displays; for example, viewing the dialogue model at 50%, 75%, etc. of its normal size.
- **Search:** An author should be allowed to execute a search of the dialogue model to locate particular dialogue components. The search function should be String based and allow the author to search the main properties of a dialogue component such as its statement and response. Dialogue components should be highlighted in the model if they match the search criteria entered by an author.
- **List:** An author should be allowed to view all dialogue components in a separate window. The window should display a list of all the dialogue components so they can be located quickly. The list format should display the statement of each dialogue component that has been created. The main viewing window should navigate to and highlight the corresponding dialogue component when it is selected by the author in the list view window.

- **Arrange:** The author should be allowed to select a function that rearranges the layout of the dialogue model so it becomes easier to navigate. All connections should remain intact and only the manner in which the model is displayed should be altered.

4.1.6 Model Verification

The models that an author can produce can become very large and complex. Authors can compose in a space that is non-restrictive where dialogue component can be placed freely and the models can be constructed at the discretion of the author. This approach to authoring allows an author to quickly and easily build their models but can produce models that are intricate and difficult to follow. To aid authors while constructing models that are appropriate for soft skills it is necessary for the authoring tool to include a suite of functions that assist the author in verifying the models are correct. Only one of the functions presented below, Script, was described in the state of the art survey, the other functions were developed during the design and evaluation of this research's artefact. To assist the author in verifying and validating their dialogue models the following functionality was identified:

- **Script:** An author should be allowed to view sequentially sections of the dialogue that occurs in the model that they have created. While the graph approach to representation is very good at capturing and representing the entire dialogue that can occur in the simulation the sequencing of a dialogue is not immediately obvious. The author needs to be allowed to view the content contained in the paths of the dialogue to insure that they are appropriate and meaningful.
- **Highlight:** An author should be allowed to highlight dialogue component that have been tagged with particular adaptive properties. This allows the author to view how adaptivity of the dialogue model will occur and verify that every adaptive property that has been created has been applied at some place in the model. This function can also be used to view the relationship between Learning Outcomes and a dialogue model when examining the educational validity of a simulation.
- **Validate:** An author should be allowed to validate that a model has been created and completed correctly. In particular validate should check that all essential data of each dialogue component has been entered by the author; i.e. the statement and response associated with each dialogue component.

4.2 Key Design Elements

This section presents the key design elements related to this research; it describes how the key requirements previously described are realised. The first subsection presents an overview and description of the authoring tool. This is followed by a subsection that describes dialogue representation. The next subsection details the adaptivity application and representation. The final three subsections present the authoring process, pedagogical framework, navigational aids and model verification respectively.

4.2.1 Authoring Tool Overview and Description

The objective of the authoring tool is to allow non-technical subject matter experts to compose adaptive soft skill simulations. The authoring tool was designed to be an application based system and was named ACTSim (Adaptive Composition Tool for Simulations). The ACTSim composition tool was developed in Trinity College Dublin in cooperation with the Department of Psychiatry as part of the ADAPT project (ADAPT 2010). The approach employed in development of ACTSim has been action based, specifically cooperative inquiry (Heron 1996). This approach allows the designers and users to work closely together and emphasises that all active participants are involved in design decisions. The idea is that the designer works *'with'* rather than *'on'* users. This concept is very suited to developing applications which are user centric.

ACTSim was developed using an iterative rapid prototyping approach supported by cooperative inquiry. This began with a detailed examination of the key requirements described in the previous section and other publications (Gaffney, Dagger et al. 2008). Each of the key elements of the ACTSim are presented in the following sections in relation to ACTSims key requirements.

4.2.2 Dialogue Representation

The dialogue that occurs within a simulation is decomposed into small components known as *'Dialogue Elements'*. The Dialogue Elements are represented graphically as nodes within the composition tool. They are primarily composed of a *'Statement'* and *'Response'* which occurs within the simulation, the Statement which can be chosen by the learner in the simulation and appropriate Response of the simulated person in the simulation. The Dialogue Elements are connected with directed edges which indicate the possible paths the learner can

take within the simulation; the edges are known as '*Connections*'. The connected Dialogue Elements form a graph known as a '*Dialogue Model*' which is built and contained in an area in the ACTSim authoring tool called the '*Dialogue Space*'; an example of such a model can be seen in Figure 4.1. Each of the nodes lettered A through I represent a Dialogue Element. The directed edges that connect them represent the paths available to the learner. Once the Dialogue Model has been completed and fully scripted video clips or sound files can be recorded and associated with the appropriate Dialogue Elements.

In the example viewed in Figure 4.1 a learner would begin at Dialogue Element A; this is the only Dialogue Element that has no Connections (directed edges/arrows) pointing at it so is the first node of the dialogue (a Dialogue Model may have several nodes that have no arrows pointing at it which would indicate several start points in the simulation). The learner in this example would be presented with one option from which to choose; the Statement associated with Dialogue Element A. Once the learner has selected this Statement (a comment, question, action or short piece of dialogue that is available in the simulation) they would be then presented with the Response associated with Dialogue Element A.

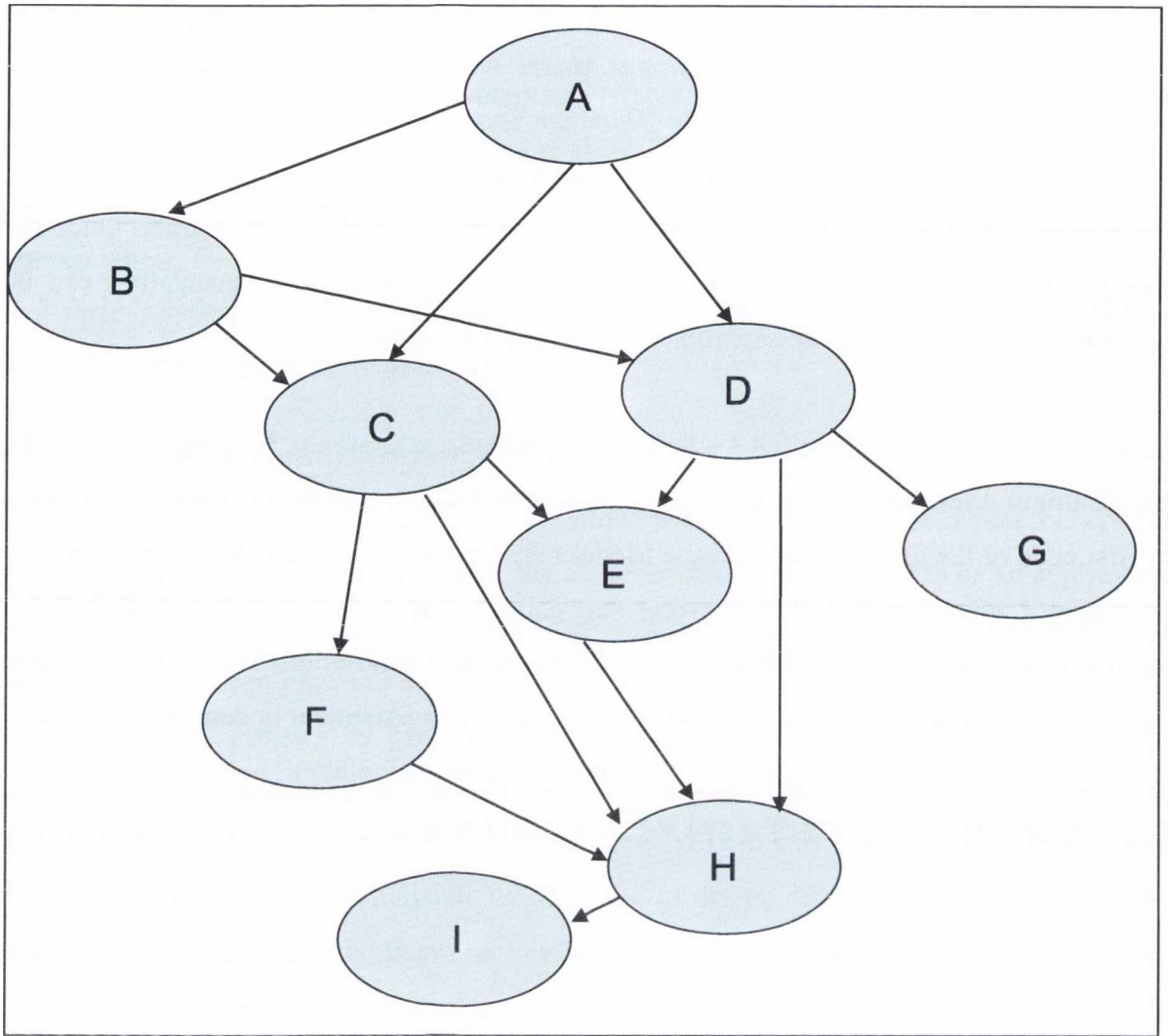


Figure 4.1: Example of Dialogue Representation

Once the Response of Dialogue Element A has been completed the learner will be presented with the paths that are available; in this case the Statements associated with Dialogue Elements B, C and D would be a learner's options. Once one of these options is selected by a learner they are presented with the associated Response of the selected Dialogue Element. When the Response has finished the learner is then presented with another set of options leading from the selected Dialogue Element.

The sequence of Statement selection, Response delivery and presentation of a new set of Statement options repeats until the learner arrives at a Dialogue Element that has no edges leading out to other Dialogue Elements. Such a leaf node indicates the end of the simulation; in this case G and I are end points.

Media File Association

The process of creating a soft skill simulation initially requires an author to create a text based model of the dialogue that will occur within a simulation. A dialogue is described with the use of Statement and Response properties that are contained within the Dialogue Elements. Once this model has been completed by an author media files, typically short video clips, are recorded based on the text stored in the Dialogue Model. Each media clip is then paired with their associated Dialogue Element within the Dialogue Model. To allow this to occur Dialogue Elements contain additional properties; a '*Statement Media File Location*'; and a '*Response Media File Location*'. These properties allow an author to indicate the file location of the each media file with their related Dialogue Element. A further additional property employed by Dialogue Elements is a '*Notes/Directions*' property. This allows an author to add directorial guidance or notes to each Dialogue Element which can assist the media file recoding process.

4.2.3 Adaptivity Application and Representation

ACTSim employs two approaches to composing personalised simulations: Adapting Dialogue Models and Adapting Triggers. The first approach is to adapt the Dialogue Model to individual learners through inclusion and exclusion. The second approach is to adapt the occurrence of educational triggers within the simulation based on assessment, feedback and reflection. Both of these approaches are presented in detail in the following sections.

Adapting Dialogue Models - Dialogue Inclusion and Exclusion

The first approach to personalisation allows an author to develop simulations which adapt the dialogue to the needs of the learner. This concept is similar to that of adapting the content of courses in traditional eLearning systems. The dialogue is adapted to the needs of the learner based on relevance and scope. Portions of the dialogue will not be available to a learner once it has been adapted. Personalisation may be as fine grained as one particular Dialogue Element being excluded or be more expansive to encompass entire sub-dialogues.

ACTSim incorporates an approach of tagging a dialogue model to achieve dialogue adaptivity. This is an intuitive and easily applied strategy which importantly does not require the author to compose complex rules or commands. Tagging is easily completed by an author in the graph representation of the dialogue model. Authors can tag Dialogue Elements individually or apply tagging to a selected set of Dialogue Elements. The composition tool

also allows the author to highlight sections of the model to view how and where adaptivity might occur.

Personalisation of a dialogue is based on carefully selected learner dimensions which are relevant to soft skills. Inclusion and exclusion of the Dialogue Model is executed based on the tag properties that have been associated with Dialogue Models. Dialogue Element tags matching properties of a learners dimension result in that Dialogue Element being included in the simulation. If the tag does not match with a learners dimension the Dialogue Element is excluded from the simulation. This strategy is applied to the entire Dialogue Model so only the components that are relevant to the learner will be included in the simulation.

Figure 4.2 presents a visualisation of a small scale example of how adaptivity might affect a Dialogue Model. Figure 4.2.i displays a Dialogue Model that has been created by an author. This Dialogue Model has been tagged with properties that are relevant to its associated authors. Figure 4.2.ii presents a visualisation of how the model might be adapted based on Learner One’s user profile. Dialogue Elements B, C and F have been tagged with properties that are not relevant to Learner One and so would not appear in Learner Ones personalised simulation. Figure 4.2.iii presents a visualisation of an adapted dialogue model for an alternative learner, Learner Two. Learner Two has a different user model to Learner One so the original Dialogue Model has been personalised with different results. Figure 4.2.iii indicates that Dialogue Elements E, G and I are not relevant to Learner Two so would not be included in their personalised simulation. This example demonstrates that even personalisation on a very small scale can create very different experiences for learners.

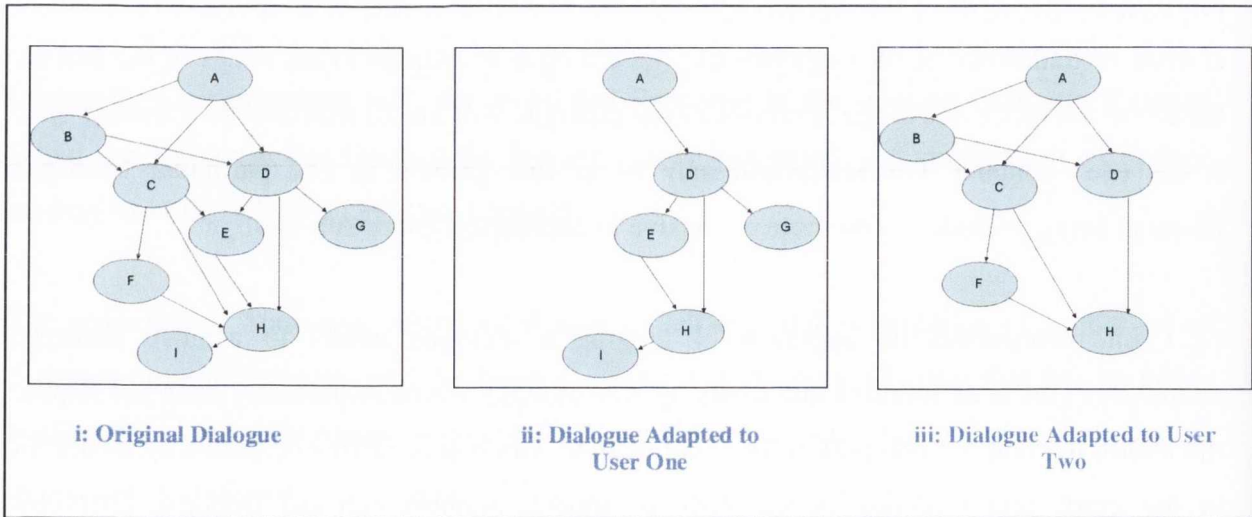


Figure 4.2: Example of Adapted Dialogue

There is however an obvious difficulty when excluding/including sections of a dialogue. Adaptation of this type could result in a simulation dialogue becoming disjointed and disconnected. The consistency and continued continuity of a dialogue is not addressed by the adaptive technique but is instead protected by the Dialogue Model authoring process; this process is described later in this chapter.

Adapting Triggers

The second approach to personalisation is based on adapting the educational principals of assessment, feedback and reflection that occur in a simulation. These are personalised triggered events, “*Triggers*”, within the simulation that are only fired if the dimensional properties in the learner’s user model are matched with tags assigned to the Trigger. The strategy employed to achieve this effect is similar to that of including and excluding the dialogue within the simulation. Triggers can be tagged by the author with properties based on appropriate adaptive dimensions. These are the same dimensions used for adapting the Dialogue Model. Triggers within a simulation will only fire if their tag matches the dimension property of a learner’s user model. If the tag does not match the learner’s dimension property the trigger will not fire. The content delivered to the learner through these triggers can be dynamically adapted to the learner. This is based upon several aspects of the simulation at run-time such as the route the learner has navigated through the dialogue or learning outcomes they have achieved.

Triggers are represented as nodes in the ACTSim composition tool which are placed on the Dialogue Elements to indicate their occurrence in the simulation. This required a redesign of the structure of nodes used to represent the Dialogue Elements. To accommodate Triggers being associated with Dialogue Elements a compartment was added to the base of the Dialogue Element node, as seen in the Dialogue Elements in Figure 4.3. Dialogue Element nodes were also redesigned from an elliptical shape to rounded rectangle. This allows adequate space for an author to easily place multiple Triggers in a single Dialogue Element node; an author may want to place several Triggers at a single point in a simulation. The circular nodes that can be seen in Dialogue Elements B, G, F and H in Figure 4.3 show how Triggers are represented in a Dialogue Model. Tagging of Triggers is executed in a similar manner to tagging Dialogue Elements although Triggers can only be tagged individually and not in groups.

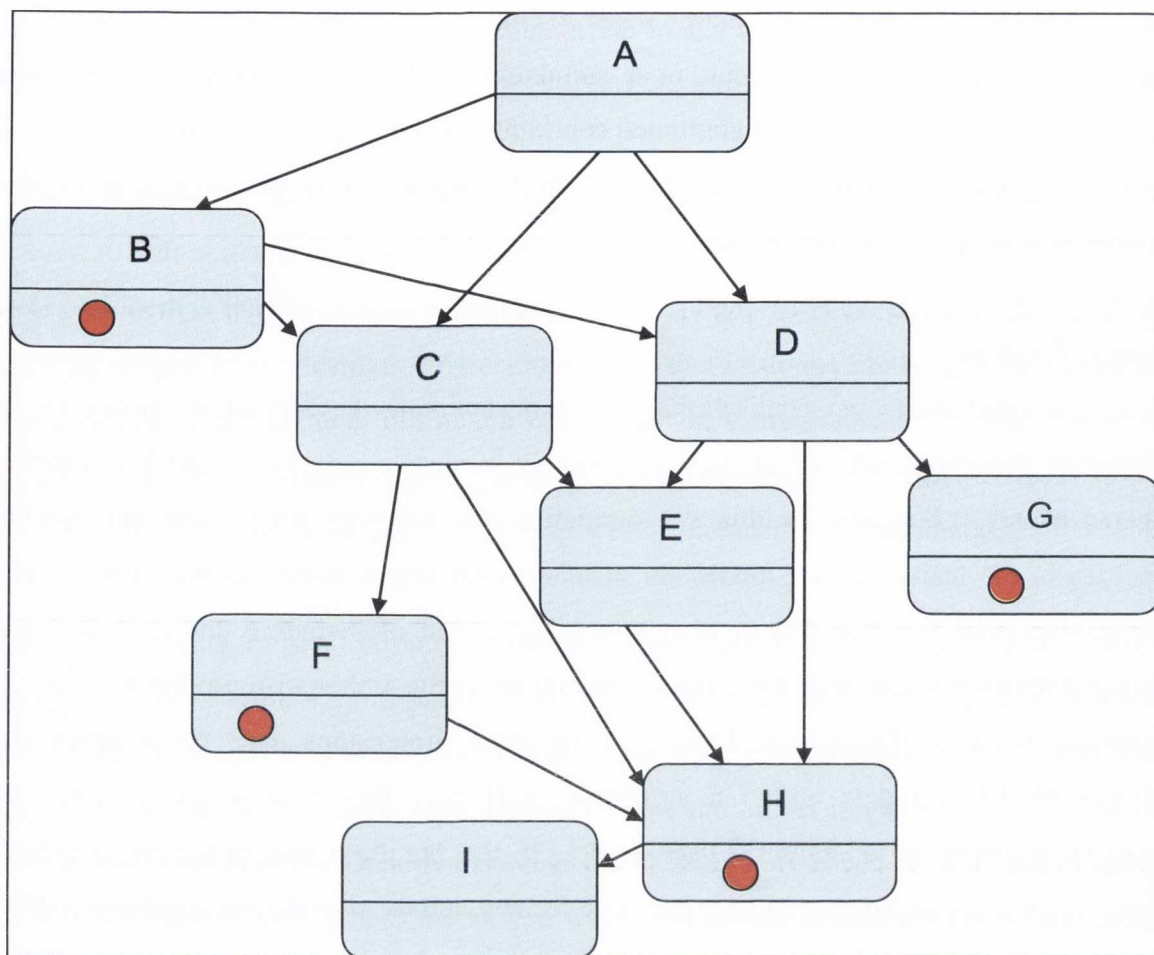


Figure 4.3: Example of Dialogue Representation with Triggers

Figure 4.4 displays an example of adaptivity similar to that seen in the previous example related to Figure 4.2. Figure 4.4 however also includes the new Dialogue Element node structure along with several Triggers that have also been appropriately tagged. Figure 4.4.ii and Figure 4.4.iii display a visualisation of two learners' adapted Dialogue Models. As before these learners have different user models so only parts of the Dialogue Model are relevant and so accessible. In this example the Triggers that have been added have also been tagged.

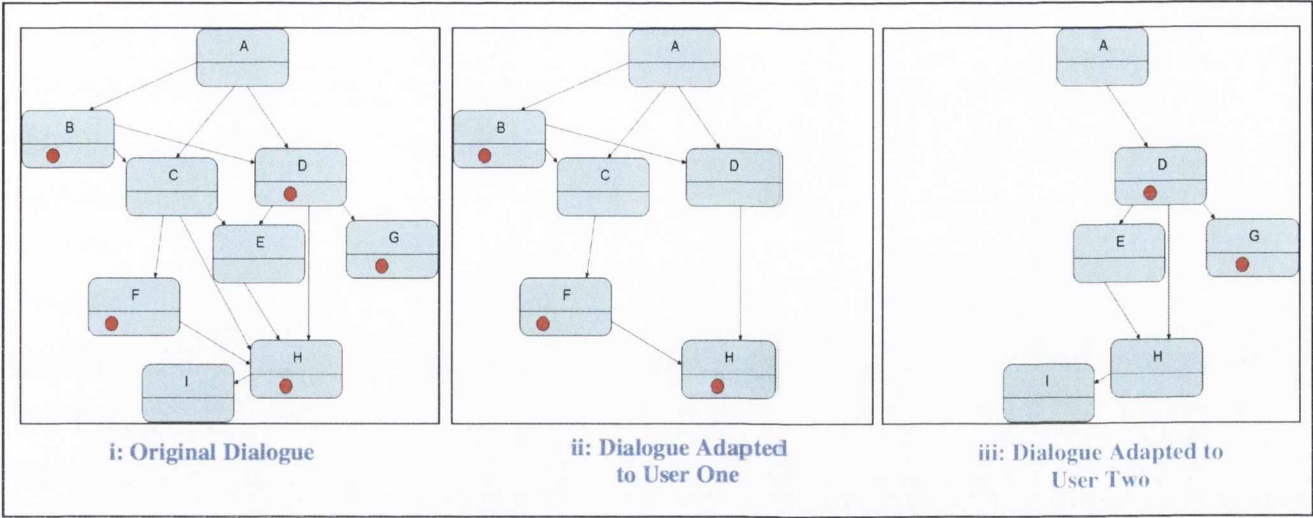


Figure 4.4: Example of Adapted Dialogue

A Trigger will only fire if the value of the tag matches that of a learner's user model dimension value. Figure 4.4 presents a visualisation of Triggers that are relevant to each learner. For Example, both Learner One and Learner Two have access to Dialogue Element D. Figure 4.4.ii shows that Dialogue Element D has associated with it a Trigger, this Trigger is relevant to the learner and will fire if Learner One accesses this Dialogue Element. Figure 4.4.iii shows that Learner Two can also access Dialogue Element D. However, the Trigger at this location is not relevant to the learner so will not fire for the learner as shown in the visualisation of the adapted model.

The concept of employing separate nodes to represent different components within a soft skill simulation model is not uncommon. However, most authoring tools usually allow authors to connect the different types of nodes together. By separating the adaptive triggers from the dialogue and not embedding them into the simulation model the authoring complexity is reduced and the model becomes easier to read. Furthermore, by placing the adaptive triggers on the Dialogue Elements instead of embedding them into the model they can be easily added, deleted and moved from one Dialogue Element to another without affecting the model that has been developed by the author.

Adaptive Dimensions

Selecting the dimensions to base adaptivity on within a system is very important. Dimensions should not only reflect characteristics of a learner but also describe aspects of the environment in which the learner is being taught. Within a soft skill training simulation

dimensions might be relevant to the pedagogy, dialogue and interaction within the system. The ACTSim system has been designed so as to separate the different models which describe the learner, simulation, and adaptivity. This allows dimensions to be added to and edited if necessary. The approach of tagging Dialogue Models means that adaptivity is not embedded in the model. The adaptive dimensions within ACTSim are extensible.

ACTSim employs a short concise list of adaptive dimensions that an author can select from; an exhaustive list of adaptive dimensions may cause the authors to become confused and overloaded. ACTSim also does not allow an author to create their own adaptive dimensions; this again creates an easy to use authoring environment that will not confuse the authors. The design of the system does however mean that adaptive dimensions can be added.

ACTSim has been initially designed to include four adaptive dimensions: learner role; learning outcomes; related subjects and categories. These adaptive dimensions were selected as they represent key characteristics which describe the user (role), the dialogue (related subjects), the types of interaction (categories) and educational aspects (learning outcomes) of the simulation. While other dimensions, such as learning style or previous knowledge, could also be applied to such an environment the selected dimensions are easily accessible to authors that do not have a background in adaptive hypermedia or a strong experience with composing eLearning courses that encompass complex pedagogical principals. Each selected adaptive dimension is described in the following sections. Included with each description are examples of the adaptive dimensions as related to psychiatric interview titles that were developed by experts during the course of this research. The purpose of the developed simulations was to assist third level medical students in their training to conduct interviews with different types of patients.

Role

The '*role*' adaptive dimension describes the function of a learner within a simulation. It is particularly relevant to soft skill simulations as a learner typically takes on one or more roles within the interactive immersive environment created by a soft skill simulation. Tagging Dialogue Elements or Triggers with a role value indicates that such components are relevant to a particular role. They will only be accessible to a learner if their role property matches.

Examples of roles used in the psychiatric interview titles that were created included ‘*Doctor*’, ‘*Nurse*’ and ‘*Care Giver*’.

Learning Outcomes

Describing the ‘*Learning Outcomes*’ of a simulation and associating them with a developed Dialogue Model is already described as a requirement of the Pedagogical Framework, described in section 4.1.4. A subject matter expert must initially create the Learning Outcomes associated with a scenario by creating Learning Outcome nouns and appending them to a Learning Outcome verbs; a more complete description of how Learning Outcomes are created can be found in section 4.2.5.

Association of Learning Outcomes is executed by tagging the Dialogue Model with Learning Outcomes that have been created by the author. This process allows adaptivity to be applied to the Dialogue Model based on Learning Outcomes that are of interest to individual learners. Particular Learning Outcomes that are to be achieved by a learner can be described in a learner’s user model. A Dialogue Model can be adapted appropriately so only the relevant Dialogue Elements, which are associated with the same Learning Outcomes, are included in the learners personalised simulation; as described by Dialogue Inclusion and Exclusion above. Examples of Learning Outcomes in the psychiatric interview titles developed included “*Explore if the patient has psychotic symptoms*” and “*Check if the patient has any associated anxiety symptoms*”.

Relevant Subjects

The ‘*subjects*’ adaptive dimension is a characteristic that describes the different areas that are discussed within a dialogue. Adaptivity can be applied to a Dialogue Model to insure that only subject areas that are relevant to learner will be included in the simulation. A learner’s user model indicates the subjects that are of interest to a learner. By matching the subjects described in a learner’s user model to a tagged Dialogue Model it can be personalised to the learner so they can only access areas that are relevant. A learner may be interested in a number of subjects which can be indicted in their user model. Conversely, a Dialogue Element or sub-dialogue might be related to several subject areas and so may be tagged with more than one relevant subject tag. This characteristic can be thought of as metadata which

describes the content of the dialogue. Examples of subjects within the developed psychiatric interviews included “*Family History*” and “*Sleep*”.

Interactivity Categories

The approach to dialogue representation decomposes the dialogue into smaller more manageable components. Each of these components can be interacted with by a learner as a selection they can make within a dialogue. The interactive nature of these components allows them to be categorised in a manner that is not typical of traditional Adaptive Hypermedia. It is possible to describe Dialogue Elements in relation to the type of interaction they embody. This allows an author to categorise the different types of interaction that can occur within the Dialogue Model that they have created. Categorization of this nature is closely related to the purpose and type of dialogue that has been created. For example, simulations used to teach interview skills would have a great deal of question types (open or closed questions), simulations could be confrontational and be categorised with levels of hostile engagement (passive to antagonistic) or simulations may comprise of interaction that requires the learner to be supportive (helpful to disagreeable). If the *subjects* adaptive dimension describes the content of a dialogue, the *category* adaptive dimension describes a dialogues style.

Adapting a Dialogue Model based on categories means that a dialogue can be focused towards particular types of interaction. A learner’s user model describes the types of interactive categories that are relevant to that particular learner. These are matched to a Dialogue Model that has been tagged with categories values so only relevant Dialogue Elements are made available to the learner. Examples of categories within the psychiatric interviews that were developed included “*Open Questions*”, “*Closed Questions*” and “*Advice*”.

Combining Adaptive Dimensions

The adaptive dimensions described can be used singularly or in combination to create a personalised dialogue. Dialogue Elements can also have multiple tag values for a single dimension. A learner can also be assigned multiple values for each of their associated adaptive dimensions. The more detailed the tagging of a Dialogue Models the greater the degree of adaptivity that will be available. The system allows Triggers can be tagged with all of the same adaptivity dimensions as the Dialogue Model. Employing the same adaptive

dimension with Triggers allows the author control over how and when a Trigger will fire. A Trigger may be related to a particular role, subject, learning outcome or category.

Triggers – an Adaptive Mechanism

Triggers are a type of mechanism which operate (adapt) using adaptive dimensions. There are three types of Triggers that can be employed in a soft skill simulation developed with ACTSim; ‘Assessment’; ‘Feedback’; and ‘Reflection’. Each type of Trigger allows the author to educationally enhance a simulation. Triggers are incorporated in simulations based on typical learning activities that learners participate in while engaged with educational environments. Tagging the Triggers means that they will only fire if they are relevant to individual learners.

The ACTSim authoring tool allows an author to add multiple Triggers to a Dialogue Element. Triggers should however only occur occasionally within a simulation. An author risks the flow of a dialogue being broken if the learner is constantly being interrupted with Triggered events. The author must also consider that not all Triggers will be available to each learner during a simulation once adaptivity has been applied. The author is responsible for finding the right balance between enhancing the simulation with Triggers and ensuring that the flow of the dialogue remains intact. The following paragraphs describe each Trigger type.

Assessment Trigger

Typically eLearning environments assess a learner’s progress during or after a course. Assessments allow instructors to measure a learner’s progress by examining what they have learned. There are two main types of Assessment; summative and formative (Taras 2005). Summative Assessments are the more traditional view of assessments that involve a learner completing a test or quiz which is scored with results delivered back to the learner. A learner’s score can be used as guide to their progress through the course.

Formative assessments are used to encourage learners to contemplate a particular line of thought. This is accomplished by examining a learner with a question or series of questions. While the learner is required to give a response the answer is not recorded or correlated by the system. The goal of a formative assessment is not to evaluate the progress of a learner but

to act as a learning aid within the simulation. Once the learner has considered the question and given a response they are provided with the correct answer.

The ACTSim authoring tool allows an author to add Assessment Triggers and assign them to be summative (default) or formative. Summative Assessment Triggers can be associated with a URI location which describes the Assessment content. Formative Assessment Triggers gather their content (question asked of the learner with corresponding correct answer) from Dialogue Elements that the learner navigates through. A Formative question and answer can be associated with each Dialogue Element in a Dialogue Model; Assessment Questions and Answers become another property associated with Dialogue Elements. As a learner navigates through a Dialogue Model the system gathers all Formative question and answers which are to be presented to a learner. The gathered content will only be presented to a learner when they fire a formative Assessment Trigger; this will only occur if a learner navigates to a Dialogue Element that contains such a Trigger. As such the series of question and answers that are delivered to a learner are adaptive as they are dependent on the route which the learner has taken. The actions of the learner determine the content of the Formative Assessment Trigger.

An author may also want a Formative Assessment to use only the Assessment Question and Answer from the Dialogue Element which contains it; i.e. not an accumulation of all Dialogue Element Assessment Question and Answers it has navigated through but instead only using a single Assessment Question and Answer. This becomes an additional option available to an author and they can decide if a Formative Assessment should be '*Instant*' or '*Cumulative*'.

Tagging the Assessment Trigger allows an author to determine its relevancy to learners and hence when it should be fired within the simulation. For example, the content associated with a particular Summative Trigger may only be relevant to a particular role or interaction category.

Feedback Trigger

Feedback during an eLearning course allows an instructor to deliver relevant information and data to a learner. This could include the total time that has elapsed since beginning the

simulation; number of logins; the percentage of the simulation that the learner has completed; what learning outcomes have been addressed by the learner; etc. ACTSim allows an author to determine where Feedback will occur within the simulation and what data is to be presented; a list of check box options allows an author to indicate what content is to be delivered to the learner. Tagging allows an author to designate the relevance of the Feedback Trigger. Feedback could be suited to learners who are interested in a particular relevant subject for example. The content delivered is also dynamic and is based upon the learner's actions and progress in the simulations.

Reflection Trigger

Instances of reflection allow a learner to consider and contemplate what they have learned during their educational experience. Within a simulation instances of reflection are focused to particular areas of learning by prompting a learner. Soft skill simulation composed with ACTSim allow a learner to record their thoughts in a text box that is provided. These notes can then be saved within the simulation so a learner can review them at a later date.

The ACTSim authoring tool allows an author to add Triggers and tag them as described with the previous two Triggers. Similarly an author can ensure that Reflection Triggers are only available to appropriate learners through tagging. ACTSim allows an author to describe the prompt that is to be delivered to the learner. This could be in the form of a statement or question which is designed to focus a learners line of thought.

4.2.4 Authoring Process

The authoring process employed by ACTSim (Gaffney, Dagger et al. 2007) is an iterative step-by-step cycle. Its focus is the development of the Dialogue Models upon which the simulations will operate. The development of the model is in fact the main focus of the simulation and an iterative step in itself within the authoring process. All other steps of the process are in place to support the development of the Dialogue Model. The authoring process includes steps for applying adaptivity, incorporating a pedagogical framework and describing the context of the simulation.

Education involves the activities of instructing, teaching, facilitating and supporting knowledge transfer/acquisition and learning. Educational development typically involves the detailed process of identifying base requirements and forecasting further demands on learners

at all levels. In the case of formal education, this process can be captured in an educational/training development plan or syllabus.

An educational development plan may be used to describe such things as educational focus, educational prerequisites, performance objectives, accreditation criteria, evaluation mechanisms and conceptual scoping. A core concept in an educational development plan is termed “*Constructive Alignment*”.

Constructive Alignment

A key aspect of any eLearning development process is alignment. This involves aligning the stated learning outcomes with the teaching methods used to achieve the outcomes and the assessment metrics used to measure the success of the teaching methods at achieving the desired outcomes (Mayes and de Freitas 2004). Figure 4.5 illustrates this cyclical process of alignment.

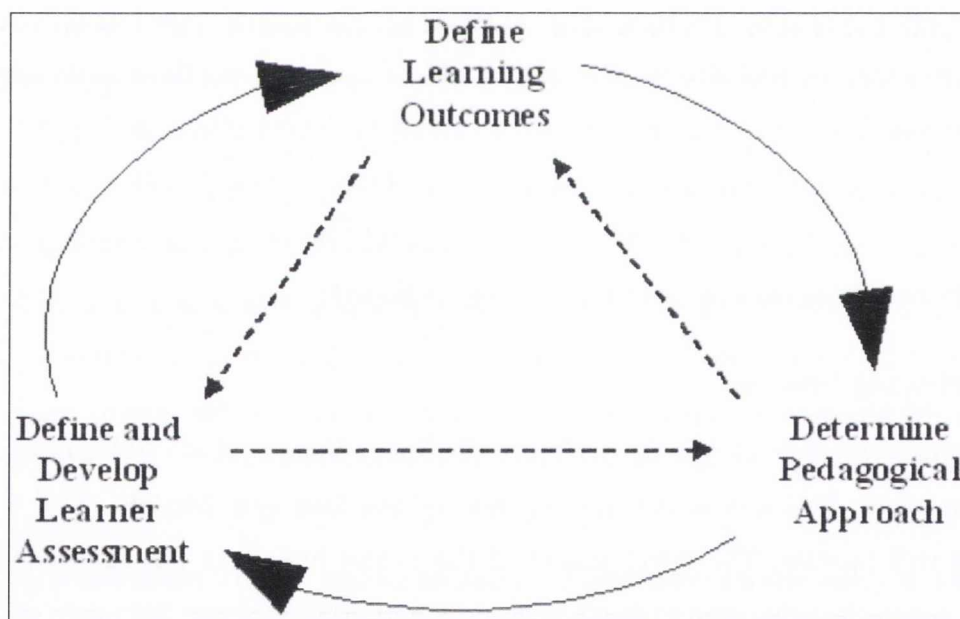


Figure 4.5: Constructive Alignment

Alignment is a cyclical process involving the identification and description of three different steps. The first step is to define learning outcomes. Learning Outcomes describe what the learner should know at the end of their learning experience. The next step is to determine the pedagogical approach, which defines the teaching strategy to be incorporated to achieve the

outcomes. The final step is to define and develop the learner assessments, which describe the ways the learner should be assessed so as to demonstrate what they have learnt.

Alignment and Simulation Development

The development process is at least iterative and at most rapid development. It must balance the incremental effort in designing a simulation and the necessary evaluation and feedback from subject expert and potential learner. Therefore the design process suggested is an incremental agile process in which an author does not necessarily need to tackle every design step in each iteration. It is only through continual iterations of the process and evaluations of the composed simulation that all design steps are fully addressed. The steps, illustrated in Figure 4.6, consist of the following processes:

Describe Scenario: this is a high level description of the simulation which includes the context of the simulation and the different roles that it contains; this can be thought of as a simulation's Abstract. It allows the author to generally summarise the overall objective of the simulation. This leads into the description of Learning Outcomes and Pedagogical Approach which can be defined accordingly.

Define Learning Outcomes: these are the basic cognitive competences to be demonstrated, which are knowledge, comprehension, application, analysis, synthesis and evaluation (Mayes and de Freitas 2004). For the purpose of this research learning outcomes have been designed to be composed of two primitives developed by an author; the Learning Verb and Learning Noun.

Determine Pedagogic Approach: this allows the author to define the educational strategies and instructional methods that are best suited to achieving the stated Learning Outcomes within the simulation. An author initially defines the overall basic objective of the simulation in the first step of the authoring process. This step of the authoring process however allows an author to refine and expand this objective and relate it to dialogue that will occur in the simulation. The Pedagogical Approach is also closely linked to the style of the dialogue incorporated within a simulation and is described within this step; this can affect such things as the tone and type of branching that might be used when composing the dialogue model.

Design and Compose Dialogue: this is the design and definition of the dialogue models. This is an iterative step within itself, whereby the designer specifies all possible dialogue paths they wish the learner to be able to access. One strategy for designing dialogue models is to initially define a core/root stem which captures and describes the most direct path through a simulation. A subject matter expert can then add branches which describes the remainder of the simulation. This strategy is presented in further detailed below in the Dialogue Model Authoring Process section.

Tag Dialogue Model: this step allows the author to apply tags to the dialogue model that they have created. Authors may tag individual components as they are created or they may tag large sections of the model (sub-dialogues) once they have been completed.

Define and Develop Triggers: this step allows the definition and development of adaptive triggers. An author can create Assessment, Feedback (which is particularly important to constructive alignment) and Reflection Triggers, populate them and places them at appropriate points within the model. As the author creates Triggers they can be tagged appropriately. As Triggers are not directly linked to the simulation model they can be added at a later stage of development once a model is near completion.

Evaluation/Verification: as with the educational development, the simulation development is an iterative process, as illustrated in Figure 4.6. To ensure that each iteration is successful, the final step, before beginning the cycle again, is an evaluation of the development process that has just been completed and the developed simulation model. The results of this evaluation are used to improve the next development process, simulation model and applied adaptivity. The author can also use verification tools to aid them in their evaluation.

Develop Content: On the final iteration of the development process, once the designer is satisfied that all educational and simulation models are completed the content is recorded. This is only completed in the final iteration as it is usually a costly and time consuming process which should not to be repeated in every cycle.

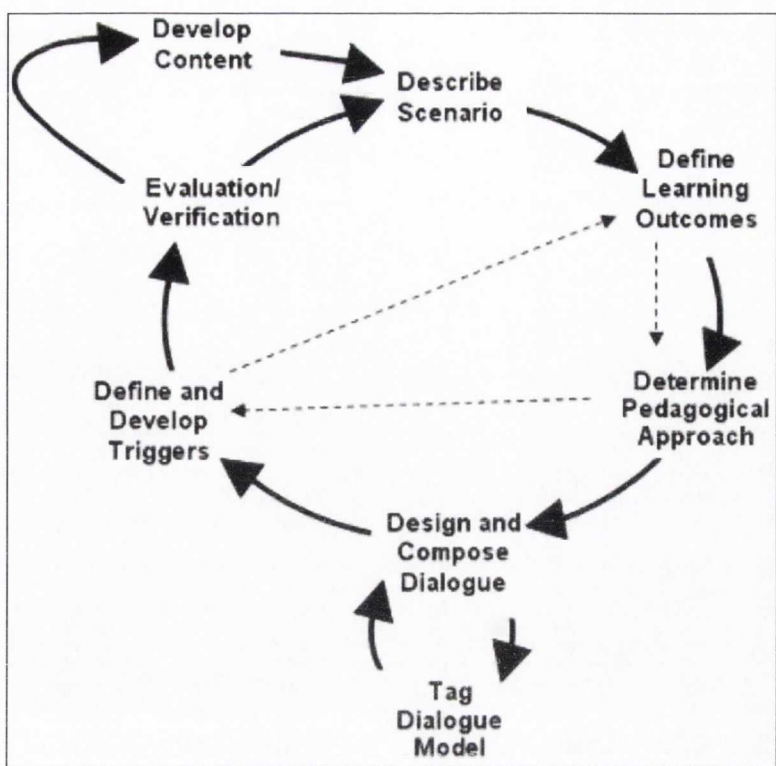


Figure 4.6: Pedagogically Informed Process for Composing Adaptive Dialectic Simulations

This pedagogically informed process creates the base for the pedagogic support framework for composing adaptive soft skill simulations.

Dialogue Model Authoring Process

The design of the graph based approach incorporated in Dialogue Representation results in an interface that is non-restrictive; an author is allowed to place and connect Dialogue Elements freely within the Dialogue Space. While this approach to Dialogue Representation is necessary to allow authors easily and rapidly develop Dialogue Models it requires a process to guide an authors model development. The Dialogue Model Authoring Process outlines an approach for creating a Dialogue Model which allows an author to effectively and efficiently create Dialogue Models which are suitable for a soft skill simulation; development of the model is focused on creating an intelligible and coherent dialogue.

The ACTSim Dialogue Model Authoring Process requires an author to initially create what is known as the simulations ‘*Main Stem*’. This is the most direct and optimal route a learner can take within a simulation. It is constructed as a string of connected Dialogue Elements (nodes) within the Dialogue Space. Once an author has completed the Main Stem they can begin to add ‘*Branches*’ or sub-dialogues which detail different aspects of a simulations dialogue. The

Main Stem and Branches are added to, edited, amended and appended through many iterations of development until the Dialogue Model is complete.

The Main Stem Branching technique also helps to protect the Dialogue Model from becoming disjointed when adapted through inclusion/exclusion. The Main Stem would not typically be tagged for adaptivity within a model so would be constant throughout all learners' adapted simulations. Sub-Dialogues will also typically be linked back into the Main Stem to allow learners to continue through the simulation. The exception to this of course is if a learner has made a selection in the simulation that would cause the scenario to end.

4.2.5 Pedagogical Framework

There are two levels of pedagogical framework employed in the ACTSim authoring tool. These allow an author to accurately describe the educational purpose of a soft skill simulation; both approaches are described in the following sections.

Pedagogical Approach

A simulations pedagogical approach describes a very high level description of the educational purpose of the simulation. The pedagogical approach allows an author to provide an abstract view of not only the educational strategy to be incorporated but also the style of the dialogue to be used in the simulation. To allow an author to input data pertaining to the pedagogical approach an open text format is employed. This approach accommodates the non-structured nature of the pedagogical approach as it is flexible and non-restrictive. The pedagogical approach will invariably change and evolve as an author develops their dialogue model and simulation. For example an author may initially believe that the learner's interaction will be passive and supportive but as the simulation develops the interaction becomes rather more confrontational.

Learning Outcomes

Compared to the Pedagogical Approach, Learning Outcomes are a much more concrete description of the educational goals of a soft skill training simulation. They describe very specific educational objectives that a learner is to achieve within a simulation. ACTSim has also been designed to allow authors to associate Learning Outcomes with Dialogue Elements and Branches within the Dialogue Model. Association of this type allows an author to insure that Dialogue Elements are serving an educational purpose within a dialogue and that each

Learning Outcome is being addressed in some manner within the Dialogue Model. The ACTSim system allows authors to associate more than one Learning Outcome with Dialogue Elements as they may address multiple learning goals.

As previously described, the creation and association of Learning Outcomes with Dialogue Elements is also used for personalising simulations. Association of Learning Outcomes with Dialogue Elements is realised within ACTSim through tagging; this is the same approach used in adaptivity tagging. ACTSim allows an author to view (highlight) Dialogue Elements that have been tagged with particular Learning Outcomes, this functionality is discussed in more detail later in this chapter.

Learning Outcomes are rigid structures within the authoring tool and are composed of two basic elements; a '*Learning Outcome Verb*' (Learning Verb) and a '*Learning Outcome Noun*' (Learning Noun). This approach to structuring, while still very descriptive, is designed to be easily accessible to all authors, even those that might not have a strong background in pedagogical design. Learning Verbs are based on a subset of Blooms taxonomy of verbs (Anderson and Sosniak 1994) and ACTSim provides authors with an initial list which they can edit and amend as required, see Appendix II – Simulation Learning Verbs. The Learning Nouns components are created by the author. These are very specific to the simulation that an author is developing. Learning Nouns typically describe different subject areas that are addressed within a dialogue but are typically far more concise than the more general related subjects that are used to adaptively personalise a simulation.

To create a Learning Outcome an author selects the appropriate Learning Verb and Learning Noun from lists that they have previously created/edited. Creating a Learning Outcomes also requires an author to provide a description of each Learning Outcome. This documentation is a useful reference for an author and can also be of great assistance if a course is passed from one instructor to another.

4.2.6 Navigational Aids

ACTSim employs several navigational aids to assist the author in navigating the Dialogue Space and locating particular Dialogue Elements. As an author develops their Dialogue Model they will employ different navigational aids depending on their situation. The following sections describe the design of each of these navigational aids.

Map

The Map functionality is typically the most used navigational aid. It allows an author to quickly and easily move from one area of the Dialogue Space to another. It also constantly provides an author with an overview of their model. The Map of the Dialogue Model is located in a separate window to the Dialogue Space so as not to obstruct an author's view of their model. The Map window can be minimized, resized and moved within the ACTSim interface. The Map view is also dynamic and updates in real time as the Dialogue Space is edited.

Importantly the Map functionality also employs an interactive element. The Map incorporates a visual display (a blue box) which highlights the area that is at that time being viewed in the Dialogue Space window of the authoring tool. The author is allowed to move the highlighted area within the Map window to navigate the Dialogue Space accordingly. Interaction with the highlighted box is executed with the mouse cursor and allows an author to efficiently and effectively move from part of their Dialogue Model to another.

Zoom

The Zoom functionality allows an author to alter the magnification of the Dialogue Space window. This allows an author to zoom in or out of their model to view different perspectives. While the Map functionality presents an overall view of an author's Dialogue Model it does not allow an author to view or interact with individual sections. Such functionality is particularly useful to an author if their development is concentrated on a single sub-dialogue or Branch of the Dialogue Model.

The ACTSim authoring tool allows two methods of operating the Zoom functionality. The first method allows an author to select the Zoom operation from the palette; increase or decrease operations are indicated in the palette with plus and minus sign respectively. Once selected, the authoring tools mouse cursor changes to a magnifying glass icon. An author can then point the mouse cursor in the Dialogue Space and left click the mouse to select the area which they wish to zoom in or out from.

The second method of operating the Zoom functionality is by setting the Dialogue Space magnitude by selecting from predetermined intervals. This operation can be selected by using a dropdown menu list which is integer based and magnitude is denominated in units of percentage. For example, an author could set the Dialogue Space view to be 25%, 50%, 100%, etc.

Search

The Search function allows authors to locate specific Dialogue Elements in their model. The search is string based and allows an author to locate Dialogue Elements based on values that have been input into Dialogue Element properties. The Search function allows an author to locate Dialogue Elements with regard to String values contained in their Statement, Response or Notes/Directions properties. To indicate a match has been made between the search value that an author has input and the Dialogue Element property the Dialogue Elements are highlighted. The Search function is accessed through a popup window within the authoring tool; the same interface also allows an author to clear all previously highlighted Dialogue Elements.

List View

The List View displays a sequential list of all Dialogue Elements that have been created by an author. The purpose of the List View is to allow authors to locate particular Dialogue Elements within the Dialogue Model that they have created. The List View displays Dialogue Elements in relation to their Statement values. This allows an author to browse a list of all Dialogue Element Statements to locate particular instances. If an author selects a Dialogue Element from the List View the Dialogue Space navigates to that particular Dialogue Element in the author's model and indicates it to the author with a cursor selection. The List View is particularly helpful if an author is searching for a Dialogue Element but is unsure of the exact phrasing of its Statement. The List View of Dialogue Elements is contained in the same window as the Map view with only the List View or Map being viewable at any one time. The List View can be accessed through tab functionality that allows an author to change back and forth from the Map view to the List View.

Arrange

The purpose of the Arrange function is to restructure a Dialogue Model so it becomes more accessible and easier for an author to read. Restructuring of the graph is based on a tree algorithm which incorporates the initial node (Dialogue Element which represents the beginning of the simulation) as the root of the tree. The Arrange function is particularly useful if a Dialogue Model is being navigated by an instructor that was not the original author. This might occur if an instructor inherits a course and wishes to redesign or edit an existing simulation title that they have acquired. The arrange function can be accessed through a menu button found at the top of the authoring tool.

4.2.7 Model Verification

There are several aspects of a Dialogue Model that require verification. The ACTSim authoring tool employs three functions that allow an author to examine the different elements and design of their model. Each of these functions is described in the following sections.

Script

The Scripting function allows an author to examine sequences of dialogue that are contained in a Dialogue Model. The purpose of this function is to allow an author to insure that the dialogue is appropriate and meaningful. Such verification is not typically necessary with more traditional eLearning Adaptive Hypermedia systems as they are designed without the need to consider continuity. Learners in conventional eLearning courses move from one component to another without there necessarily being a direct link between the two. With dialogue based soft skill simulations however the interactive dialogue must be intelligible and relevant to the previous interaction.

Scripting allows an author to verify that connections between Dialogue Elements are valid by displaying the sequence of the dialogue as it would be viewed by the learner. In order to access the scripting function an author initially selects the Dialogue Elements to be included in the scripts dialogue sequence; the order of selection determines the sequencing order of the Dialogue Elements. The author can then execute the Scripting function within the Dialogue Space to view the sequencing of the Statement and Responses. A separate function in ACTSim also allows an author to script an entire Dialogue Model without needing to select any Dialogue Elements. This function is useful for generating scripts to be employed for recording video or audio at the end of a simulations development.

Highlight

The Highlight function allows an author to view which Dialogue Element within a Dialogue Model have been assigned particular tag values. This allows an author to insure that all adaptivity and Learning Outcome tags have been applied within a simulation. An author can also ascertain the frequency and location of tags within a model. Using the highlight function an author may decide that a particular tags have been used too much, not enough or in the wrong areas of a model. The Highlight function has also been designed to allow an author to view combinations of tags within a Dialogue Model. An author can verify that tags are being combined correctly.

Validate

The Validate function allows the author to verify that essential data has been entered into Dialogue Elements; specifically Statement, Response and the media files associated with both. These are data instances which are vital to a simulation operating correctly and it is necessary that, at different stages of development, an author can automatically check values have been added for these properties in each Dialogue Element. ACTSim has been designed so as to allow an author select which of these properties or combination of properties are being included in the Validate function. For example, before media file have been recorded there would be no need for an author to validate the existence of media file properties.

4.3 Phases of Development

While the design elements in this chapter have been presented concurrently it should be noted that they were introduced into the system over two primary phases of development. As previously described in chapter one, the main objective of the first phase of development was to address basic (non-adaptive) authoring features, specifically the dialogue representation. The main objective of the second phase of development was to introduce adaptivity features and functionality. The second phase of development was also used to introduce additional authoring requirements, not related to adaptivity, and to re-evaluate features already incorporated into the system. It should be noted that only one element of the Authoring Process, the Dialogue Model Authoring Process, was introduced in the first phase of development.

The primary process for each phase followed the action based research approach as outlined in chapter one. This was to follow a cycle of plan/reflect, act and evaluate with the results of the first phase feeding into the second phase; as presented to the left of Figure 4.7 (Mac Issac 1995). However, within each primary phase of development there were also many sub-cycles of action based research taking place; further details of this approach are outlined in chapter six. To the right of Figure 4.7 a summary of the design elements with respect to each of the two primary phases of development is presented.

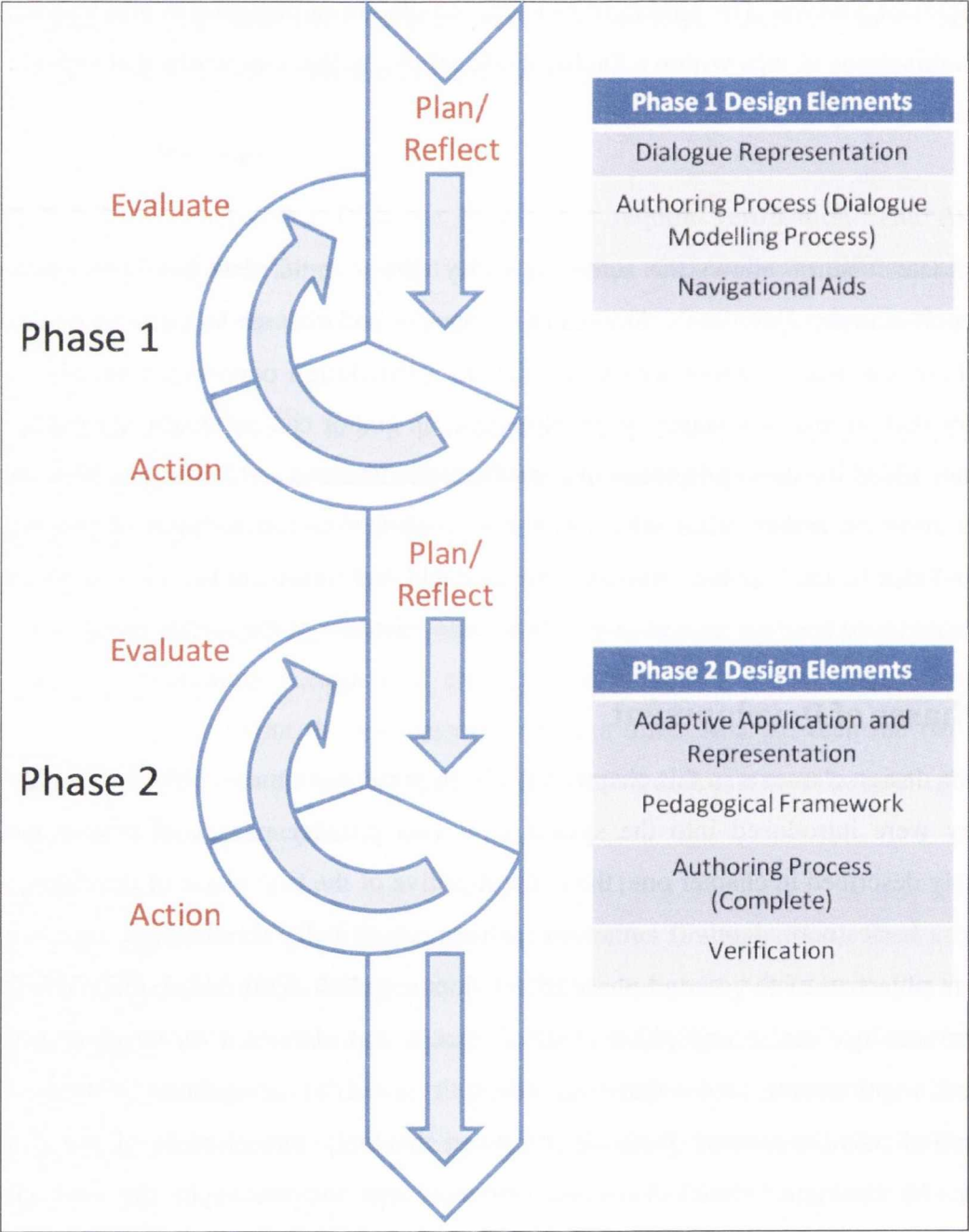


Figure 4.7: Phases of Development and Design Elements (Mac Issac 1995)

4.4 Summary

This chapter has described the design of the main artefact of this research, the ACTSim authoring tool. The chapter initially presented the key requirements of ACTSim and in turn described the design of each of these requirements. Aspects of ACTSims design included dialogue representation, adaptivity application and representation, the associated authoring process, pedagogical framework, navigational aids and model verification. The final section of this chapter outlined how the different requirements were introduced with respect to the two different phases of development. The following chapter describes in detail the implementation of each of the components described in this chapter.

5 Implementation

This chapter describes the implementation of the ACTSim authoring tool. It discusses the process of implementation through a series of sections. Each section details a different aspect of the implementation. The first section, 5.1, discusses the architecture of ACTSim and its associated systems. The following sections present the component architecture, 5.2, and technological architecture view of the composition tool, 5.3. The next section, 5.4, describes the structuring of simulation title models. The following section, 5.5, describes an authored simulation while the final section, 5.6, presents a summary of the chapter.

5.1 ACTSim Architecture

The ACTSim authoring tool is one of several subsystems that are required to compose and deliver adaptive soft skill simulations. It is a component of what can be considered a typical eLearning web services architecture (Musa and de Oliveira 2005). Titles that have been created by authors in ACTSim are stored locally as xml files. Stored course titles are uploaded by authors using a remote xml data store. Once titles have been uploaded to the remote server they can be accessed by an adaptive engine which then applies adaptive rules. The adaptive rules create individual customised simulations using each learners associated user model. Figure 5.1 presents the architecture involved in composing, adapting and delivering personalised soft skill simulations.

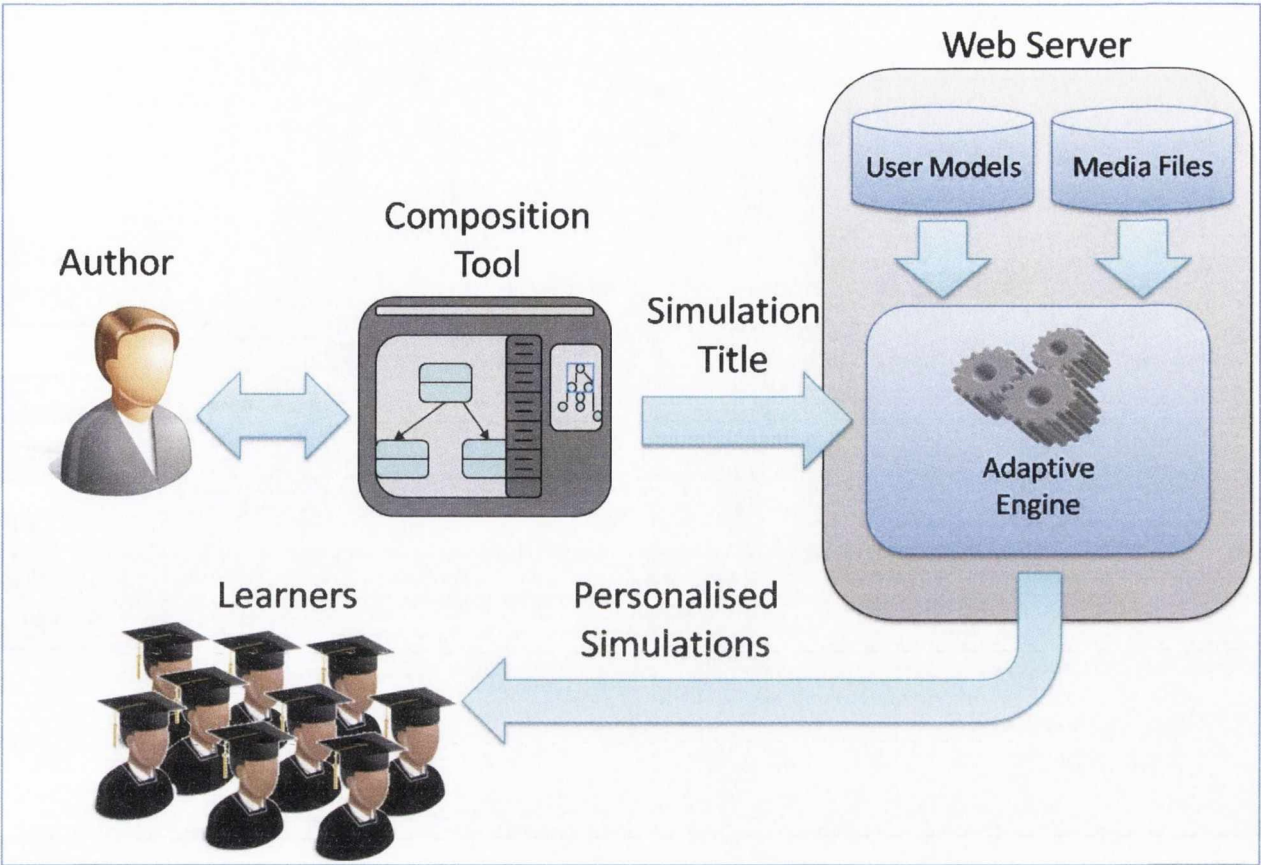


Figure 5.1: Overview of Soft Skill Simulation Development and Delivery

5.2 Components Architecture View of ACTSim

The purpose of the ACTSim authoring tool is to allow non-technical subject matter experts to easily and rapidly develop course titles. In order for ACTSim to be accessible in a typically office or educational environment it has been built as a standalone and platform independent application. Within ACTSim there are five main components used to describe its architecture; Canvas; Dialogue Element; Feedback Trigger; Assessment Trigger; and Reflection Trigger. Their architectural relationship in presented in Figure 5.2. An author directly interacts with the Canvas component which acts as the main containment component of the ACTSim authoring tool. It is this component that contains all other simulation components and is also used to store simulation scenario properties. The following sections describe each of the components in detail.

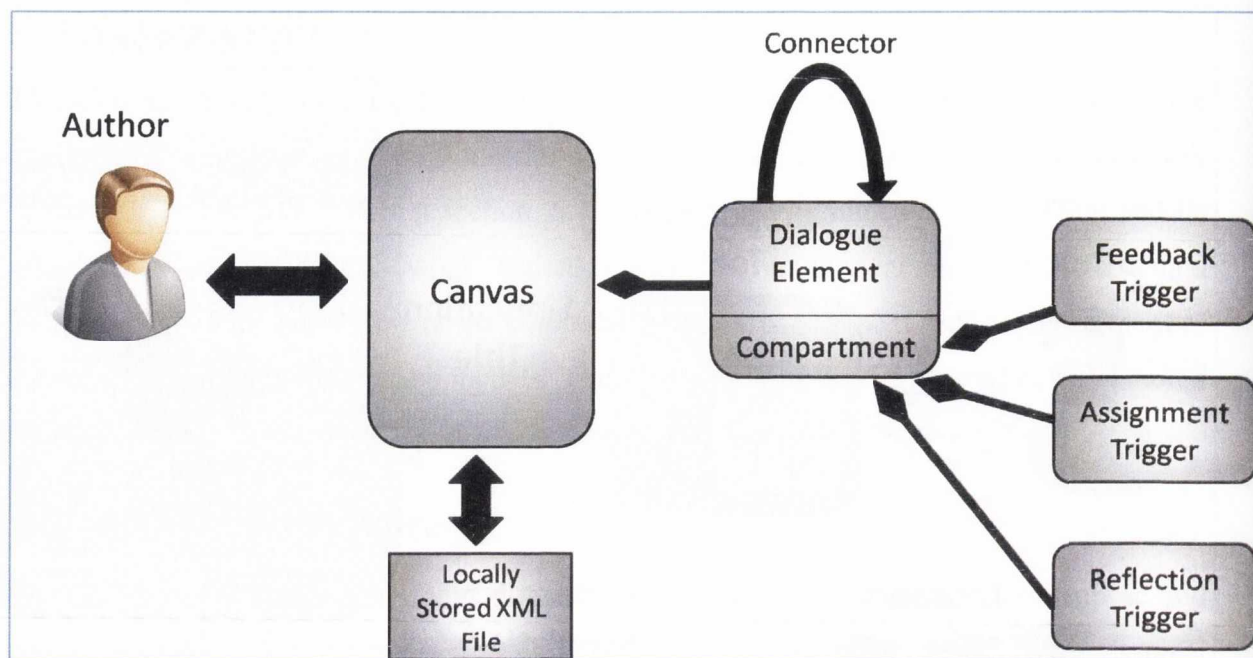


Figure 5.2: ACTSim Architecture

5.2.1 Canvas

All functions within the ACTSim authoring tool operate on or about the Canvas component. It can be conceived of as being the main body of the model that contains all data and information pertaining to the soft skill simulation. Within ACTSim application the Canvas component can be thought of as the area in which an author builds and composes a Dialogue Model.

Architecturally speaking only Dialogue Elements can be placed on the Canvas component, as indicated in Figure 5.2. The Canvas component also acts as a storage location which contains all of the scenario properties of a simulation. Simulation scenario properties are hidden from an author within the composition tool until they are accessed through appropriate functionality. The simulation scenario properties contained within the Canvas component include:

Name: The name of the soft skill simulation; a simple String property.

Identifier: A short set of characters or numbers unique to each soft skill simulation as determined by the author.

Description: A short text based description of the soft skill simulation which may include its purpose and intended learner group; this can be thought of as a simulation's Abstract.

Pedagogical Approach: A text based description of the educational strategies that will be used in the simulation; this may also include a description of the style of the dialogue that is to be employed by the simulation.

Learning Verbs: A list of Learning Outcome Verbs used to define Learning Outcomes; an author is supplied with an initial list, based on Blooms taxonomy, which can be edited and appended. Each Learning Verb is comprised of a name and description.

Learning Nouns: A list of Learning Outcome Nouns used to define Learning Outcomes; Learning Nouns are defined by an author. Each Learning Noun is comprised of a name and description.

Learning Outcomes: A list of Learning Outcomes which describe all the possible Learning Outcomes that can be achieved within a simulation; each Learning Outcome consists of a Learning Verb and a Learning Noun which have been previously defined within the authoring tool.

Roles: A list of Roles a learner can function as within a soft skill simulation. Each Role is comprised of a name and description.

Subjects: A list of Subjects that describe the different areas that are discussed within a soft skill simulation dialogue. Each Subject is comprised of a name and description.

Categories: A list of Categories that describe the different types of interaction that are contained within a soft skill simulation. Each Category is comprised of a name and description.

5.2.2 Dialogue Element

Dialogue Elements primarily represent the Statement and Response that occurs in the dialogue of a soft skill simulation. They are the main building blocks an author uses to create a Dialogue Model. Dialogue Element components are added to and contained within the

Canvas component of the ACTSim architecture. Dialogue Elements can be connected to each other to indicate a simulation's dialogue paths within a Dialogue Model and also incorporate a compartment which can be used to contain Triggers; the Dialogue component architecture can be seen in Figure 5.2. ACTSim provides several methods of functionality to add and connect Dialogue Elements within the Canvas component. The Dialogue Element component also contains additional properties to Statement and Response. All of a Dialogue Element's properties can be accessed by an author by double clicking its graphical representation. Additional properties assist the author in managing Dialogue Elements and creating an effective and complete Dialogue Model. They include:

Id: Each Dialogue Element is assigned a unique identifier by an author which is stored as a String; id properties are typically assigned in a numerical sequence as each Dialogue Element is created.

Statement: The Statement property is text based and describes the dialogue options that are available to a learner within a soft skill simulation.

Statement URI: A media file that is used to deliver a soft skill simulation; these are typically a short video or sound file that captures the text contained within the Statement property.

Response: The Response property is text based and describes the response of the simulated person within the dialogue of a soft skill simulation. It is coupled with the Statement property and acts as a suitable reaction to the selected Statement.

Response URI: A media file that is used to deliver a soft skill simulation; these are typically a short video or sound file that captures the text contained within the Response property.

Learning Outcomes: A list of Learning Outcomes that have been associated with the Dialogue Element which describe the educational goals that the Dialogue Elements addresses. These are to be realised as Learning Outcome tags that have been assigned to the Dialogue Element.

Roles: A list of Roles that have been associated with a Dialogue Element which describe the type of learner function that it has been designed to accommodate. These are to be realised as Role tags that have been assigned to the Dialogue Element.

Subjects: A list of Subjects that have been associated with the Dialogue Element which describe the dialogue areas that it encompasses. These are to be realised as Subject tags that have been assigned to the Dialogue Element.

Categories: A list of Categories that have been associated with the Dialogue Element which describe the type of interactions that characterise its Statement property. These are to be realised as Categories tags that have been assigned to the Dialogue Element.

Assessment Question and Answer: The Assessment Question and Answer property is text based and used to describe the associated formative assessment information of a Dialogue Element.

Notes: A text based property that describes notes or instructions that are associated with each Dialogue Element. This property is typically used to summarise directorial direction or actions to be used in video/sound recording.

5.2.3 Trigger Components

Trigger components (Feedback, Assessment and Reflection) are contained by the Dialogue Element component. An author can only add a Trigger to the compartment of a Dialogue Element that has been previously added to a Canvas. This means that Trigger components cannot be added directly to the Canvas and can *only* be placed in the compartment of a Dialogue Element. This approach is necessarily restrictive but does still allow Trigger components to be moved from one Dialogue Element compartment to another. Authors can access Trigger component properties by double clicking the component. While Trigger Components contain properties that are specific to their type they also contain properties which are common. These include:

Learning Outcomes: A list of Learning Outcomes that have been associated with the Trigger which describe the educational goals that it is designed to be fired with; Learning Outcome tags that have been assigned to the Trigger.

Roles: A list of Roles that have been associated with a Trigger which describe the type of learner function that it has been designed to be fired with; Role tags that have been assigned to the Trigger.

Subjects: A list of Subjects that have been associated with the Trigger which describe the dialogue areas that it should be fired with; Subject tags that have been assigned to the Trigger.

Categories: A list of Categories that have been associated with the Trigger which describe the type of interactions that it has been designed to fire with; Categories tags that have been assigned to the Trigger.

The following sections describe each type of Trigger Component.

Feedback Trigger Component

Feedback Triggers allow an author to indicate the location of Feedback that could be delivered to a learner within a simulation. By tagging a Feedback Trigger an author can personalise a Feedback Triggers firing action. There are also several different types of feedback that can be delivered to a learner. Each type of feedback is implemented as a Boolean property within the Feedback Trigger. An author determines the type(s) of feedback to be delivered when a Feedback Trigger is fired by selecting them from a list within the Triggers interface. A particular type of Feedback is presented to a learner if it is stored as a 'true' primitive within the Triggers data model. Properties that are specific to Feedback Triggers include:

Name: The name assigned to the Feedback Trigger; typically this will give a brief indication of the Triggers application.

General Description: A text based description of the Feedback Triggers objective within the simulation; typically describing its purpose and intended adaptive function.

Duration: A binary property which indicates if the Feedback Trigger will present a learner with data pertaining to the amount of time they have spent active within the simulation.

Progress: A binary property which indicates if the Feedback Trigger will present a learner with data pertaining to their progress within the simulation; presented as the number of nodes accessed by the learner compared to the number of total nodes within the Dialogue Model.

Assessment Results: A binary property which indicates if the Feedback Trigger will present a learner with results of previously completed Summative Assessments.

Percentage Complete: A binary property which indicates if the Feedback Trigger will present a learner with data pertaining to the percentage of the simulation that they have completed.

Number of Attempts: A binary property which indicates if the Feedback Trigger will present a learner with data pertaining to the number of times a learner has started a simulation from the beginning.

Assessment Trigger Component

Assessment Triggers allow an author to indicate the location of Assessments that could be delivered to a learner within a simulation. By tagging an Assessment Trigger an author can personalise an Assessment Triggers firing action. Assessment Trigger components contain several different properties which can be accessed by an author through the components interface and allow an author to control the type and purpose of the Assessment. The following descriptions detail the options that are available to an author. Properties that are specific to Assessment Triggers include:

Name: The name assigned to the Assessment Trigger; typically this will give a brief indication of the Triggers application.

General Description: A text based description of the Assessment Triggers objective within the simulation; typically describing its purpose and intended adaptive function.

Type: ACTSim is designed to support two types of Assessments: Formative and Summative (as described in previously in this Thesis). The Type property allows an author to select from one of these two options.

Execution: The Execution property is associated with Formative Assessments. It allows an author to select from two options; Instant and Cumulative. An Instant Formative Assessment results in the Assessment Question and Answer of the Dialogue Elements in which the Trigger has been placed only being included in the Formative Assessment Trigger; i.e. a single Formative Assessment Question is presented to the learner followed by the coupled Answer. A Cumulative Formative Assessment results in all Assessment Question and Answers from previous Dialogue Elements that a learner has previously accessed to that point of the simulation, being included in Formative Assessment Trigger; i.e. a learner is presented a series of Assessment Questions and Answers which have been accumulated by the learner based on the path they have followed within the Dialogue Model.

Timing: The Timing property allows an author to indicate when the Assessment Trigger should fire. There are three Timing options; Before, After and Highlight. The Before option indicates that an Assessment Trigger should fire before an author is presented with the content (Statement and Response URI's or media files) of a Dialogue Element that they have selected to enter within a simulation. The After option indicates that an Assessment Trigger should fire after an author has been presented with a Dialogue Element's content. The Highlight option indicates that the Assessment does not automatically fire but instead appear as an option within the simulation screen that an author may or may not select.

Assessment URI: The Assessment URI property is associated with Summative Assessments. It is used to indicate the file location of the Summative Assessment that is to be delivered in the simulation.

Reflection Trigger Component

Reflection Triggers allow an author to indicate the location of Reflection instances that could be delivered to a learner within a simulation. By tagging a Reflection Trigger an author can personalise a Reflection Triggers firing action. Reflection Trigger Components are less complex than Assessment and Feedback Triggers so do not require an author to include a

name or general description within their property values. The only property specific to a Reflection Trigger Component is a Prompt; this is the cue delivered to a learner which directs the path of their Reflection if the Trigger is fired in the personalised simulation.

5.3 Technological Architecture View

ACTSim has been implemented with several different open source technologies. The primary development platform used was Eclipse (Eclipse 2005). Eclipse is a software development environment that comprises of an Integrated Development Environment (IDE) (Arthorne and Laffra 2004) and extensible plug-in system. The Graphical Modelling Framework (GMF) plug-in (Ehrig, Ermel et al. 2005), part of the Eclipse Modelling Project, is central to ACTSims implementation. It provides a generative bridge between the Eclipse Modelling Framework (EMF) and Graphical Editing Framework (GEF) (Budinsky, Steinberg et al. 2003). GMF allows a model driven approach to generating fully functional graphical editors. The Eclipse GMF environment was selected for development for the following reasons:

- GMF allows rapid prototyping of complex graphical tooling systems suited to the development process of this research.
- A further Eclipse plug-in, the Rich Client Platform (RCP) plug-in, was employed in development to allow applications to be easily and effectively configured for client (author) deployment.
- Generated Graphical Editors incorporated several aspects of functionality that were required by ACTSim. This included an interactive graphical editor space for adding, moving and deleting graph based components. Other functionality included navigational aids such as an interactive map, zoom functionality and a list view of all Dialogue Elements.

5.3.1 GMF Development

GMF projects are dependent on developing and generating several different models. Figure 5.3 displays an overview of the main components used in creating a GMF project. The first model to be developed is the Domain Model (ecore model). This model captures the underlying logic upon which the Graphical Editor will operate. The Domain Model defines the different components that will appear in the Graphical Editor and their relationships with one another. The next model developed is the Graphical Definition Model (gmfgraph model). This model captures information that is related to the graphical elements of the Graphical

Editor. At this point of development it does not have any direct connection with the Domain Model. The final model to be developed is the Tooling Definition Model (gmftool model) which describes the graphical tools such as the palette, menu bar and tool bar; again this model has no ‘knowledge’ of the two models that have already been developed.

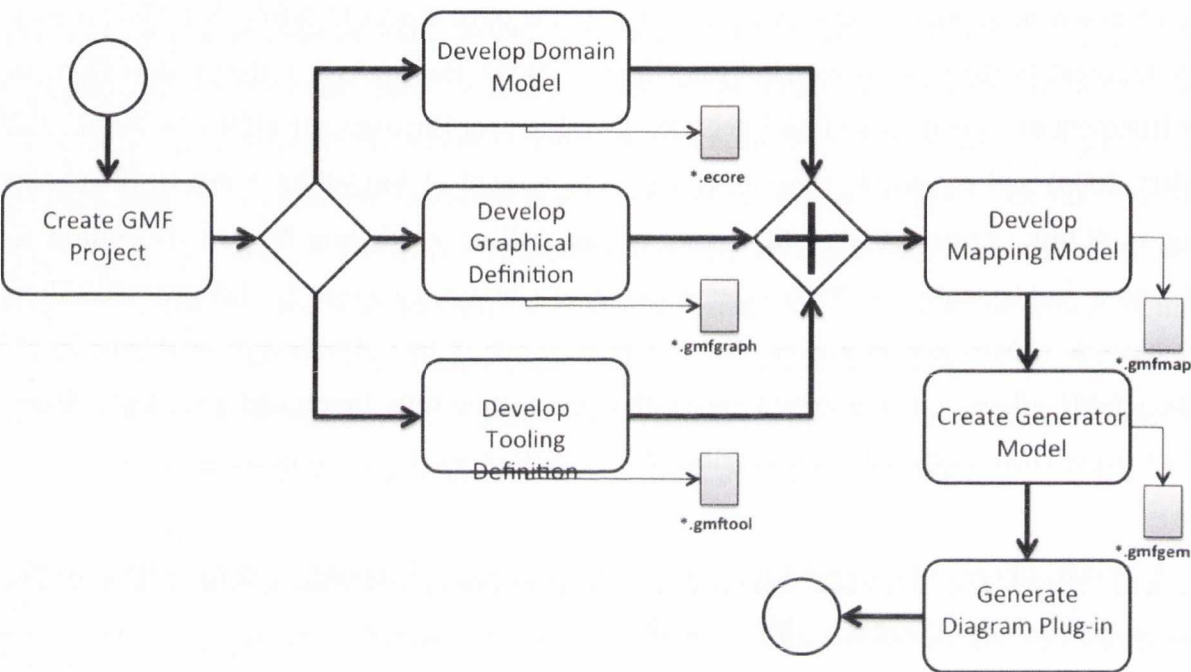


Figure 5.3: GMF Overview

Once each of these models have been created, defined and edited they are combined by generating a Mapping Model (gmfmmap model). The mapping model links the Domain Model, Graphical Definition Model and Tooling Definition Model to one another. The initial instance of the Mapping Model defines a basic mapping which will typically need to be edited so the appropriate and correct mappings are defined.

The next step in development is to create a Generator Model (gmfgem model). This model is generated using the Mapping Model and contains a description of the implementation details of the GMF Graphical Editor. Details of this model can again be edited and developed to suit the uniqueness of the specific Graphical Editor that is being created. The code for the basic Graphical Editor is generated using the Generator Model. The code can be updated, amended and augmented to create the final Graphical Editor. Typically the GMF editor acts as an Eclipse plug-in but can also be developed as a standalone application (independent of Eclipse) using the RCP plug-in.

5.3.2 ACTSim GMF Development

This section describes the different models that were used to develop ACTSim with Eclipse and its GMF Modelling Project plug-in.

ACTSim Domain Model

GMF allow several approaches to developing a Domain Model. This research employed the Annotated Java approach as it was found to be user friendly and effectively captured all components and relationships to be included in the implementation. Each component that was to be included in the Graphical Editor was described using a specialised notation of Annotated Java. Figure 5.4 displays the annotation used to describe the Canvas component, its properties and their types. For example, `String getName()` describes the Name property of the Canvas component. `EList <String> get Roles()` denotes that the Roles property is stored as a List of Strings (EList is simply an EMF extension of the Java List primitive). The final entry in the Canvas Java Annotation indicates that there is a Dialogue Element component and that the Canvas component acts as its container. Annotated Java files were developed for each of the ACTSim components, each describing their associated properties and relationships, see Appendix III – Annotated Java.


```

package actsim;
import org.eclipse.emf.common.util.EList;
import org.eclipse.emf.ecore.EObject;
/**
 * @model
 */
public interface Canvas extends EObject {
    /**
     * @model
     */
    String getName();
    /**
     * @model
     */
    String getIdentifier();
    /**
     * @model
     */
    String getGeneral();
    /**
     * @model
     */
    String getPedagogicalApproach();
    /**
     * @model
     */
    EList<String> getRoles();
    /**
     * @model
     */
    EList<String> getSubjects();
    /**
     * @model
     */
    EList<String> getCategories();
    /**
     * @model
     */
    EList<String> getLearningVerb();
    /**
     * @model
     */
    EList<String> getLearningNoun();
    /**
     * @model
     */
    EList<String> getLearningOutcome();
    /**
     * @model
     */
    String getFileLocation_user_setting();
    /**
     * @model type="DialogueElement" containment="true"
     */
    EList<DialogueElement> getDialogueElement();
}

```

Figure 5.4: Annotated Java for Canvas Component

The ACTSim Domain Model was generated using the Annotated Java files that were created for each component. Components included the Canvas, Dialogue Element, Assessment Trigger, Feedback Trigger and Reflection Trigger. Figure 5.5 displays a screen capture of the ACTSim Domain Model with the Assessment and Dialogue Element component properties

expanded. The GMF development environment also allows a graphical display of the Domain Model to be generated (ecore_diagram model) which is displayed in Figure 5.6.

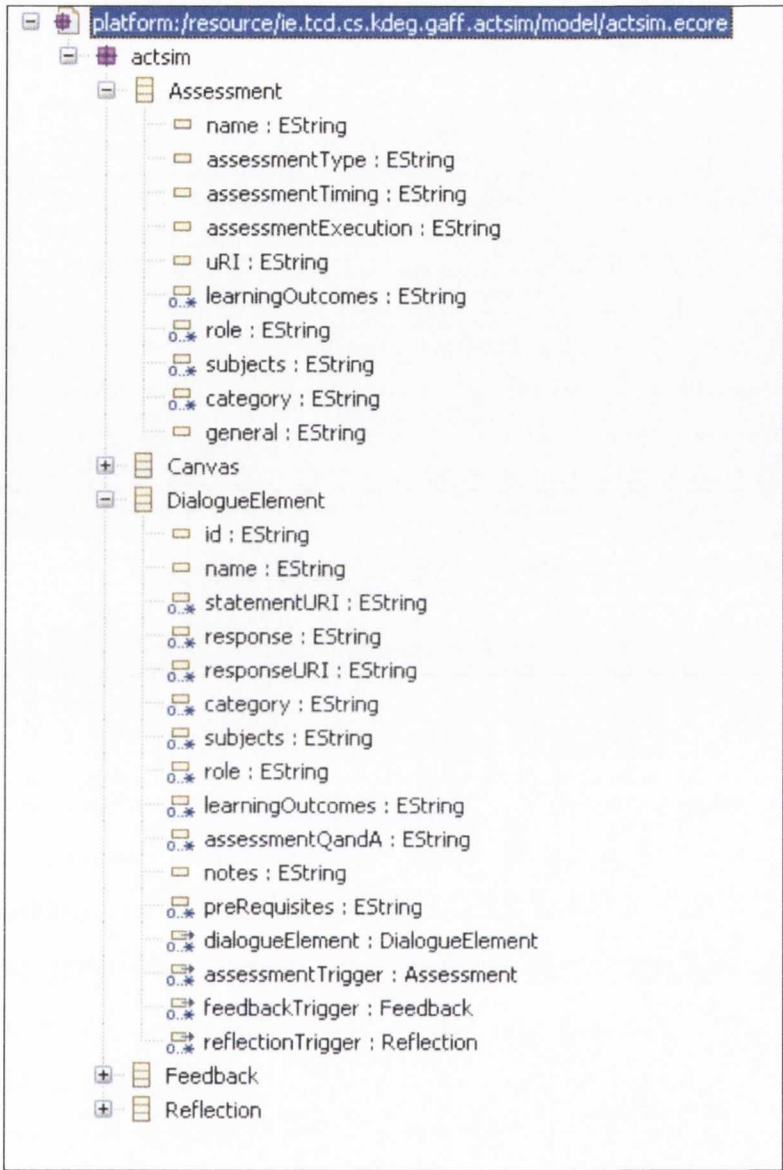


Figure 5.5: ACTSim Domain Model

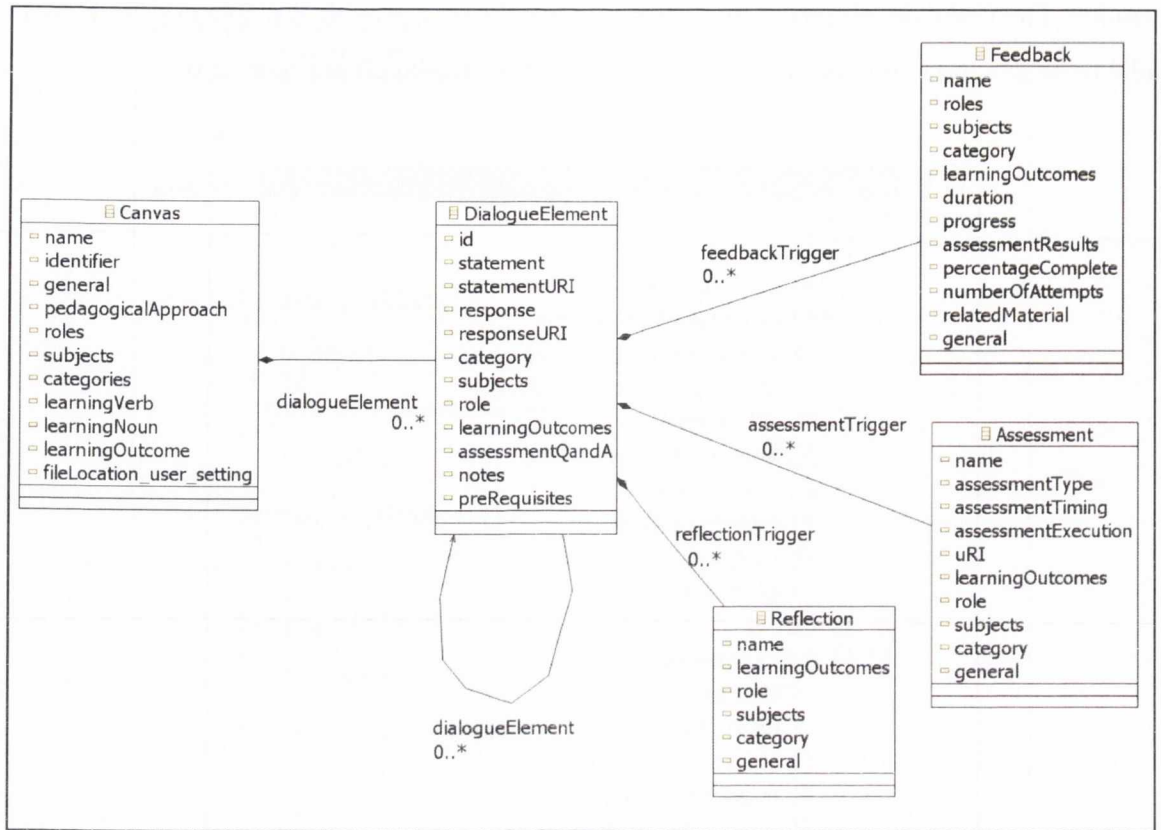


Figure 5.6: Diagram of ACTSim Domain Model

Figure 5.6 clearly shows each component, their relationships to one another and their associated properties. The Dialogue Element component is contained by the Canvas component and the three Trigger components are contained by the Dialogue Element component; the Domain model indicates that a containment relationship exists between the Triggers and Dialogue Element components but does not indicate how this relationship will be implemented. The Connection element can also be viewed as a self pointing relationship from the Dialogue Element; this indicated that a Connection can be made from Dialogue Element instances to other Dialogue Element instances.

ACTSim Graphical Definition Model

The ACTSim Graphical Definition Model was implemented to have at its root as the Canvas component as seen in Figure 5.7. Contained within the model are the Figure Gallery, a number of node definitions, a connection definition, a compartment definition and several Diagram label definitions. The Figure Gallery contains graphical descriptions of all nodes and connections that will exist in the Graphical Editor. With the exception of the Connection element, used to connect Dialogue Elements to one another, each Figure element follows a

similar pattern. The Figure Descriptors each contain a child element which describes the shape of the Figure which in turn contains a child Label Figure element. The Label Figure denotes which of the components properties should be displayed in the graphical representation. The Dialogue Element Figure Descriptor, for example, denotes that its graphical representation will be a Rounded Rectangle which will display an Id and Name property; the Child Access elements `getFigureDialogueElementIdFigure` and `getFigureDialogueElementNameFigure` respectively.

The Figure Gallery creates the graphical definition of the nodes and connections. There is still no relationship between this graphical definition and how they will be implemented. The Node and Connection definitions contained within the Graphical Definition Model (Node Assessment, Node DialogueElement, Node Feedback, Node Reflection and Connection DialogueElementDialogueElement) acknowledge that there will exist instances of these element types within the Domain Model and they will be mapped appropriately in the Mapping Model. The Node definitions are each designated Figures from the Figure Gallery within the Graphical Definition Model; for example, Node Dialogue Element is assigned the DialogueElement Figure.

The Diagram Label elements are also defined within the Graphical Definition model. These allow each label that appears in a Figure to be associated with an image or icon. Trigger components later have their Diagram Labels edited so the Label will not appear in the Figure but its associated icon will. This allows Trigger Figures in the Graphical Editor to be represented as icons without any property text appearing. The final element contained in the Graphical Definition Model is the Compartment for Triggers which is mapped to the Dialogue Element Figure in the Mapping Model.

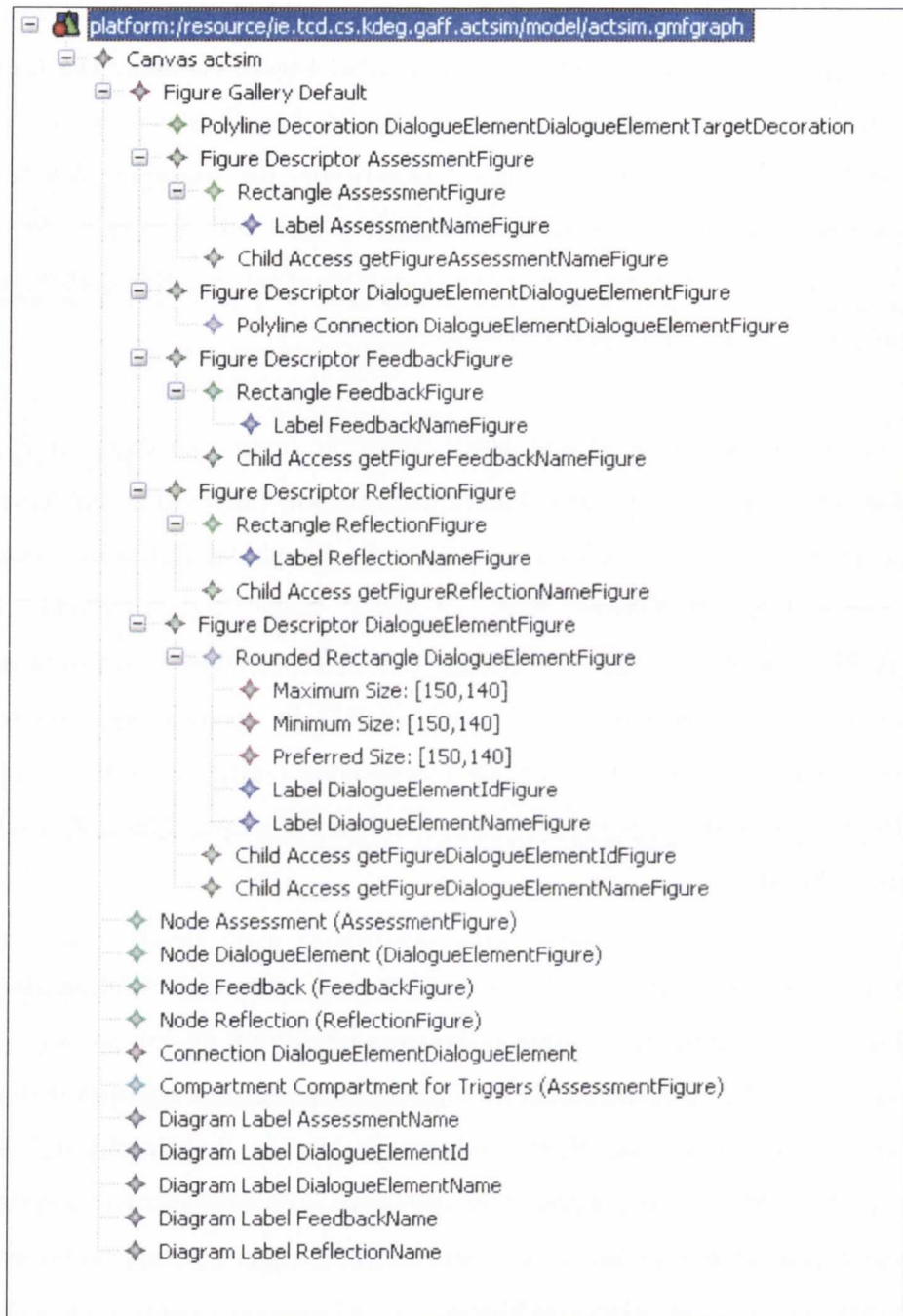


Figure 5.7: ACTSim Graphical Definition Model

The ACTSim Graphical Definition Model simply defines the building blocks of the Graphical Editor. It defines four types of nodes, one connector and one compartment. It also defines how each of these components will be visually represented. The Graphical Definition Model does however not ‘*know*’ how these components work together. It is the Domain Model that defines the behaviour of the Graphical Editor.

ACTSim Tooling Definition Model

The ACTSim Tooling Definition Model defines the palette functions within the Graphical editor. The ACTSim Tooling Model, as seen in Figure 5.8, contains two groupings. The first Tool Group accommodates functions associated with the Dialogue Elements; specifically creating Dialogue Elements and creating connections between them. The second Tool Group accommodates the creation of Triggers; specifically creating Assessment, Feedback and Reflection Triggers. Each of the Creation Tool function Elements contains a Bundle Image element which associates the palette function button with a specified image.



Figure 5.8: ACTSim Tooling Definition Model

ACTSim Mapping Model

The Mapping Model combines and maps together the Domain Model, Graphical Definition Model and Tooling Definition Model. The ACTSim Mapping Model, as seen in Figure 5.9, has contained in its root (Mapping element) three child elements; a Top Node Reference; a Link Mapping; and Canvas Mapping. The Mapping Model also includes copies of the three models it is mapping, as seen at the base of the Mapping Model, to allow easy reference and editing.

The Canvas Mapping element describes the Canvas component mapping which is the main containment component of the graphical editor. This requires very little description within the Mapping Model as there are no graphical or tool mappings to be defined. The Link Mapping

element is also very simple as it only describes the connection that can be made between Dialogue Elements and is simply mapped to the correct tooling definition.

The Top Node Reference is the most complex aspect of the Mapping Model. It describes the Dialogue Elements node within the Graphical Editor. The Top Node Reference contains a Node Mapping which maps the Dialogue Element node of the Domain Model to its graphical representation in the Graphical Definition Model. The Node Mapping in turn contains two Feature Label Mapping elements, three Child References and a Compartment Mapping. It is the structuring and mapping of these elements which is critical to correctly defining the functionality of the Dialogue Element node and in turn the behaviour and functionality of the three types of Trigger components. The Feature Label Mappings links the property values of Dialogue Elements, Id and Name (the Name is used as Statement property for ease of implementation within GMF), that are to be displayed within the Graphical Editor representation with appropriate Diagram Labels. This results in each Dialogue Element displaying their Id and Statement values within their graphical representation along with associated icons.

The Compartment Mapping element contained by the Node Mapping maps the Compartment defined in the Graphical Definition Model to the Dialogue Element node; this creates the Dialogue Element compartment which will be used to contain Triggers. The Child Reference elements contained by the Dialogue Element Node Mapping each define the mapping of the three Triggers. The Child Reference elements each contain a Node Mapping and a Feature Label Mapping. These two elements map the graphical components to the corresponding nodes and the Feature Label Mappings are used to define the icon to appear in each graphical representation. The Child Reference elements Compartment property was mapped to the Compartment Mapping element contained by the Dialogue Element Node Mapping to indicate it was to be used as the containment feature for all Triggers. The Mapping Model was used to generate the Generator Model.

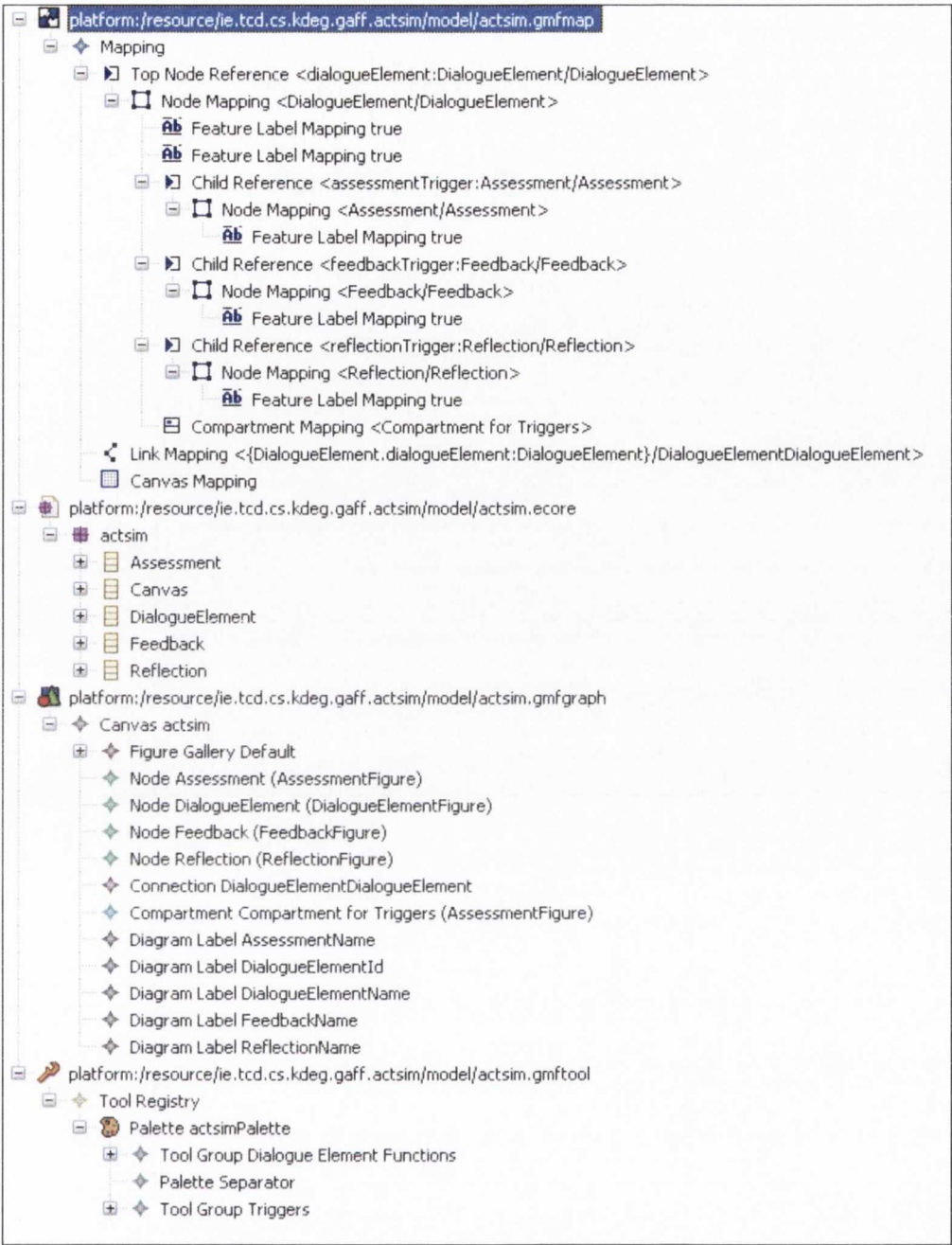


Figure 5.9: ACTSim Mapping Model

ACTSim Generator Model

The ACTSim Generator Model, as seen in Figure 5.10, is used to generate the Graphical Editor code. It is structured in a similar manner to the Mapping Model but each element within the model contains much more detail as there are many more properties associated with each element. The Generator Model allows the detail of the generated code to be edited; for the purpose of the ACTSim authoring very little of this model was altered. The most significant edit was to update the Gen Compartment element so Triggers added to the

Dialogue Element compartment would not be fixed in a list format but instead would be moveable within the compartment and to other Dialogue Elements.



Figure 5.10: ACTSim Generator Model

5.3.3 ACTSim Graphical Editor

The initial and basic Graphical Editor generated by the Generator Model included the following important functionality:

Basic Functionality: The generated Graphical Editor implements the basic functionality that would be expected in an application of this kind. This includes file operations such as Open, Close, Save and New. The generated application also allows an author to undo and redo

completed operations and implements a warning system to protect data if an author attempts to close the Graphical Editor without saving a file.

Dialogue Space: The Canvas component previously described in this chapter is realised as ACTSims Dialogue Space. This is the area where an author can build and design their Dialogue Model. Within this space an author can add, connect, move and delete Dialogue Elements. The configuration of the GMF models also result in the author's ability to add, move and delete Triggers within Dialogue Element compartments.

Palette: Included in the ACTSim Graphical Editor is a Palette that contains the most significant functionality used for constructing a Dialogue Model. These are functions that were defined by the Tool Model which are divided into two groups. The first grouping defined allows an author add and connect Dialogue Elements. The second grouping allows an author to add any of the three Trigger types. The default Palette also includes a third grouping which is comprised of zoom functionality and a mouse cursor button; the mouse cursor button allows an author to opt out of using a function after it has been selected and return to the default mouse cursor.

Dialogue Element: The Graphical Editor provides the desired structure of a Dialogue Element; this includes a graphical display of the associated Id and Statement properties of each Dialogue Element within a rounded rectangle which includes a compartment for Triggers.

Triggers: The Graphical Editor allows the three different Trigger types to be added, deleted and moved within Dialogue Element compartments. Furthermore, Triggers cannot be directly moved to the Dialogue Space but can be transferred from one Dialogue Element compartment to another Dialogue Element compartment. Triggers are graphically displayed as icons within compartments and display no property information.

Navigational Aids: Several of the required ACTSim navigational aids were included in the initial basic Graphical Editor. The generated application contains a map, zoom functionality, a list feature and an arrange option. The map is contained within a separate window and allows a user to manipulate the view in the Dialogue Space through a drag and drop interface.

The magnification of the Dialogue Space can be altered through several means of interaction. The Graphical Editor provides a list view of all Dialogue Elements that have been created in a Dialogue Model. Finally, an arrange operation which allows users to automatically reorganise Dialogue Models is also implemented.

Additional Functionality: The Graphical Editor generated using the GMF and RCP plug-ins also included several other useful functions and features:

- All windows within the application are moveable and resizable.
- The Graphical Editor allows multiple titles to be opened at one time with tab functionality.
- Included in the Graphical Editor application is a feature that allows an author to add a 'Note' to the Dialogue Space. This can be used by an author to record observations or comments and attach them directly to the Dialogue Model.
- Another feature found to be useful was a colour fill function. This allows an author to group Dialogue Elements by assigning them a particular fill colour. This feature was also used to distinguish the main stem of the Dialogue Model from the other Dialogue Elements and branches of the Dialogue Model. The Highlight, Search and Validate functions, described in later sections of this chapter, were not affected by this feature as they all manipulate the line colour of Dialogue Elements.
- The Graphical Editor also provides several shortcuts to executing operations within the application. For example, if the mouse cursor is rested in a empty area within the Dialogue Space a prompt will appear at that point in the Dialogue Space allowing the user to add a new Dialogue Element; a similar short cut is implemented for Triggers. The Graphical Editor also allows users to connect Dialogue Elements without needing to use the Palette. This can be executed by resting the mouse cursor on an existing Dialogue Element. After a short time a small arrow icon appears as a prompt appended to the Dialogue Element. This icon can be selected to create a connection by dragging new a Connector from the Dialogue Element to another Dialogue Element to complete the graphical edge.

5.3.4 ACTSim Authoring Tool

Figure 5.11 displays a screen capture of the ACTSim Authoring Tool which includes all the functionality described in the previous sections; the screen capture displays ACTSim with an open Dialogue Model; a video screen capture of the authoring tool being used can also be found online (Gaffney 2013). Typically GMF Graphical Editors act as Eclipse plug-ins and simply extend the Eclipse IDE. The RCP plug-in allowed the ACTSim Graphical Editor to be removed from the Eclipse IDE and act as an independent application. RCP allowed the application to be deployed as a '*product*' contained within an independent filing system which is compatible with different operating systems. The only local requirements for running the ACTSim authoring tool is a Java installation and networking capabilities.

While the GMF Graphical Editor implemented the basic graphical interface there were a number of additional implementation issues that needed to be addressed to fully meet the previously outlined requirements. The following section address the implementation of these requirements.

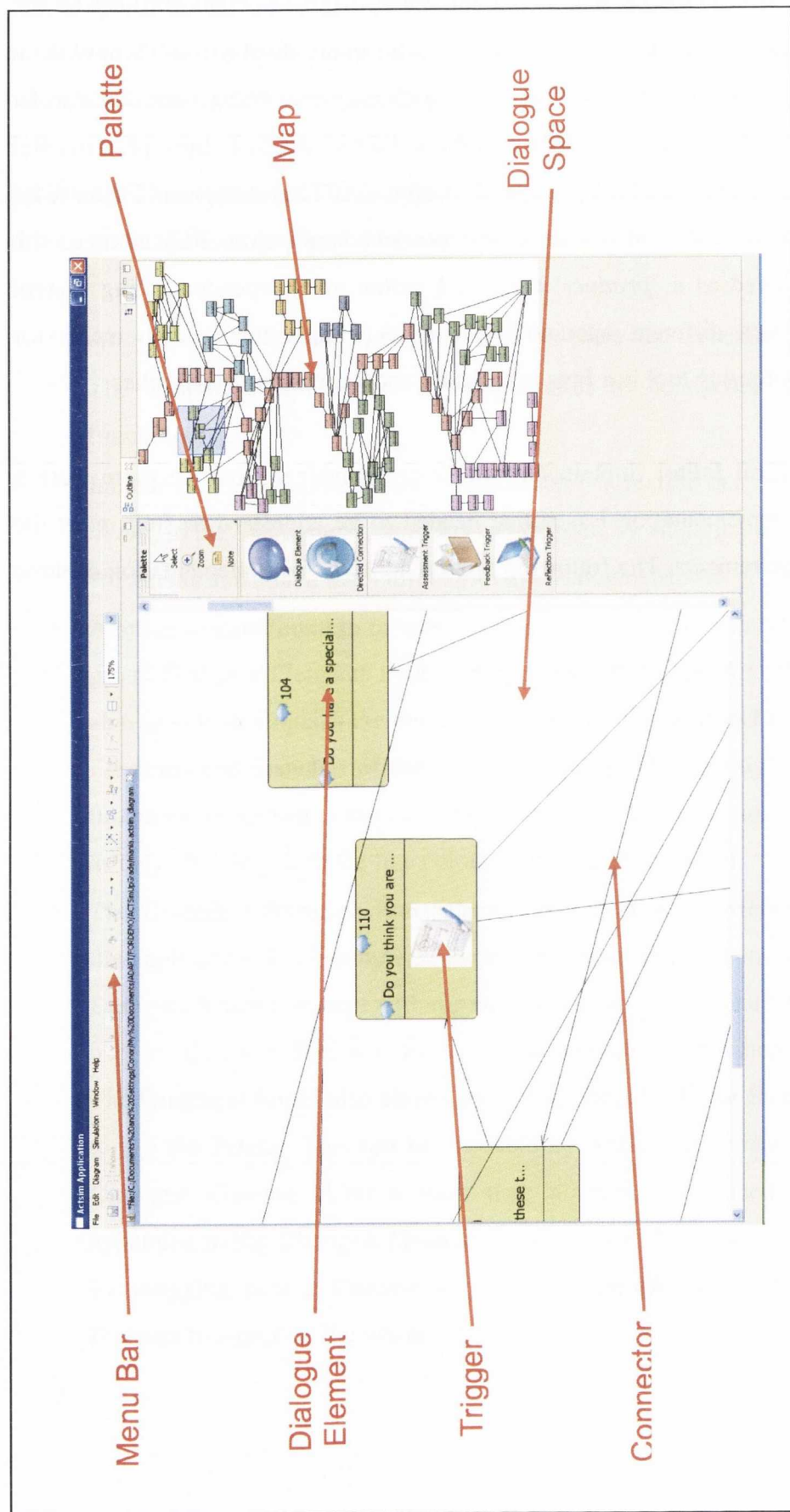


Figure 5.11: ACTSim Authoring Tool

5.3.5 Further Functionality Implementation

The generated Graphical Editor created the basic ACTSim application which addressed a number of requirements. There were, however, some requirements that the basic Graphical Editor did not implement. This section details the implementation of functionality that addressed the remaining requirements. It was completed by editing and augmenting the code generated by the Generator Model for the basic Graphical Editor.

Accessing, Editing and Updating Component Properties

The generated Graphical Editor does not allow authors to access, edit or update the properties of the main architectural components in a suitable manner. The authoring tool required that an author should be allowed to interact with the simulation properties (stored in the Canvas component), the Dialogue Element properties and all Trigger properties in separate customised interfaces. Customised interfaces for each component were developed with the Standard Widget Toolkit (SWT) (Northover and Wilson 2004); an open source Eclipse plug-in. SWT is a Java based toolkit designed to provide efficient portable access to user-interface facilities of an operating system on which it is implemented. This toolkit allowed the implementation of popup windows that were designed for each architectural component. The design and implementation of each of these popup windows is described later in this chapter.

There were, however, three issues to be addressed before the SWT popup windows were implemented. The popup windows needed to be accessible within the ACTSim authoring tool; they needed to be populated with the relevant data; and if edited the properties of the simulation title needed to be updated.

Access

Dialogue Elements and Triggers components each needed to be accessible by double clicking on their Dialogue Model instances. This was achieved by editing each components generated EditPart. A GMF EditPart describes a components graphical representation interface including its behaviour. Double click functionality was introduced by editing the DefalutEditPolicies. Once a Dialogue Element or Trigger model element was double clicked its associated developed EditPolicy was executed; for each popup window an EditPolicy was created that stored the accessing functions of the SWT implementation, for example a DialogueElementEditPolicy was created for the Dialogue Element component. An EditPolicy

can be thought of as the programme that is called when an author attempts to access a component's properties. Accessing the Canvas component properties was implemented in a similar manner except the calling function was accessed through a drop down menu rather than double clicking the Canvas component.

Populate

The GMF Graphical Editor interprets the graphical model and underlying data model as two separate models that are working in tandem. An author does not directly access a components data; instead the data is extracted from the data model and presented to an author within a components interface. To achieve this, an instance of the data model is created in the program and populated with a components data. In order to populate each component interface with the relevant data a components EditPolicy retrieves the graphical element that is currently under selection (determined by the double click execution); components graphical elements are called Graphical EditParts. The EditPolicy then creates an underlying model element of the correct type to be populated through the Graphical EditPart. The created model element is assigned the values of the Graphical EditParts associated underlying model element; this is all the data that is stored by the components underlying model. All data from the created model element within the program can then be accessed and is used to populate the SWT popup window.

Update

The data that is being viewed by an author in the popup window is not a direct access to a simulations model data. If the data in the window has been edited or updated and an author wants to save it to the simulations model a separate command is called from the EditPolicy. The command, for example `ChangeDialogueElementPropCommand`, is an extension of the `AbstractTransatctionalCommand` and takes all the component properties read from the SWT interface as its arguments. The command ascertains the correct model instance to update through the selected graphical element and updates the correct underlying model elements properties with the new values that were taken as arguments. Once a save is executed the SWT interface closes, with the data model and graphical representation model updated instantly.

Component Property Interfaces

This section describes the implementation of each of the component property popup windows. Popup windows were developed for Simulations Properties (stored in the Canvas component), Dialogue Elements and all three Trigger types. The component property interfaces were initially storyboarded and have evolved through several different implementations after numerous design evaluations; the design evaluations process is described in more detail in the following chapter. Where possible, similar designs between different interfaces have been employed to give the authoring tool an accustomed feel and author's a sense of continuity within the application.

The SWT popup windows were implemented so, when opened, the rest of the ACTSim application freezes and cannot be accessed until the popup window is closed. This approach insures that multiple interfaces for a single element instance cannot be opened by an author and that the data displayed in each SWT popup windows will remain consistent with the data stored in the underlying simulation model.

The following sections describe each component's customised SWT popup window interface. The screen captures that have been employed in these sections have been taken from a title developed to aid the training of medical students.

Scenario Properties Interface

The Scenario Properties popup window, as seen in Figure 5.12, captures the properties that describe the overall characteristics of the simulation being developed. The popup window is divided into three sections. The first, the largest area to the top left of the window, contains the Name and identifier of the simulation. This section also contains a text box which details a simulation description and its Pedagogical Approach. The second section of the popup window, the smallest section at the base of the window, contains two buttons; 'Save' and 'Cancel'. These buttons are consistent with all other popup windows and allow an author to update simulation properties or cancel without saving. The third section of the interface, to the right of the popup window, contains six buttons. Each of these buttons opens another popup window which contains the scenario properties indicated by the button name. These are the interfaces that allow an author to create and edit a simulation's tagging properties.

There is a separate popup window for the Learning Verbs, Learning Nouns, Learning Outcomes, Roles, Subjects and Categories. With the exception of the Learning Outcomes each of the popup windows are implemented in a similar manner. An author accesses these respective interfaces by selecting the relevant button. Once any of these popup windows have been opened the Scenario Properties Interface freezes and cannot be accessed until the newly opened SWT popup window has been closed.

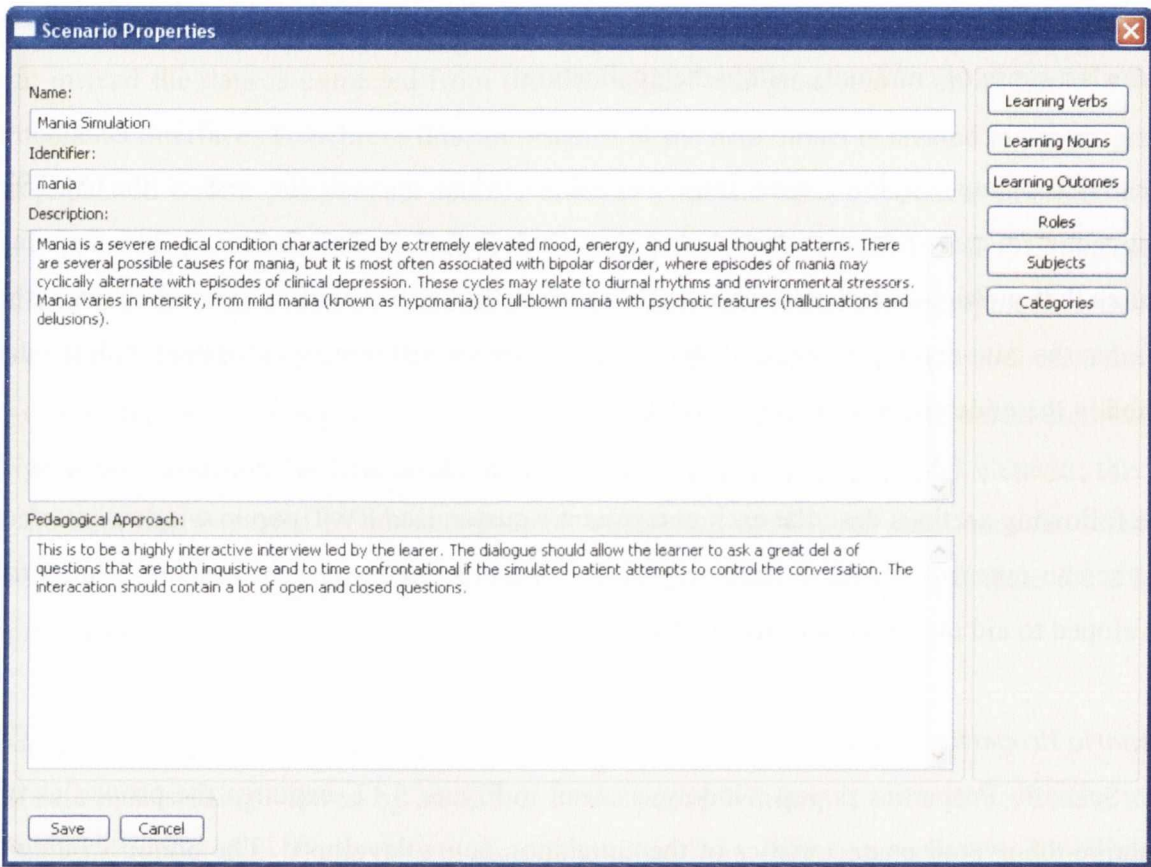


Figure 5.12: Scenario Properties Interface

The Learning Verbs popup window, as seen in Figure 5.13, is similarly implemented to the Learning Nouns, Roles, Subjects and Categories popup windows; see Appendix IV – Component Interfaces. They each contain three sections. The first is a large section to the left of the window which contains the list of properties, in this case the simulations Learning Verbs. The second section to the right of the list displays the details of the selected value in the list section; this includes the name of the property and an associated brief description. The text boxes in this section are empty if no item has been selected from the list of properties. This section of the popup window also allows authors to add update and delete values in the list section using the four buttons found near the base of the section; ‘Update’, ‘Add’,

'Delete' and 'Clear'. The Clear button allows an author to clear the text from the Name and Description text boxes and deselect the highlighted value from the list of properties placing the popup window in state for accepting new input. The third section at the base of the window contains the Save and Cancel buttons.

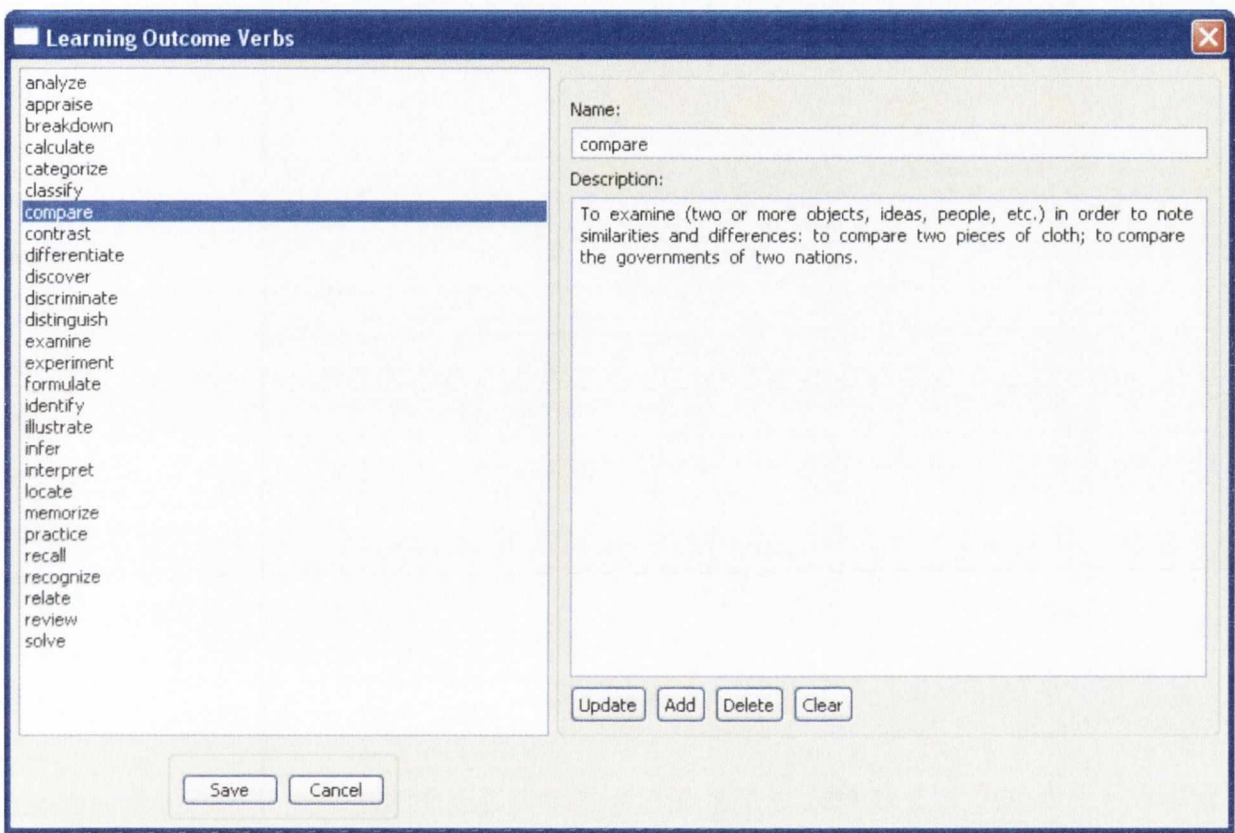


Figure 5.13: Learning Verbs Interface

The Learning Outcomes interface, as seen in Figure 5.14, operates in a slightly different manner. Instead of having a name text box to input a string there are two lists which contain a Learning Verb and a Learning Noun; instances of each are stored accessed in their respective interfaces. To create a Learning Outcome an author selects the appropriate verb and noun from the lists, enters a description and saves the Learning Outcome. When a Learning Outcome is selected from the list of values the corresponding verb and noun are highlighted in the verb and noun lists.

Learning Outcomes

breakdown if the patient has any psychotic symptoms

Explore the problem a patient presents with using appropriate interview techniques

Elicit the various symptoms that the patient presents with which confirm the diagnosis of a manic phase of Bipolar Affective disorder.

Explore the risk of self harm/suicide with the patient

Check the patient's mood and associated symptoms

Check if the patient has any associated anxiety symptoms

Explore how to challenge the false beliefs of a patient

Explore the level of insight the patient has in relation to their presentation and the detrimental effect the symptoms are having on functioning

Identify the precipitating factor to this episode of illness and any things the patient has done to feel better

Explore the patient's past psychiatric history and family psychiatric

Discuss different treatment options with the patient and organise

Verb:

Explore

Elicit

Observe

Check

Identify

Discuss

analyse

breakdown

Noun:

the problem a patient presents with using appropriate interview techniques.

the various symptoms that the patient presents with which confirm the diagnosis of a manic phase of Bipolar Affective disorder.

the risk of self harm/suicide with the patient

the patient's mood and associated symptoms

if the patient has any associated anxiety symptoms

if the patient has any psychotic symptoms

how to challenge the false beliefs of a patient

the level of insight the patient has in relation to their presentation and the detrimental effect the symptoms are having on functioning

the precipitating factor to this episode of illness and any things the patient has done to feel better

Description:

Reality testing the patient's fixed false beliefs

Save

Cancel

Update

Add

Delete

Clear

Figure 5.14: Learning Outcomes Interface

Dialogue Element Interface

The Dialogue Element interface, as seen in Figure 5.15, captures the properties that describe each Dialogue Element. The popup window is divided into five sections. The first is the area to the left of the window which contains a Dialogue Elements most basic properties including its Id, Statement, Response and associated media file locations. The ‘Browse’ buttons in this section of the window opens the operating systems standard file browsing interfaces which allow an author to indicate the location of a Dialogue Elements media files once they have been created. The second section, ‘Dialogue Element Descriptors’, is at the top of the window to the right. It contains all the tagging data related to a Dialogue Element. This section contains a list for each of the four tagging properties that are available to the author; Categories, Subjects, Roles and Learning Outcomes. Each list is populated with all of the values that have been created by an author in the Scenario Properties interface. A value, or set of values, has been associated with a Dialogue Element if it is highlighted in the displayed lists. Values are selected within each list using a mouse cursor. An author can select several values from a single list by holding the Ctrl keyboard button and left clicking the values with the mouse cursor.

Below the Dialogue Element Descriptors section are the three remaining sections of the Dialogue Element interface. To the left of these three is the ‘Assessment’ section which contains two text boxes used to display and edit a Dialogue Elements Formative Assessment Question and Answer. To the right of the Assessment section is the ‘Notes/Directions’ section which contain a text box allowing an author to detail directions or notes related to a Dialogue Element. The Final section, below the Notes/Directions section contains the Save and Cancel button.

Dialogue Element Properties

Dialogue Element ID:
127

Statement Transcript:

I can see you are obviously very happy with John and maybe we could talk some more about him later. For now, however I was hoping I could ask you about how you've been coping recently with everything?

Statement Media File Location:

E:\Mania Interview\Windows Media 512K\q127_New272.wmv

Browse

Response Transcript:

I'm fine, great, wonderful. It's all good, like I say, busy as a bee!

Response Media File Location:

Browse

Dialogue Element Descriptors:

Categories

Initial Statement

Orientation to/Permission for consult

Open question

closed question

Clarifying/Checking Question

Invitation to discuss other options

Summarising Statement

Advice

Validation / Legitimation

Treatment plan

Advice and request to continue with

Subjects

Greeting

No Category

Psychotic Symptoms - Delusions

Psychotic Symptoms - Hallucinations

Appetite/Weight

Treatment/Options

Mood Symptoms (Depression/Elation)

Energy/Concentration

Interest/Enjoyment

Social history

Medical/ Medication History

Hopelessness and Suicide

Sleep

Mood Symptoms (Anxiety)

Mood Symptoms (Psychotic features)

Family History

Roles

Doctor

Consultant

Nurse

Learning Outcomes

breakdown if the patient has any psy

Explore the problem a patient presen

Ellicit the various symptoms that the p

Explore the risk of self harm/suicide w

Check the patient's mood and associa

Explore if the patient has any psycho

Explore how to challenge the false be

Explore the level of insight the patient

Identify the precipitating factor to thi

Explore the patient's past psychiatric

Discuss different treatment options w

Assessment

Question:

Answer:

Notes/Directions:

Save

Cancel

Figure 5.15: Dialogue Element Interface

Assessment Trigger Interface

The Assessment Trigger interface, as seen in Figure 5.16, captures the properties that describe each Assessment Trigger. The popup window is divided into three sections. The first is the area to the left of the window which contains an Assessment Trigger's main properties including its Purpose, Assessment Type, Assessment Timing, Execution and Assessment URI. The Purpose (General Description) of the Assessment Trigger incorporates a text box. The values for the Assessment Type, Assessment Timing and Execution properties are selected from drop down lists. The options available to the author in each drop down menu are as follows:

- **Assessment Type:** Formative; Summative.
- **Assessment Timing:** Before; After; Highlight.
- **Execution:** Instant; Cumulative

The Browse button in this section of the window opens an operating systems standard file browsing interface which allows an author to indicate the location of an Assessment Trigger's Summative file location. The file location is inserted into the Assessment URI property box when saved.

Assessment Trigger Properties

Purpose:

This Assessment is targeted at learners in a Doctor role. It's Summative questions are designed specifically for the Doctor role.

Assessment Type:

Summative

Assessment Timing:

Before

Execution:

Instant

Assessment URI:

C:\Medicine\URI\Assessment\test_one.txt

Assessment Descriptors:

Categories

- Initial Statement
- Orientation to/Permission for c
- Open question
- closed question
- Clarifying / Checking Question
- Invitation to discuss other opt
- Summarising Statement
- Advice
- Validation / Legitimation
- Treatment plan
- Advice and request to continu

Subjects

- Greeting
- No Category
- Psychotic Symptoms - Delusio
- Psychotic Symptoms - Hallucina
- Appetite/ Weight
- Treatment/ Options
- Mood Symptoms (Depression/E
- Energy/ Concentration
- Interest/Enjoyment
- Social history
- Medical/ Medication History
- Hopellessness and Suicide
- Sleep
- Mood Symptoms (Anxiety/)
- Mood Symptoms (Psychotic fe
- Family History

Roles

- Doctor
- Consultant
- Nurse

Learning Outcomes

- breakdown if the patient has .
- Explore the problem a patient
- Elicit the various symptoms th
- Explore the risk of self harm/s
- Check the patient's mood and
- Check if the patient has any z
- Explore if the patient has any
- Explore how to challenge the
- Explore the level of insight th
- Identify the precipitating fact
- Explore the patients past psy
- Discuss different treatment of

Save

Cancel

Figure 5.16: Assessment Trigger Properties Interface

The second section of the Assessment Trigger interface, to the right of section that contains the main properties, is the ‘*Assessment Descriptors*’ section. This section contains all the data related to an Assessment Triggers Tags. It is implemented and operates in similarly to a Dialogue Elements Dialogue Element Descriptors section and allows an author to view and edit the tags associated with an Assessment Trigger. Below the Assessment Descriptors section is the area which contains the save and cancel functions.

Feedback Trigger Interface

The Feedback Trigger interface, as seen in Figure 5.17, captures the properties that describe each Feedback Trigger. This interface is similar to the Assessment Trigger interface and is divided into three sections. The ‘*Feedback Options*’ section contains six check boxes, each associated with the type of feedback that could be presented to a learner: Assessment Results, Duration, Number of Attempts, Percentage Complete, Progress and Timing. By checking a check box an author indicates that the associated type of feedback should be delivered if the Trigger is fired within a simulation. To the right of the Feedback Options section is the ‘*Feedback Descriptors*’ section which stores a Feedback Triggers tagging properties. It is implemented and operates in a similar manner to an Assessment Triggers Assessment Descriptors interface section. Below the Feedback Descriptors section is the section that contains the save and cancel operations.

Feedback Trigger Properties

Feedback Options

☒ Assessment Results
 ☐ Duration
 ☒ Number of Attempts
 ☒ Percentage Complete
 ☒ Progress
 ☐ Timing

Feedback Descriptors:

Categories

Initial Statement
 Orientation to/Permission for consultation
 Open question
 closed question
 Clarifying / Checking Question
 Invitation to discuss other options
 Summarising Statement
 Advice
 Validation / Legitimation
 Treatment plan
 Advice and request to continue with interview

Subjects

Greeting
 No Category
Psychotic Symptoms - Delusions
 Psychotic Symptoms - Hallucinations
 Appetite/ Weight
 Treatment/ Options
 Mood Symptoms (Depression/Elation)
 Energy/ Concentration
 Interest/Enjoyment
Social History
 Medical/ Medication History
 Hopelessness and Suicide
Sleep
Mood Symptoms (Anxiety)
 Mood Symptoms (Psychotic features)
 Family History

Roles

Doctor
 Consultant
 Nurse

Learning Outcomes

breakdown if the patient has any psychotic symptoms
 Explore the problem a patient presents with us
 Elicit the various symptoms that the patient presents
 Explore the risk of self harm/suicide with the patient
 Check the patient's mood and associated symptoms
 Explore if the patient has any associated anxiety
 Explore how to challenge the false beliefs of a patient
 Explore the level of insight the patient has in their condition
 Identify the precipitating factor to this episode
 Explore the patient's past psychiatric history and current symptoms
 Discuss different treatment options with the patient

Save

Cancel

Prompt:

Is this line of questioning helpful in building a relationship with the patient?

Reflection Descriptors:

Categories	Subjects	Roles	Learning Outcomes
Initial Statement Orientation to/Permission for consult Open question closed question Clarifying / Checking Question Invitation to discuss other options Summarising Statement Advice Validation / Legitimation Treatment plan Advice and request to continue with	Greeting No Category Psychotic Symptoms - Delusions Psychotic Symptoms - Hallucinations Appetite/ Weight Treatment/ Options Mood Symptoms (Depression/Elation) Energy/ Concentration Interest/Enjoyment Social history Medical/ Medication History Hopelessness and Suicide Sleep Mood Symptoms (Anxiety) Mood Symptoms (Psychotic features) Family History	Doctor Consultant Nurse	breakdown if the patient has any ps; Explore the problem a patient preser Elicit the various symptoms that the Explore the risk of self harm/suicide. Check the patient's mood and associ. Check if the patient has any associa Explore if the patient has any psychi Explore how to challenge the false bi Explore the level of insight the patie Identify the precipitating factor to th Explore the patients past psychiatric Discuss different treatment options v

Save Cancel

Figure 5.17: Feedback Trigger Properties Interface

Figure 5.18: Reflection Trigger Properties Interfac

Reflection Trigger Interface

The Reflection Trigger interface, as seen in Figure 5.18, captures the properties that describe each Reflection Trigger. This interface is similar to the Assessment Trigger and Feedback Trigger interfaces and is also divided into three sections. To the left of the interface is the first section which contains a large text box which allows an author to enter a Reflection Trigger's Prompt. To the right of this section are the '*Reflection Descriptors*' section, implemented and operated in a similar manner to the previous two Trigger interfaces in the same location, and a section that contains the interfaces save and cancel function.

Dialogue Model Functions

There are a number of functions that were added to the basic ACTSim Graphical editor which operate on the Dialogue Model. These fall into one of two categories; functions that operate on a selected part of the Dialogue model; and functions that operate on the entire Dialogue Model. Both types employ an IAction implementation which operates as either the selected part of the Dialogue Model or the entire Dialogue Model. IAction is part of an Eclipse user interface toolkit that provides classes for developing SWT related operations. The following sections describe implementation of both categories of functions.

Multiple Tagging

The Dialogue Element popup window interface allows an author to tag individual Dialogue Elements. A further functional requirement of ACTSim was that authors should be allowed to tag multiple Dialogue Elements in a single operation. In order to tag multiple Dialogue Elements an author initially selects the Dialogue Elements to be tagged. This, for example, can be executed by an author by holding the Ctrl button on the keyboard and left clicking the appropriate Dialogue Elements with the mouse cursor; such a process is identical to file selection within Windows and other operating systems. To tag the selected Dialogue Elements an author calls the appropriate tagging popup window interface. This can be executed in a number of ways but right clicking one of the selected Dialogue Elements and selecting from a popup menu is the most convenient approach.

Each of the four adaptive tagging properties (Categories, Subjects, Roles and Learning Outcomes) is assigned an individual interface. The popup windows are implemented using SWT and are initially populated with all property tagging values that are available to the

author. These values are read through the selected Dialogue Elements (IStructuredSelection) by finding their graphical container (the relevant Canvas component) and then reading the Canvas's underlying data model properties. The tagging properties are displayed in the SWT popup window in list format, as seen in Figure 5.19. An author can select single or multiple values to be used for tagging, apply them to the selected Dialogue Element or remove the tags from the Dialogue Elements if the Dialogue Elements have been previously tagged.

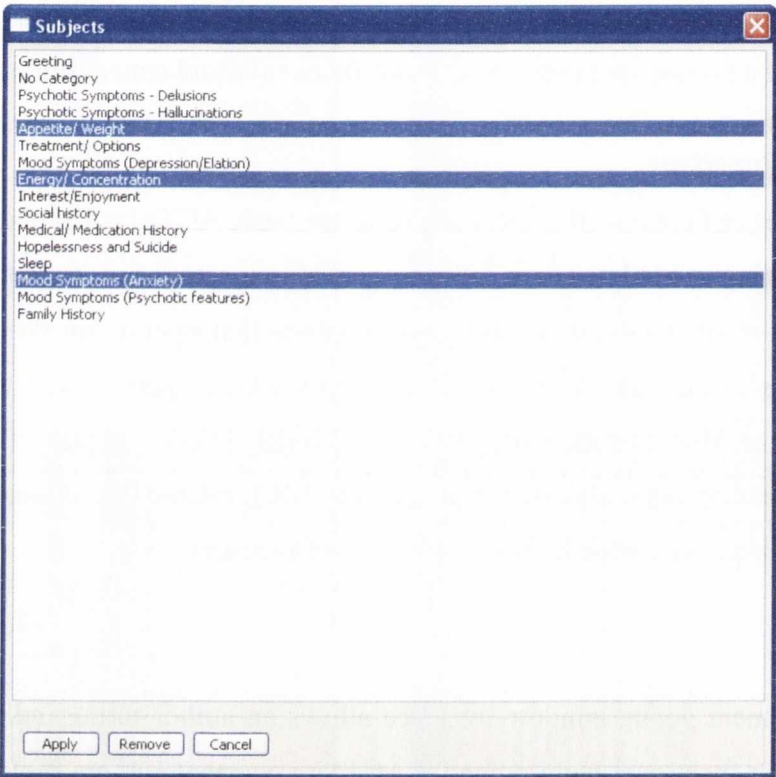


Figure 5.19: Subjects Multiple Tagging Interface

The tagging function operates on the selected Dialogue Elements by iterating through each graphical element of the selection locating its associated underlying model element and adding the new property value. If the property already contains the value, i.e. the Dialogue Element has already been tagged with that value, the operation does nothing and moves to the next Dialogue Element. The remove component of the tagging function operates in a similar manner.

Script

The script function allows authors to sequentially view the Statements and Responses of Dialogue Elements in a selected segment of a Dialogue Model. This allows an author to

inspect a sequence of dialogue to insure it is coherent and intelligible. This function is implemented and executed in a similar manner to the multiple tagging function as both operate on selected Dialogue Elements. Once an author has selected the Dialogue Elements to be scripted the function is called and the author is presented with an SWT popup window, as seen in Figure 5.20. Unlike the multiple tagging function there is no need to access the selections parent Canvas component. The popup window is populated immediately with the Statement and Response of each of the selected Dialogue Elements (IStructuredSelection). The program iterates through the graphical elements locating the underlying model element for each Dialogue Element and reads their associated Statement and Response property values. These string values are written to the popup window with a line separator dividing each Dialogue Element. The selected Dialogue Elements are stored and iterated through in the same order that they are selected by the author. This allows an author to investigate how unconnected Dialogue Elements might appear or how editing connections might affect the end dialogue.

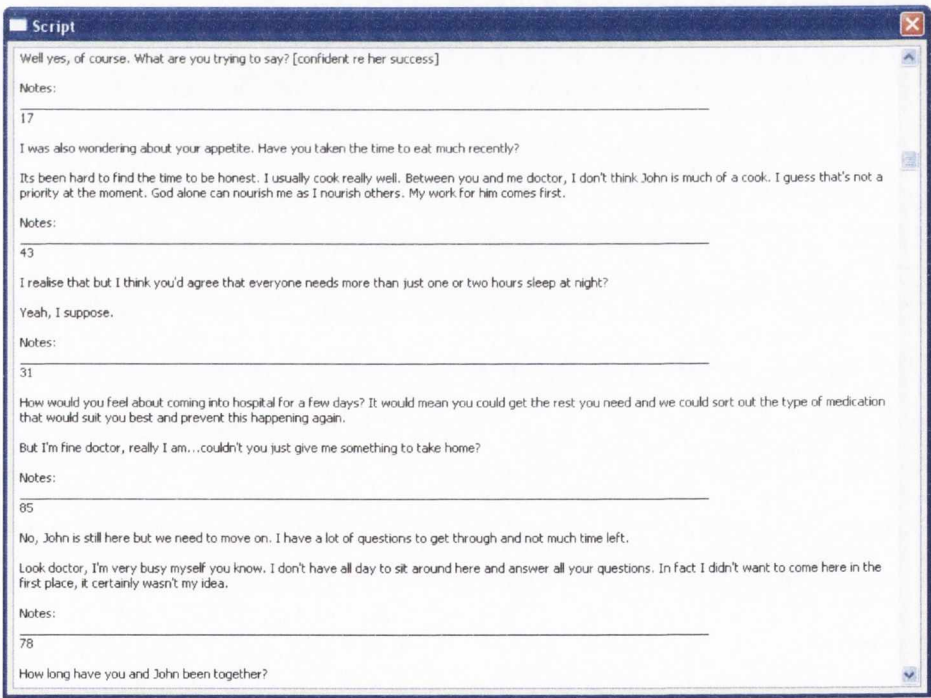


Figure 5.20: Script Function Interface

Highlight

The highlight function allows an author to view Dialogue Elements that have been tagged with particular tags or combination of tags within a Dialogue Model. Unlike the previous two functions described, the highlight function operates on an entire model. This means that the author does not select Dialogue Elements but instead immediately executes the function to

open the SWT popup window interface. The author can then input appropriate values into the popup window and the function uses these as it iterates through the entire Dialogue Model.

To execute the highlight function an author selects it through a menu drop down or can right click anywhere in the Canvas (unused Dialogue Space) and select the appropriate menu function. The highlight function interface can be seen in Figure 5.21. The function initialises its operation with the graphical Canvas component, as opposed to the previous two functions that initiate their operation with the selected Dialogue Elements. The highlight function implements the `IObjectActionDelegate`, another of Eclipse user interface classes, to execute an `IAction`. To populate the Highlight popup window with all of the simulations tagging values (data displayed in each list) the function engages with and reads the properties of the underlying data model of the graphical Canvas component.

Within the SWT interface an author is presented with four lists, one for each of the four tagging properties. The interface allows an author to select single or multiple values within each adaptive property list or across a combination of adaptive property lists. The highlight function uses the selected combination to highlight Dialogue Elements which contain these values. The function allows an author to use the specified highlight combination in one of two approaches. The first approach incorporates an 'OR' Boolean operator, this approach will highlight all Dialogue Elements that contain *any* of the specified highlight values. The second approach incorporates an 'AND' Boolean operator, this approach will only highlight Dialogue Elements that contain *all* of the specified highlight values.

The highlight function operates by altering the line colour of each Dialogue Element that matches the input values based on the selected Boolean operator; the Boolean operator can be select by the author in the '*Highlight Preferences*' section near the base of the of the popup window. This section also allows an author to select the line colour orientation to be used to highlight the Dialogue Elements. The popup window includes a '*Clear*' button which clears a model from any previous highlight operations by resetting all Dialogue Elements line colour to its default.

Highlight Dialogue Elements

Highlight Options:

Categories

Initial Statement

Orientation to/Permission for consultation

Open question

Closed question

Clarifying / Checking Question

Invitation to discuss other options

Summarising Statement

Advice

Validation / Legitimation

Treatment plan

Advice and request to continue with interview

Subjects

Greeting

No Category

Psychotic Symptoms - Delusions

Psychotic Symptoms - Hallucinations

Appetite/Weight

Treatment/Options

Mood Symptoms (Depression/Elation)

Energy/Concentration

Interest/Enjoyment

Social history

Medical/ Medication History

Hopelessness and Suicide

Sleep

Mood Symptoms (Anxiety)

Mood Symptoms (Psychotic Features)

Family History

Roles

Doctor

Consultant

Nurse

Learning Outcomes

breakdown if the patient has any psychotic symptom

Explore the problem a patient presents with using a

Elicit the various symptoms that the patient presents

Explore the risk of self harm/suicide with the patient

Check the patient's mood and associated symptoms

Check if the patient has any associated anxiety sym

Explore if the patient has any psychotic symptoms

Explore how to challenge the false beliefs of a patier

Explore the level of insight the patient has in relator

Identify the precipitating factor to this episode of illr

Explore the patients past psychiatric history and fan

Discuss different treatment options with the patient

Highlight Preferences:

Line Color

Red

Border Size

OR

AND

Highlight

Clear

Cancel

Figure 5.21: Highlight Function Interface

161

The highlight function operates on the Canvas component. The function uses the Canvas as a starting point to iteratively access all Dialogue Elements that it contains. As the function accesses each graphical instance of every Dialogue Element it reads their underlying data model element. The program then compares the selected input values with the Dialogue Element data properties. If there is a match, as dictated by the Boolean operator, the associated graphical representation of the Dialogue Element is altered through a GMF diagram command (`setForegroundColor`). The Clear button works in a similar manner except the default colour is applied to the graphical component of all Dialogue Elements.

Search

The Search function allows an author to locate Dialogue Elements that contain a specific string in their Statement, Response or Notes properties. The Search function operates on an entire Dialogue Model. It is implemented and accessed in a manner that is similar to the Highlight function. An author inputs a search string using an SWT popup window, as seen in Figure 5.22. The search string is entered in the text box provided and the author can select which property is to be searched using the Statement, Response or Notes/Directions radio buttons which are located below the text box. A search match is indicated in the Dialogue Model by altering the line colour of the Dialogue Element.

The approach to emphasising search matches is identical to that used by the Highlight function to highlight tags. Implementation is also very similar to the Search function. The Search function accesses the entire Dialogue Model through the Canvas component and iteratively checks the data model of each Dialogue Element through its corresponding graphical component. If a match is found the graphical representation is altered through a GMF diagram command. The Clear button acts identically to the Highlight functions operation of the same name.

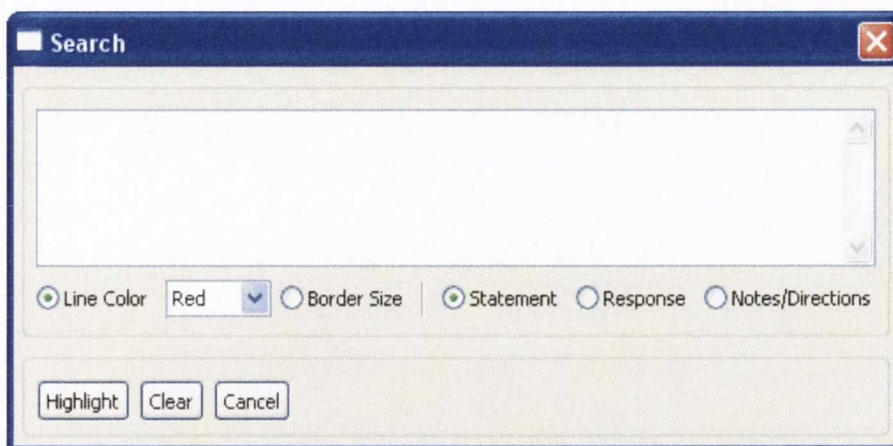


Figure 5.22: Search Function Interface

Validate

The Validate function allows an author to automatically inspect a Dialogue Model to insure it contains vital data and information. The Validate function again operates on the entire Dialogue Model. It is accessed and implemented similarly to the Highlight and Search functions. The objective of the Validate function is to highlight Dialogue Elements that may be incomplete. Specifically it targets Dialogue Elements properties that are vital to the published soft skill simulation. These include the Statement, Response, Statement media file and Response media file properties. A Dialogue Element will be highlighted if, depending on the option(s) selected by the author, they do not contain values. The validation function is operated through a SWT popup window, as seen in Figure 5.23. This allows an author to select options of verification through check boxes. If the Dialogue Element does not contain values in properties as selected by the author using these check boxes they will be highlighted in a manner similar to the Highlight and Search functions. The Validate function differs from the Highlight and Search function as it does not match input Tags or Strings, instead it determines if the appropriate property values are empty (null). If they are empty the graphical component of the Dialogue Element is altered by changing its line colour.

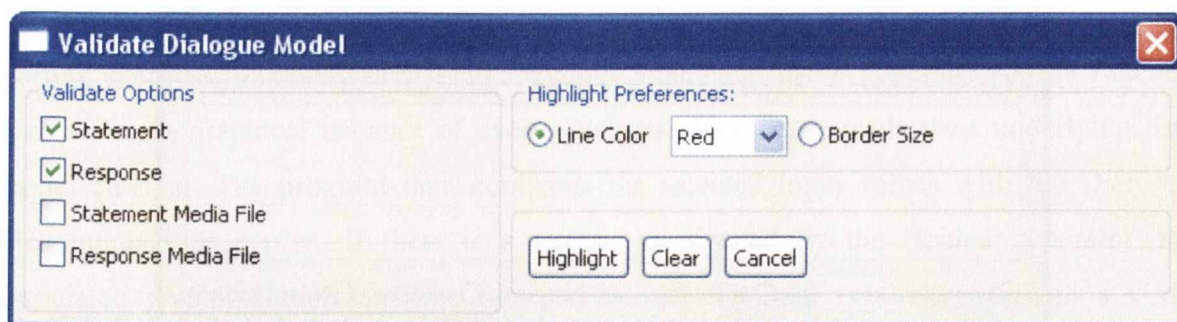


Figure 5.23: Validate Function Interface

Full Script

The Full Script function allows an author to generate the entire dialogue that is contained within a Dialogue Model. This function is used at the end of a simulations development when the Statement and Response media files are to be recorded. The generated script can be used by actors and, if applicable, the director when shooting. The full script function is accessed in a similar manner to the Highlight, Search and Validate functions. Implementation of the Full Script function is slightly different as it does not require any input from an author. Furthermore, while it does access the Dialogue Elements graphical component through the Canvas to read each element's data in the underlying model, it does not need to match or compare any data. This means that once the full script function is executed it automatically generates the full script which is presented to the author in the text box of a SWT popup window, as seen in Figure 5.24. The full script can be copied and pasted into another editor such as Word or Latex for printing.

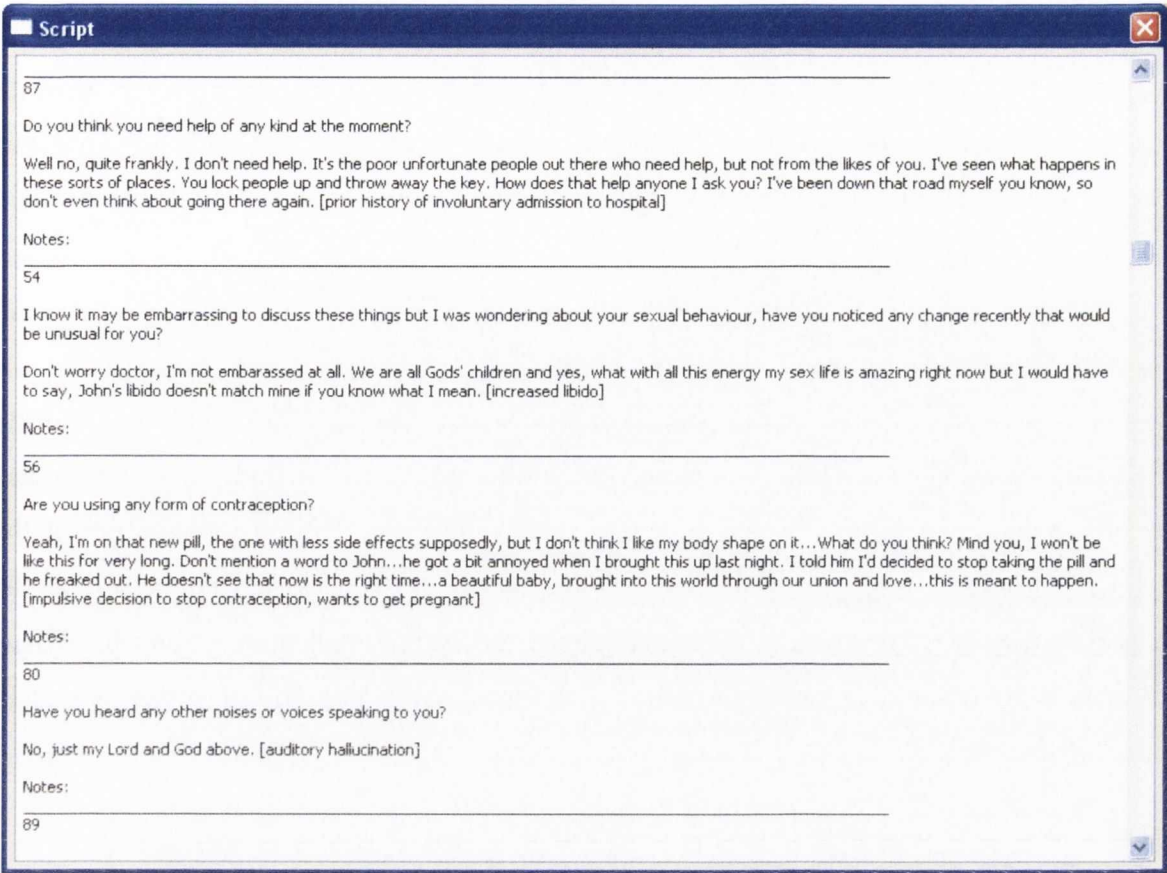


Figure 5.24: Full Script Function

Additional Implementation

There were a number of minor additional issues with the basic Graphical Editor that also requires addressing. Such issues, for example, included adding *set* methods (Java *get/set* methods) for all component *EList* properties. Unlike string properties, no code was generated for *EList* set methods so required implementation. The two more significant implementation issues are described in the following sections.

‘Add Dialogue Element and Connect’ Function

While the basic Graphical Editor provided a number of timesaving short cut functions for adding and connecting Dialogue Elements there was no function that allowed an author to do both in a single operation. The ‘Add Dialogue Element and Connect’ function was developed to allow an author to create a Dialogue Element and connect it to an existing Dialogue Element in a single execution. The function is accessed in the ACTSim authoring tool through a popup menu which is engaged by right clicking on an existing Dialogue Element in the Dialogue Model. The function is implemented as an *IObjectActionDelegate* and employs

an IAction. The new Dialogue Element is initially added to the Dialogue Space at a predetermined and convenient distance directly below the existing Dialogue Element. A connection is then created from the initially existing Dialogue Element to the new Dialogue Element. The underlying model is updated immediately as both of these graphical actions are executed.

Authoring Process

It is important that an author should be allowed to access the ACTSim Authoring Process during the development of their simulation. The Authoring Process is used by an author as a guide or blueprint to follow as they compose their simulation. The implementation of the Authoring Process is very simple. To access the Authoring Process an author selects the appropriate option from a drop down menu at the top of the application (menu bar). This executes a SWT popup window that consists of an image of the Authoring Process, as seen in Figure 5.25.

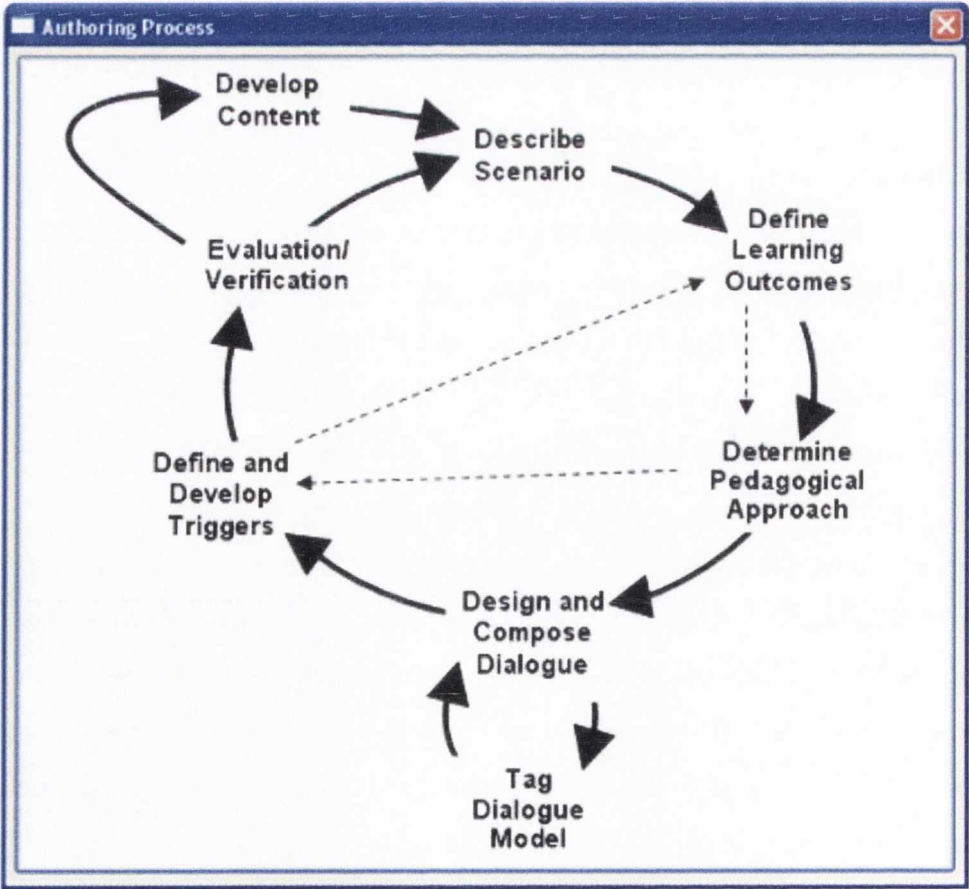


Figure 5.25: Authoring Process Interface

5.4 ACTSim Simulation Title Model

GMF Graphical Editors by default employ two files to store their data; both structured in xml. The first is used to specify the graphical details of the model that is created; the second is used to define the underlying data of the model. This is consistent with the concept of two models working in tandem to fully describe a simulation title, as previously described in this chapter. GMF does however allow the default to be altered so titles can be stored in a single file. This approach was selected as it simplifies saving and publishing procedures. It also allows an author to copy or move simulation titles with ease as it only requires them to do so with a single file.

Storing all data in a single file was implemented by setting a Generator Model property, *'Same File for Diagram And Model'*, to be true. The Generator Model also allowed the file extension of ACTSim to be set as `actsim_diagram`. The ACTSim authoring tool automatically creates a default file with this extension when a new file is created by an author.

ACTSim title files are structured in xml and are composed of two main components. The first component describes the underlying data associated with each file which includes all Canvas properties, Dialogue Element properties and Trigger properties. An example of the beginning of an ACTSim title file can be seen in Figure 5.26. It displays some of the properties of the Canvas component including associated Roles, Subjects and Categories. Figure 5.27 displays another segment of the same simulation title file; it presents several of the Dialogue Elements data elements which stored as children of the Canvas element. Within each Dialogue Element EList properties are stored as child elements while String properties are stored as attributes. Each Dialogue Element also contains a unique identifier which is used to link data model elements to their graphical counterparts. Trigger elements are stored as children of Dialogue Elements that they are associated with and are similarly structured.


```

<?xml version="1.0" encoding="UTF-8"?>
<xmli:XML xmlns:xmli="http://www.omg.org/XMli" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:actsim="http://actsim.ecore" xmlns:notation=
<actsim:Canvas xmli:id="10012" name="Mania Simulation" identifier="mania" general="Mania is a severe medical condition characterized by extremely elevated mood, energy
<roles>Doctor~~a person licensed to practice medicine, as a physician,</roles>
<roles>Consultant~~The job of a medical consultant is to provide medical advice for a variety of purposes. A medical consultant is someone who is very knowledgeable ab
<roles>Nurse~~A nurse is a healthcare professional who, in collaboration with other members of a health care team, is responsible for: treatment, safety, and recovery of ac
<subjects>Greeting~~An initial statement to begin the simulation</subjects>
<subjects>No Category~~No subject applies</subjects>
<subjects>Psychotic Symptoms - Delusions~~A fixed false belief and is used in everyday language to describe a belief that is either false, fanciful or derived from deception
<subjects>Psychotic Symptoms - Hallucinations~~A delusional perception is where a genuine perception (ie. correctly sensed and interpreted) is given some additional (an
<subjects>Appetite/Weight~~The subjects weight or eating patterns are discussed</subjects>
<subjects>Treatment/ Options~~Possible treatments or options that the subject could follow.</subjects>
<subjects>Mood Symptoms (Depression/Elation)~~The patient expresses depression or elation</subjects>
<subjects>Energy/ Concentration~~The patient shows reduced or a great deal of energy/concentration</subjects>
<subjects>Interest/Enjoyment~~The patient has become very interested in the dialogue</subjects>
<subjects>Social history~~The patients social history is discussed in detail.</subjects>
<subjects>Medical/ Medication History~~The patients medical or medication history</subjects>
<subjects>Hopelessness and Suicide~~The patient shows signs of hopelessness and suicide</subjects>
<subjects>Sleep~~The subject of sleeping patterns is examined.</subjects>
<subjects>Mood Symptoms (Anxiety)~~A physiological state characterized by cognitive, somatic, emotional, and behavioral components. These components combine to cre
<subjects>Mood Symptoms (Psychotic features)~~Psychosis is a generic psychiatric term for a mental state often described as involving a "loss of contact with reality.
<subjects>Family History~~The patient family history is examined</subjects>
<categories>Initial Statement~~An initial statement to begin the simulation</categories>
<categories>Orientation to/Permission for consultation~~A meeting for deliberation, discussion, or decision</categories>
<categories>Open question~~A non-leading or directing question that allows the subject to convey information about themselves. To gather more than a yes or no respons
<categories>closed question~~A yes or no question to gather specific information</categories>
<categories>Clarifying / Checking Question~~To confirm information that has been conveyed by the subject.</categories>
<categories>Invitation to discuss other options~~Allow the subject to continue a particular course of options.</categories>
<categories>Summarising Statement~~Summarise information that has been discussed previously</categories>

```

Figure 5.26: Segment of ACTSim Title File (Beginning of File)

```

<dialogueElement xmi:type="actsim.DialogueElement" xmi:id="_1171000818" id="1" name="Good morning, my name is Dr White. Nice to meet you." dialogueElement="_1171654597">
  <statementURI>E:\Mania Interview\Windows Media 512K\Q 001_New\146.wmv</statementURI>
  <response>Nice to meet you Dr White. That's a very pure and angelic name, and how are you today?</response>
  <category>Initial Statement</category>
  <subjects>Greeting</subjects>
</dialogueElement>
<dialogueElement xmi:type="actsim.DialogueElement" xmi:id="_1171654597" id="2" name="I'm very well thank you. I am one of the psychiatric doctors here and to begin with I'd like to get
  <statementURI>E:\Mania Interview\Windows Media 512K\Q 002_New\147.wmv</statementURI>
  <response>Absolutely Doctor. I've given this a lot of thought and well I've discussed it with John, over and over...you know John don't you, he's my new boyfriend...we havent been together
  <category>Orientation to/Permission for consultation</category>
  <subjects>No Category</subjects>
</dialogueElement>
<dialogueElement xmi:type="actsim.DialogueElement" xmi:id="_1179799952" id="3" name="Well, yes, he seems nice, however, I was hoping that we could talk about any problems you m
  <statementURI>E:\Mania Interview\Windows Media 512K\Q 003_New\148.wmv</statementURI>
  <response>Well doctor, there's no problem really, in fact life couldn't be better...life is for living and one must love life, don't you agree?</response>
  <category>Open question</category>
  <subjects>No Category</subjects>
</dialogueElement>
<dialogueElement xmi:type="actsim.DialogueElement" xmi:id="_1179750568" id="4" name="Yes, of course but, can you tell me, have there been any concerns that have brought you here
  <statementURI>E:\Mania Interview\Windows Media 512K\Q 004_New\149.wmv</statementURI>
  <response>On yes, I nearly forgot, forgive me doctor, there has been so much happening that the trivial things just get left behind, so much to say and do, I've probably let myself get carri
  <category>Open question</category>
  <subjects>No Category</subjects>
</dialogueElement>
<dialogueElement xmi:type="actsim.DialogueElement" xmi:id="_1180151781" id="5" name="It certainly sounds like I should take a look at your throat, but maybe we could come back to th
  <statementURI>E:\Mania Interview\Windows Media 512K\Q 005_New\150.wmv</statementURI>
  <response>Well, Doctor, its been quite wonderful really. I feel my energy is eternal. I have so much to do, so much to give, there aren't enough hours in the day, its pretty special...I tried to
  <category>closed question</category>
  <subjects>No Category</subjects>
</dialogueElement>

```

Figure 5.27: Segment of ACTSim Title File (Dialogue Element Properties Data)

The second component of simulation title files contains the graphical descriptions of each Dialogue Element, Connection and Trigger. Figure 5.28 displays a segment of the same simulation title file previously presented. It shows the beginning of the graphical component which is contained by the notation:Diagram tag. The xml code presented here describes the graphical data of three Dialogue Elements; each Dialogue Element has associated with it an ‘element’ attribute which denotes the Dialogue Elements unique identifier so it can be linked to the data contained in the previous component of the file. The graphical data is defined with a combination of types, ids and x-y coordinates that are read and understood by the ACTSim authoring tool. Figure 5.29 presents a segment of the same title file which describes the graphical Connections that exist between each Dialogue Element; these are tagged as ‘edges’.

```
<notation:Diagram xmi.id="11004" type="Actsim" element="10012" name="mania.actsim_diagram" measurementUnit="Pixel">
  <children xmi.type="notation:Node" xmi.id="11008" type="1003" element="_1171000818">
    <children xmi.type="notation:Node" xmi.id="11009" type="4007"/>
    <children xmi.type="notation:Node" xmi.id="11010" type="4008"/>
    <children xmi.type="notation:Node" xmi.id="11011" type="5001">
      <styles xmi.type="notation:SortingStyle" xmi.id="11012"/>
      <styles xmi.type="notation:FilteringStyle" xmi.id="11013"/>
    </children>
    <styles xmi.type="notation:ShapeStyle" xmi.id="11015"/>
    <layoutConstraint xmi.type="notation:Bounds" xmi.id="11016" x="516" y="-24"/>
  </children>
  <children xmi.type="notation:Node" xmi.id="11018" type="1003" element="_1171654597">
    <children xmi.type="notation:Node" xmi.id="11019" type="4007"/>
    <children xmi.type="notation:Node" xmi.id="11020" type="4008"/>
    <children xmi.type="notation:Node" xmi.id="11021" type="5001">
      <styles xmi.type="notation:SortingStyle" xmi.id="11022"/>
      <styles xmi.type="notation:FilteringStyle" xmi.id="11023"/>
    </children>
    <styles xmi.type="notation:ShapeStyle" xmi.id="11025"/>
    <layoutConstraint xmi.type="notation:Bounds" xmi.id="11026" x="786" y="144"/>
  </children>
  <children xmi.type="notation:Node" xmi.id="11028" type="1003" element="_1179299952">
    <children xmi.type="notation:Node" xmi.id="11029" type="4007"/>
    <children xmi.type="notation:Node" xmi.id="11030" type="4008"/>
    <children xmi.type="notation:Node" xmi.id="11031" type="5001">
      <styles xmi.type="notation:SortingStyle" xmi.id="11032"/>
      <styles xmi.type="notation:FilteringStyle" xmi.id="11033"/>
    </children>
    <styles xmi.type="notation:ShapeStyle" xmi.id="11035"/>
    <layoutConstraint xmi.type="notation:Bounds" xmi.id="11036" x="1002" y="258"/>
  </children>
</notation:Diagram>
```

Figure 5.28: Segment of ACTSim Title File (Dialogue Element Graphical Data)


```

<edges xmi:type="notation:Edge" xmi:id="_33Wp4AG9Ed2P97VWgRIN_Q" type="3001" source="11008" target="11018">
<styles xmi:type="notation:ConnectorStyle" xmi:id="_33Wp4QG9Ed2P97VWgRIN_Q"/>
<styles xmi:type="notation:FontStyle" xmi:id="_33Wp4gG9Ed2P97VWgRIN_Q"/>
<element xsi:nil="true"/>
<bendpoints xmi:type="notation:RelativeBendpoints" xmi:id="_33Wp4wG9Ed2P97VWgRIN_Q" points="[75, 47, -195, -121]$(195, 121, -75, -47)"/>
</edges>
<edges xmi:type="notation:Edge" xmi:id="_33Wp5AG9Ed2P97VWgRIN_Q" type="3001" source="11018" target="11028">
<styles xmi:type="notation:ConnectorStyle" xmi:id="_33Wp5QG9Ed2P97VWgRIN_Q"/>
<styles xmi:type="notation:FontStyle" xmi:id="_33Wp5gG9Ed2P97VWgRIN_Q"/>
<element xsi:nil="true"/>
<bendpoints xmi:type="notation:RelativeBendpoints" xmi:id="_33Wp5wG9Ed2P97VWgRIN_Q" points="[0, 0, 0, 0]$(0, 0, 0, 0)"/>
</edges>
<edges xmi:type="notation:Edge" xmi:id="_33Wp6AG9Ed2P97VWgRIN_Q" type="3001" source="11018" target="12238">
<styles xmi:type="notation:ConnectorStyle" xmi:id="_33Wp6QG9Ed2P97VWgRIN_Q"/>
<styles xmi:type="notation:FontStyle" xmi:id="_33Wp6gG9Ed2P97VWgRIN_Q"/>
<element xsi:nil="true"/>
<bendpoints xmi:type="notation:RelativeBendpoints" xmi:id="_33Wp6wG9Ed2P97VWgRIN_Q" points="[0, 0, 0, 0]$(0, 0, 0, 0)"/>
</edges>
<edges xmi:type="notation:Edge" xmi:id="_33ga4AG9Ed2P97VWgRIN_Q" type="3001" source="11028" target="11038">
<styles xmi:type="notation:ConnectorStyle" xmi:id="_33ga4QG9Ed2P97VWgRIN_Q"/>
<styles xmi:type="notation:FontStyle" xmi:id="_33ga4gG9Ed2P97VWgRIN_Q"/>
<element xsi:nil="true"/>
<bendpoints xmi:type="notation:RelativeBendpoints" xmi:id="_33ga4wG9Ed2P97VWgRIN_Q" points="[0, 0, 0, 0]$(0, 0, 0, 0)"/>
</edges>
<edges xmi:type="notation:Edge" xmi:id="_33ga5AG9Ed2P97VWgRIN_Q" type="3001" source="11028" target="11968">
<styles xmi:type="notation:ConnectorStyle" xmi:id="_33ga5QG9Ed2P97VWgRIN_Q"/>
<styles xmi:type="notation:FontStyle" xmi:id="_33ga5gG9Ed2P97VWgRIN_Q"/>
<element xsi:nil="true"/>
<bendpoints xmi:type="notation:RelativeBendpoints" xmi:id="_33ga5wG9Ed2P97VWgRIN_Q" points="[0, 0, 0, 0]$(0, 0, 0, 0)"/>
</edges>

```

Figure 5.29: Segment of ACTSim Title File (Connection Graphical Data

While an ACTSim simulation title file contains all the data required by the adaptive engine it is not structured appropriately. Before a simulation title file is uploaded it must be reconfigured in such a manner that is compatible with the adaptive engine. A simulation title file is reconfigured for upload using an xml transform (XSLT). The xml transform iterates through the ACTSim title file, reads the relevant data and writes it to a new file so it is in a format that is appropriate for the adaptive engine.

The ACTSim tile file is iterated through by tokenizing each element; an additional jar file (saxon9) was added to the Eclipse IDE and subsequent deployment RCP product to allow the xml transform to read element attributes of each token. Nearly all of the data required by the adaptive engine is stored in the underlying data model component of the title file. The simulation tile graphical description component does however contain the Connection details (edges) that exist between each Dialogue Element which were also referenced through each Dialogue Elements unique identifier.

Figure 5.30 presents the beginning of a transformed actsim_diagram file. It displays some of the simulation scenario properties such as the different roles relevant to the simulation. Figure 5.31 presents a further segment of the same transformed simulation title file. It contains the data of single Dialogue Element, all data is configured as child elements of the parent simulation_element with no data stored as attributes. The unique id of the simulation title file is retained but rather than having the Connections as separate elements within the file they are stored as child elements for each Dialogue Element. Trigger components are also contained as child elements within the transformed file.

```

<?xml version="1.0" encoding="iso-8859-1" ?>
<adaptive_simulation xmlns:xmli="http://www.omg.org/XML" xmlns:dyn="http://exslt.org/dynamic"
  xmlns:actsim="http://actsim.ecore">
  <scenario>|
    <identifier>mania</identifier>
    <learnerinformation>
      <description>
        <name>Mania Simulation</name>
        <general>Mania is a severe medical condition characterized by extremely elevated mood, energy, and unusual thought patterns. There are several possible causes for mania, but it is most often associated with an imbalance in the brain's chemistry.
        </general>
        <roles>
          <role>
            <id>Doctor</id>
            <description>a person licensed to practice medicine, as a physician, </description>
          </role>
          <role>
            <id>Consultant</id>
            <description>The job of a medical consultant is to provide medical advice for a variety of purposes. A medical consultant is someone who is very knowledgeable about the medical community, and is responsible for providing advice to other healthcare professionals.
            </role>
          </role>
          <id>Nurse</id>
          <description>A nurse is a healthcare professional who, in collaboration with other members of a health care team, is responsible for treatment, safety, and recovery of acutely or chronically ill individuals.
          </role>
        </roles>
      </description>
    </learnerinformation>
  </scenario>

```

Figure 5.30: Segment of Transformed ACTSim Title File (Beginning of File)


```

<simulations_elements>
  <simulation_element id="_1171000818">
    <relationships>
      <relationship xmlns:xlink="http://www.w3.org/1999/xlink" xlink:href="_1171654597">
        <relationship_type xlink:href="followedby"/>
        <relationship_cardinality xlink:href=""/>
        <relationship_description xlink:href="en"/>
      </relationship>
    </relationships>
    <dialog_element>
      <id>1</id>
      <statement_transcript>Good morning, my name is Dr White. Nice to meet you.</statement_transcript>
      <response_transcript>Nice to meet you Dr White. That's a very pure and angelic name, and how are you today?</response_transcript>
      <roles/>
      <dialog_element_uris>
        <statement_uri>E:\Mania Interview\Windows Media 512K\q_001_New146.wmv</statement_uri>
        <response_uris/>
      </dialog_element_uris>
      <learning_outcomes/>
      <subjects>
        <subject>Greeting</subject>
      </subjects>
      <dialog_element_categories>
        <dialog_element_category>Initial Statement</dialog_element_category>
      </dialog_element_categories>
      <symptoms/>
      <assessments/>
    </dialog_element>
    <adaptivity>
      <roles>
        <active>true</active>
      </roles>
      <prerequisites>|
        <active/>
      </prerequisites>
      <triggers/>
    </adaptivity>
  </simulation_element>

```

Figure 5.31: Segment of Transformed ACTSim Title File (Dialogue Element)

5.5 Authored Simulations

The previous sections presented a detailed description of the ACTSim authoring tool and its different functionality. To augment this description, and to present a complete overview of the authoring tool, this section outlines a completed ACTSim simulation model and corresponding soft skill simulation.

Figure 5.32 presents an ACTSim simulation model developed in the psychiatry domain. The objective of the simulation was to allow medical students to interact with a mania patient. The figure highlights that an ACTSim simulation model can be come very large and complex. There are many different routes a learner could transverse and hundreds of options described by the simulation model developer.

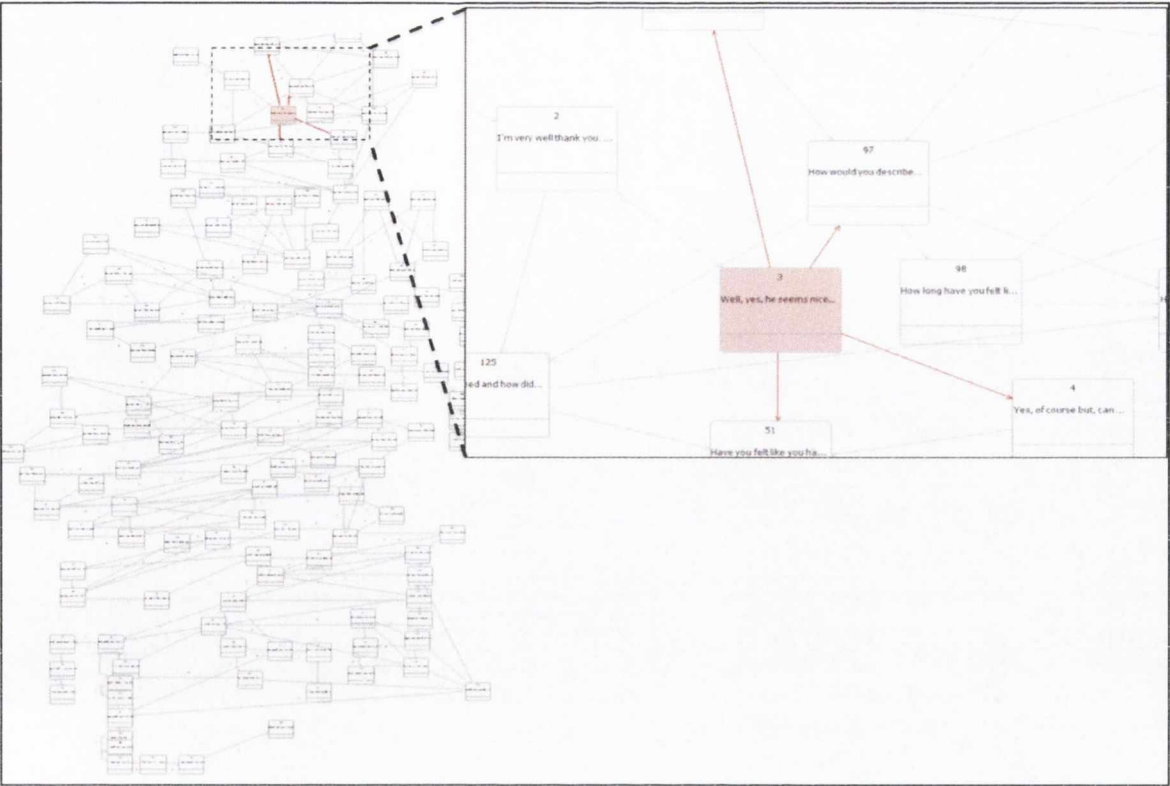


Figure 5.32: Overview of Simulation Model with Magnified Section and Highlighted Dialogue Element

Figure 5.33 presents a screen capture of the mania soft skill simulation described by the model above. There is both audio and video of the simulated female patient that a learner interacts with during the learning experience.



Figure 5.33: Simulation Screen Capture: Response

At the end of each video segment (Dialogue Element) the interface updates and the learner is presented with several key pieces of data, as displayed in Figure 5.34. A learner is presented with the transcript of the decision they have just made, or the “*Current Decision*”, and with the options that are available to continue with the simulation. In the case of Figure 5.34, the learner is presented with four options under the “*Decide*” heading. The screen capture presented in Figure 5.34 maps directly to the magnified section in the simulation model presented previously in Figure 5.32. The Dialogue Element and four simulation options highlighted in red correspond to the decision point presented in Figure 5.34 and its four options or connected Dialogue Elements.

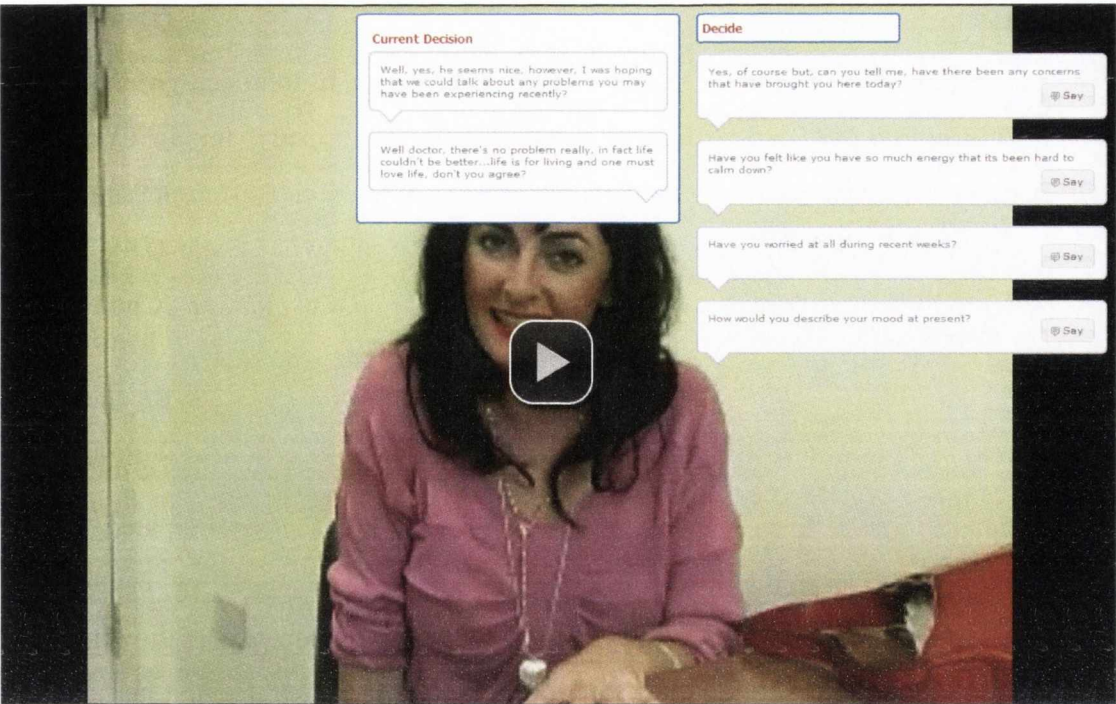


Figure 5.34: Simulation Screen Capture: Statement and Response Transcript with 4 Learner Options

5.6 Summary

This chapter has presented the implementation of the ACTSim authoring tool. It initially outlined the ACTSim architecture in relation to its delivery platform and associated models. This was followed by a detailed description of ACTSim architectural components; Canvas, Dialogue Element and Trigger components. The next section presented a technological architecture view of ACTSim. The second to last section briefly outlined a developed ACTSim simulation model and corresponding soft skill simulation. The final section described, in detail, the components and composition of ACTSims simulation titles. The following chapter presents a detailed account of the evaluation of the ACTSim authoring tool.

6 Evaluation

This chapter describes the evaluations based on an action research approach (McNiff and Whitehead 2011; Kember 2000). The nature of the system developed (Human Computer Interaction - HCI) required many evaluations to be completed at different stages of development with results of each evaluation feeding into the following cycle (Dix 2004). Action research employs many cycles of plan/reflection, action and evaluation (Chou 2010; O'Shea and Reddy 2007; Hendricks 2006).

The first section of this chapter presents an overview of the evaluation key areas; including the overall evaluation goals, evaluation process and user groups that participated in the evaluation experiments. The following section, 6.2, describes the types of evaluations associated with this research as well as a timeline of their completion. The following two sections describe the approaches and results of evaluations completed in the first and second phase of development respectively, sections 6.3 and 6.4. The final sections, 6.5 and 6.6, present a review of related work and a summary of the chapter.

6.1 Evaluation Overview

This section presents an overview of key areas related to the evaluations completed as part of this research. The first subsection outlines the two overall goals of the evaluations. The next subsection describes the research and evaluation process, as previously outlined in chapter one and chapter four of this thesis. The following subsection describes a brief account of the type of environment in which most of the evaluations were completed. This is followed by a subsection that details the different user groups that were involved in completing the evaluations. The final subsection describes the evaluation of soft skill simulations related to this research.

6.1.1 Evaluation Goals

The evaluations described in this chapter serve two purposes:

1. **Incremental Development:** The ACTSim composition tool has been developed through many iterations of design during each phase of development. Evaluations have been an essential component of these iterative cycles. Evaluations have insured that the increments of development have been informed and rigorous.

2. **Overall Evaluation:** The evaluations described in this chapter have closely examined the usability of the ACTSim composition tool. Specifically they have examined effectiveness, efficiency and user satisfaction; a formal definition of usability can be found in Appendix V – Usability Definition.

6.1.2 Research and Evaluation Process

As outlined in chapter one and chapter four of this thesis, development of the ACTSim authoring tool was also divided into two distinct phases. The first phase of development focused on designing, implementing and evaluating an authoring tool that did not address the Adaptivity Application and Representation requirements. The objective of the first phase of development was to initially create a tool that allowed authors to rapidly compose non-adaptive soft skill simulations and to then introduce adaptivity features, with other additional functionality, in the second phase of development.

This approach to development was employed as it insured that the initial authoring tool effectively and efficiently allowed authors to create soft skill simulation Dialogue Models in a user friendly manner. Subsequent designs and evaluations examined adaptivity functionality and other more advanced operations with the knowledge that the underlying initial design successfully addressed typical application functionality and basic soft skill authoring requirements.

Figure 6.1, presented below, extends Figure 4.7 presented at the end of chapter four to include the minor cycles associated with each phase of development. Each minor cycle consists of a plan/reflection component, an action component and an evaluation component. These three components are realised within the software development of this research as design, implementation and evaluation respectively. During the course of this research three different types of evaluations were employed. Specifically, these were predictive evaluations, interpretive evaluations and usability evaluations; each of these types of evaluation is described in more detail later in this chapter.

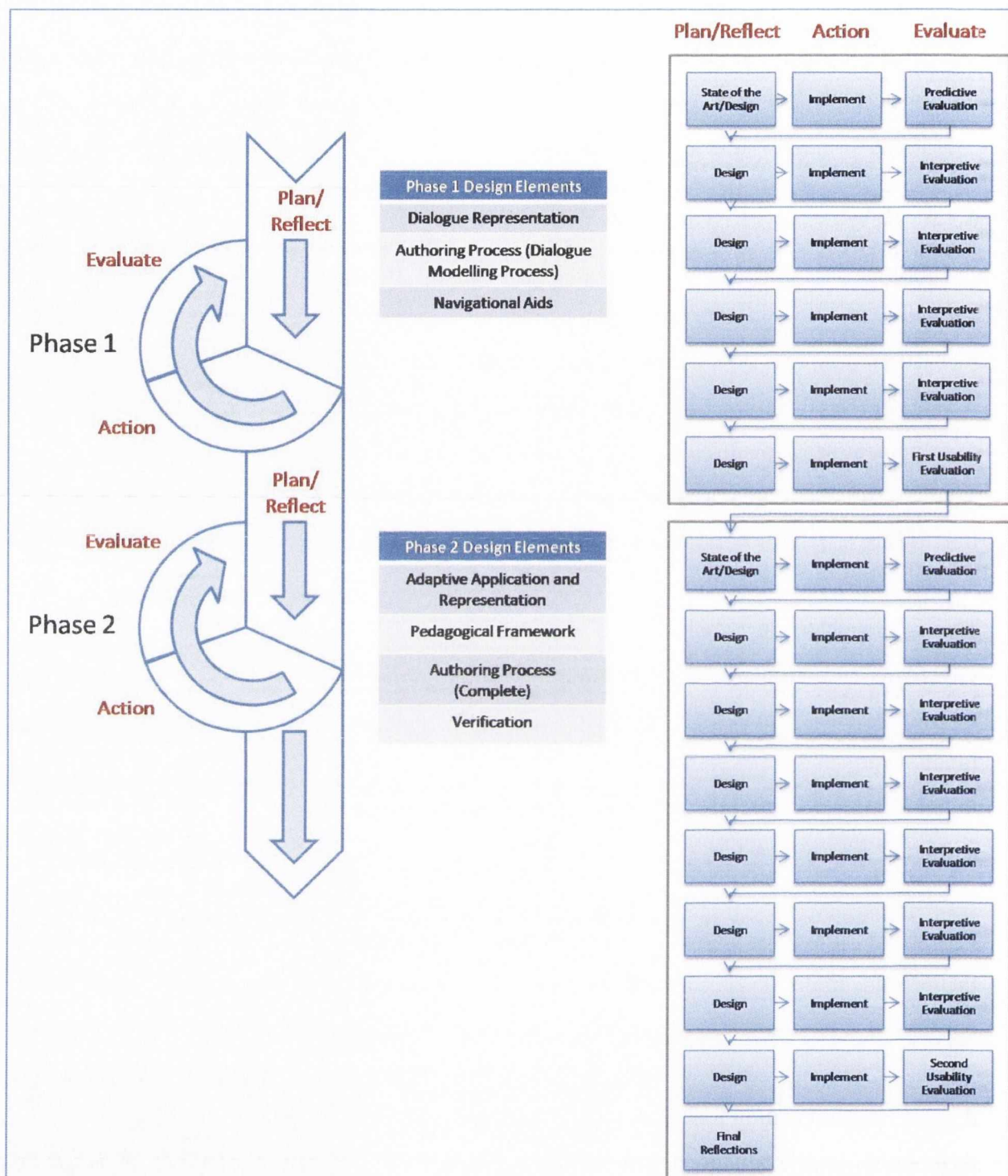


Figure 6.1: Phases of Development, Design Elements and Action Based Phase Sub-Cycles

As highlighted in Figure 6.1, the plan/reflection components are related to the state of the art survey and design components of this research, chapters three and four respectively of this thesis. The action components relate to implementation, as described in chapter five of this thesis.

6.1.3 Real World Evaluations

One of the goals of this research was to create a tool that could be effectively used in the real world by authentic users. To accommodate this objective most of the evaluations conducted as part of this research were completed within an authentic real world environment. This typically consisted of the ACTSim authoring tool being used within a conventional office setting. This allowed ACTSim to be evaluated within the same environment for which it was designed. Evaluations examined ACTSim as part of typical work practices; the results of these evaluations were predominately qualitative in nature.

Employing evaluations within real world settings also allowed users to complete some evaluations in their own time. While ACTSim allows authors to rapidly develop dialogue models for soft skill simulations, the modelling effort still requires time that may not have been consecutively available to typical users. Time constraints of authors inevitably resulted in a Dialogue Model being completed over a number of days. This approach is also more typical of authentic use of the composition tool. It is unlikely that an author would compose a Dialogue Model from beginning to end in a single sitting in the real world. This also added to the accuracy of some of the evaluations conducted.

The users that participated in the evaluation experiments were also real world soft skill simulation experts. The following subsection presents a description of the users that participated in the evaluation of the ACTSim authoring tool as well as an outline of the different user groups; all experiments related to this research were also required to conform to ethical approval requirements, see Appendix VI – Ethical Approval for further detail.

6.1.4 Users

A key criticism of many academic based eLearning authoring tools is that they often employ undergraduates and postgraduates to approximate domain experts. All evaluations in this research were completed by real world subject matter experts who are involved in teaching soft skills and are typical of the target users. Typically, the users were already responsible for teaching soft skills and would normally be developing courses and instructional curriculum. This researches intervention was to allow users to develop simulation courses in their domain of expertise. The simulations composed by the users addressed real world educational issues

and, as such, were authentic instructional simulated scenarios and indicative of the action based research approach.

Soft skill simulations can be incorporated into many different areas of instruction. It was therefore necessary to evaluate ACTSim across a number of domains employing users from different backgrounds. This allowed ACTSim to be examined with respect to different pedagogical goals, types of dialogue interaction and levels of adaptivity. As such, the users in the evaluations conducted as part of this research accurately reflect real world usage. The evaluations that are described in this thesis have employed users from the following areas:

Psychiatry: Soft skill simulations were developed by members of the Department of Psychiatry in Trinity College Dublin. The objective of the simulations developed was to allow medical students to practice clinical interviews.

Customer Care: As part of this research customer care simulations were developed in cooperation with a very large well known software company. The objective of these simulations was to allow customer care agents to converse with European sales customers by offering advice and answering queries.

Exam Stress: Informative soft skill simulation was developed using the ACTSim authoring tool as part of an initiative to assist students with exam stress. The objective of this simulation was to allow third-level students to query and converse with a simulated expert in the area of exam stress.

Healthcare (Social Interaction for Adolescent with Autistic Spectrum Disorders): Several soft skill simulations were developed using ACTSim that were designed to assist adolescents with their social skills. The objective of these simulations was to teach mentally challenged children socially acceptable behaviour.

Healthcare (Informative Adolescent Instruction): Invalid adolescents diagnosed with a serious ailment often find it difficult to communicate with their Doctor or healthcare professional. Informative soft skill simulations were composed to offer an alternative to real

world interaction. The objective of the simulations was to allow adolescent patients to converse with a simulated virtual healthcare professional.

Education (Renaissance Tutorial): The objective of the Renaissance Tutorial simulation was to allow third level learners to interact with an expert as if it were a one-on-one learning experience.

Table 6.1 presents a brief description of the number of users that were employed from each user group as well as an outline of their professional position, expertise and simulation design experience. A more complete description of each evaluation user group and objectives of the simulations that they developed can be found in Appendix VII – Evaluation User Groups.

Table 6.1: Evaluation Participants

Domain	Number of Users	Professional Position	Expertise	Simulation Experience	Simulations Created
Psychiatry	1	Head of Department	Many years experience teaching and developing courses.	High Level Design	Numerous simulations were developed during the course of this research with the department of Psychiatry. Titles included: depression, mania and schizophrenia.
	1	Senior Lecturer			
	2	Course Developers	Psychology and Psychiatry background with some teaching and training experience.	Alternative Authoring Tool Experience	
	2	Lecturers	Clinical and teaching experience.	None	
Customer Care	2	Head of Customer Care Team and Team Instructional Development	Users were head of a customer care team in a large multinational company; responsibilities included team training and development.	None	Simulations titles developed were audio based. They simulated a telephone conversation with a customer.
Healthcare	1	PhD Candidate	The user had previously completed a Masters in the area of Autistic Adolescent Social Interaction	Soft Skill Design Experience	The simulation model developed was designed to assist autistic adolescents in social situations.
	1	Research Masters Candidate	The user was conducting research in the area of instruction to seriously ill children.	None	Models developed would allow children to ask simulated doctors about their illness.
Education	1	Instructional Designer	Many years experience in the area of eLearning and instructional design. Completing a PhD.	Instructional Design Experience	A Renaissance tutorial was developed that simulated one-to-one interaction with a lecturer.

6.1.5 Soft Skill Simulation Evaluations

The evaluations contained in this thesis focus on *authoring* soft skill simulations with the ACTSim composition tool. The effectiveness of the actual soft skill simulations produced by the ACTSim authoring tool is not part of the evaluation of the ACTSim composition tool. Evaluations of the simulations also examine the skills of the designer rather than just the authoring tool and so are not considered applicable to evaluating the ACTSim composition tool. However, as part of a broader project called ADAPT (ADAPT 2010), the project in which the ACTSim composition tool was developed, evaluations of some of the simulations produced were completed, these are briefly described below.

To date the simulations developed in ACTSim have been used by medical students in several universities including Trinity College Dublin, the University of Edinburgh and Imperial College London. ACTSim has also been successfully employed to develop corporate simulations in a large well known multinational company. A number of evaluations of these simulations have been completed. The evaluations were qualitative and subjective in their approach and employed questionnaires as the primary means of gathering data.

The results of the evaluations have been very positive. For example, in Trinity College Medical School; 68.4% of students agreed they had acquired new skills or improved existing ones because of the simulations; 84.2% agreed that they will be able to apply what they have learned from these simulations to the real world; and 68.4% believed the simulations used in the course were an effective method of education. These evaluations and strong uptake of the courses illustrate that the simulations composed using ACTSim are an effective and valuable tool.

6.2 Evaluation Types and Timelines

ACTSim has been rigorously evaluated throughout its development, a necessity with user based applications (Shneiderman 1992). The evaluations have been user centric and several kinds of evaluation techniques have been employed at different stages of ACTSims development (Preece 1994). The goal of the evaluations has been to gather data about the usability of ACTSim and improve effectiveness, efficiency and user satisfaction (Jokela and Iivari 2003). As previously mentioned the evaluations performed have included predictive,

interpretive and usability evaluations. Techniques from each of these categories of evaluations are displayed in the timeline in Figure 6.2; each of these evaluation types is described in detail in this section.

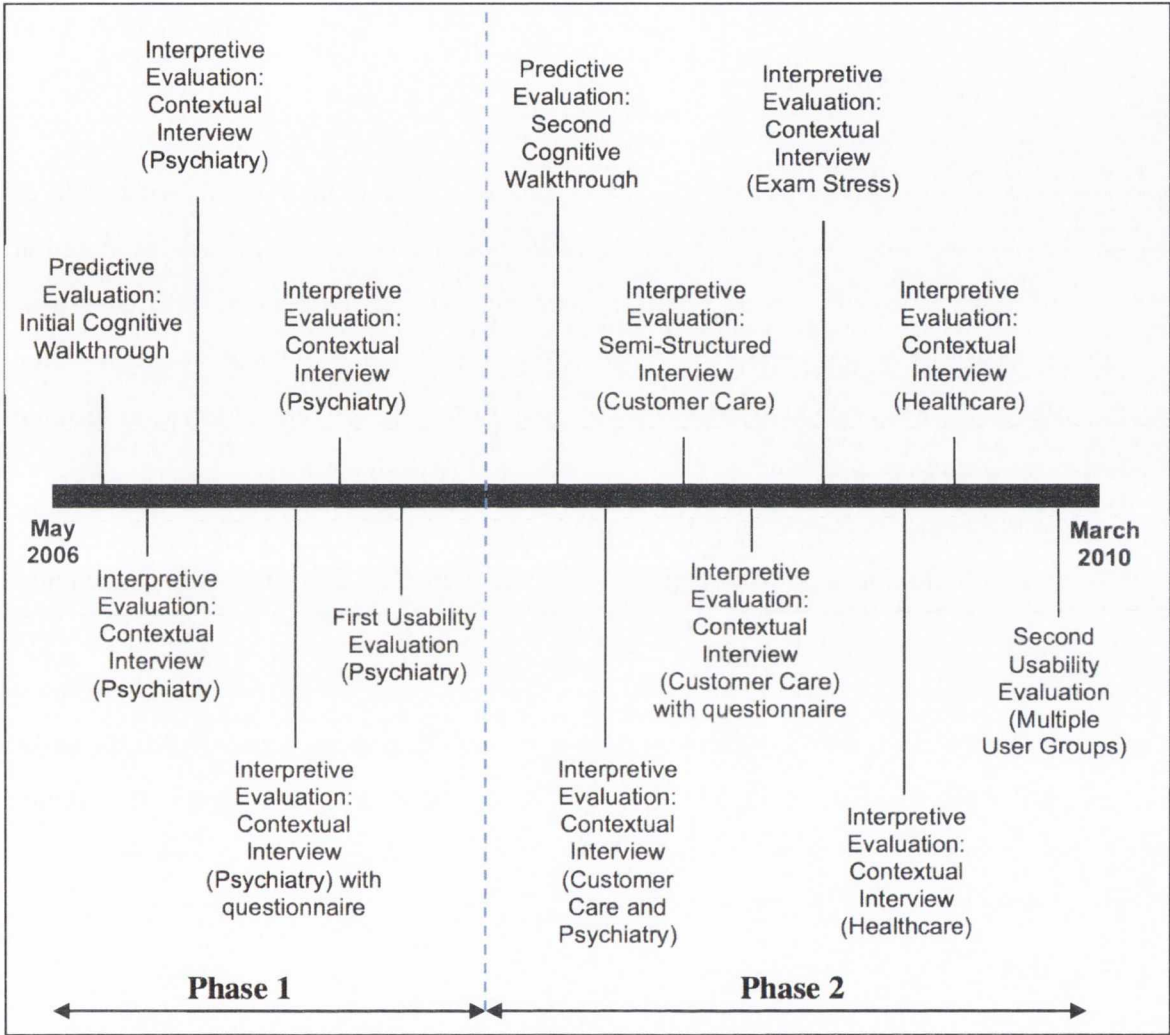


Figure 6.2: Evaluation Timeline

The following subsections describe predictive, interpretive and usability evaluations respectively. They include a brief account of the evaluation type, their occurrence in the evaluation timeline and the user groups associated with the evaluation.

6.2.1 Predictive Evaluations

This research employed cognitive walkthroughs (Rieman and Franzke 1995) as part of its evaluation strategy. A cognitive walkthrough is a particular type of predictive evaluation (Preece 1994) that does not require a prototype. This approach instead employs paper based specifications and storyboards of a system. These allow tasks to be completed cognitively

which highlight issues and allow them to be detected very early in the development processes. Cognitive walkthrough in this research were developed using the requirements previously described in this thesis. Typically a cognitive walkthrough involves expert reviewers but due to the innovative nature of the research the walkthroughs were completed with users that had previous experience of developing soft skill simulations or experts in eLearning and adaptive hypermedia.

During the course of this research cognitive walkthroughs were completed at two points of development, as highlighted in Figure 6.3. The first cognitive walkthrough was executed at the beginning of phase one of development, where requirements associated with basic soft skill simulation authoring were examined. The second cognitive walkthrough was completed at the beginning of phase two where additional requirements, particularly those associated with adaptivity, were evaluated.

The first cognitive walkthrough, which primarily employed two users from the Psychiatry user group, allowed prototyping of ACTSim to begin very quickly. The second cognitive walkthrough, which also employed two users from the Psychiatry user group, was built upon the existing prototype. It allowed components to be redesigned and new elements to be introduced. The second cognitive walkthrough was also completed with experts in the area of authoring Adaptive Hypermedia in eLearning.

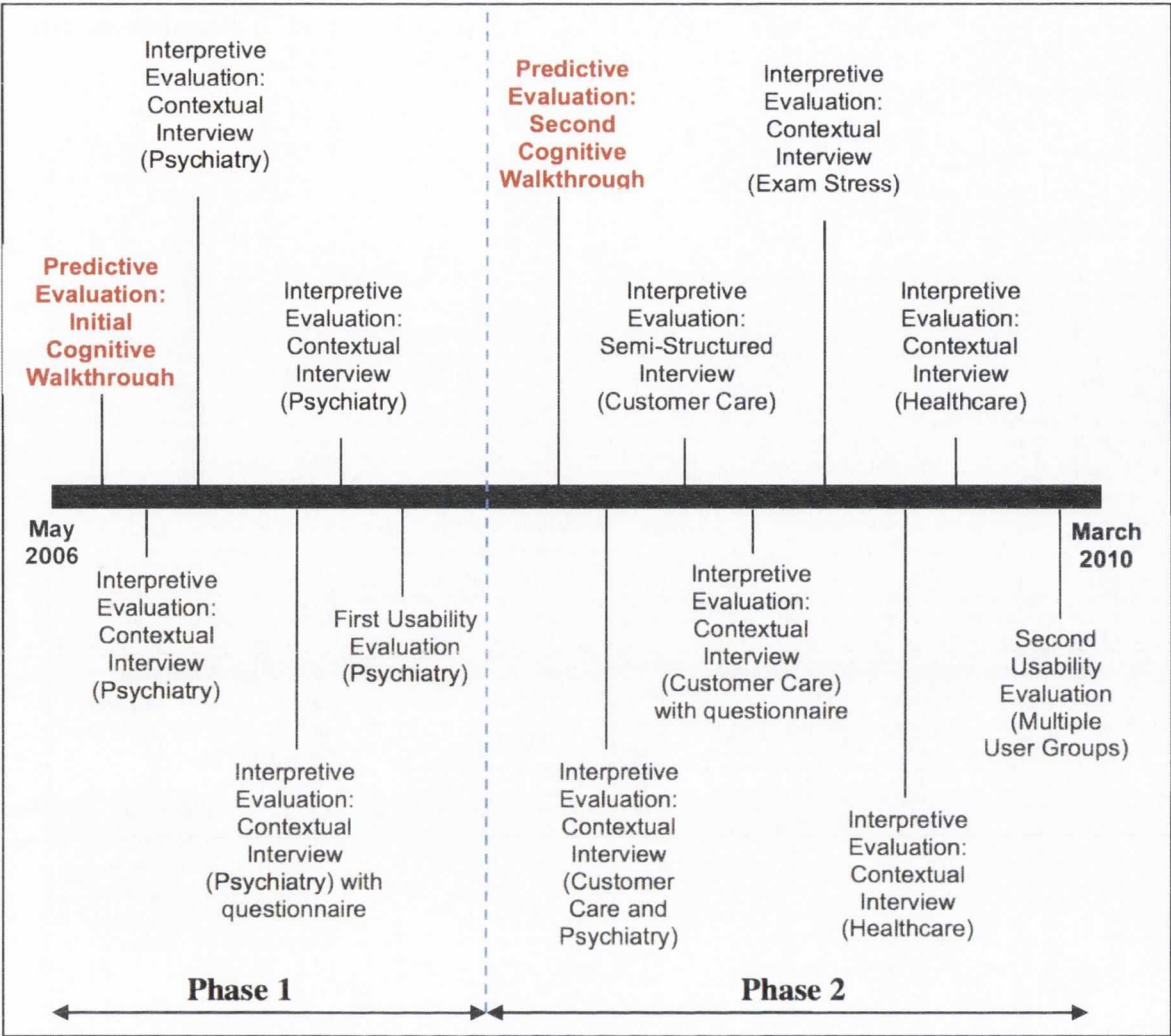


Figure 6.3: Evaluation Timeline – Predictive Evaluations Highlighted

6.2.2 Interpretive Evaluations

The evaluation techniques used most frequently within this research have been interpretive based. Interpretive evaluations assist the understanding of complex interactions that occur in natural settings (Walsham 1993). They are a formative target driven (Preece 1994) approach which can be completed quickly. Contextual interviews (Holtzblatt and Jones 1993) in particular were employed as they complement the action research approach used to develop the ACTSim composition tool. They allowed users to effectively and efficiently influence design decisions which resulted in rapid prototyping.

Interpretive evaluations were employed at many stages in both phases of development, as highlighted in Figure 6.4, and were completed with users from several different domains. These included users from psychiatry (two authors in the first phase and five authors in the

second phase), customer care (two authors), exam stress (one author) and healthcare (two authors).

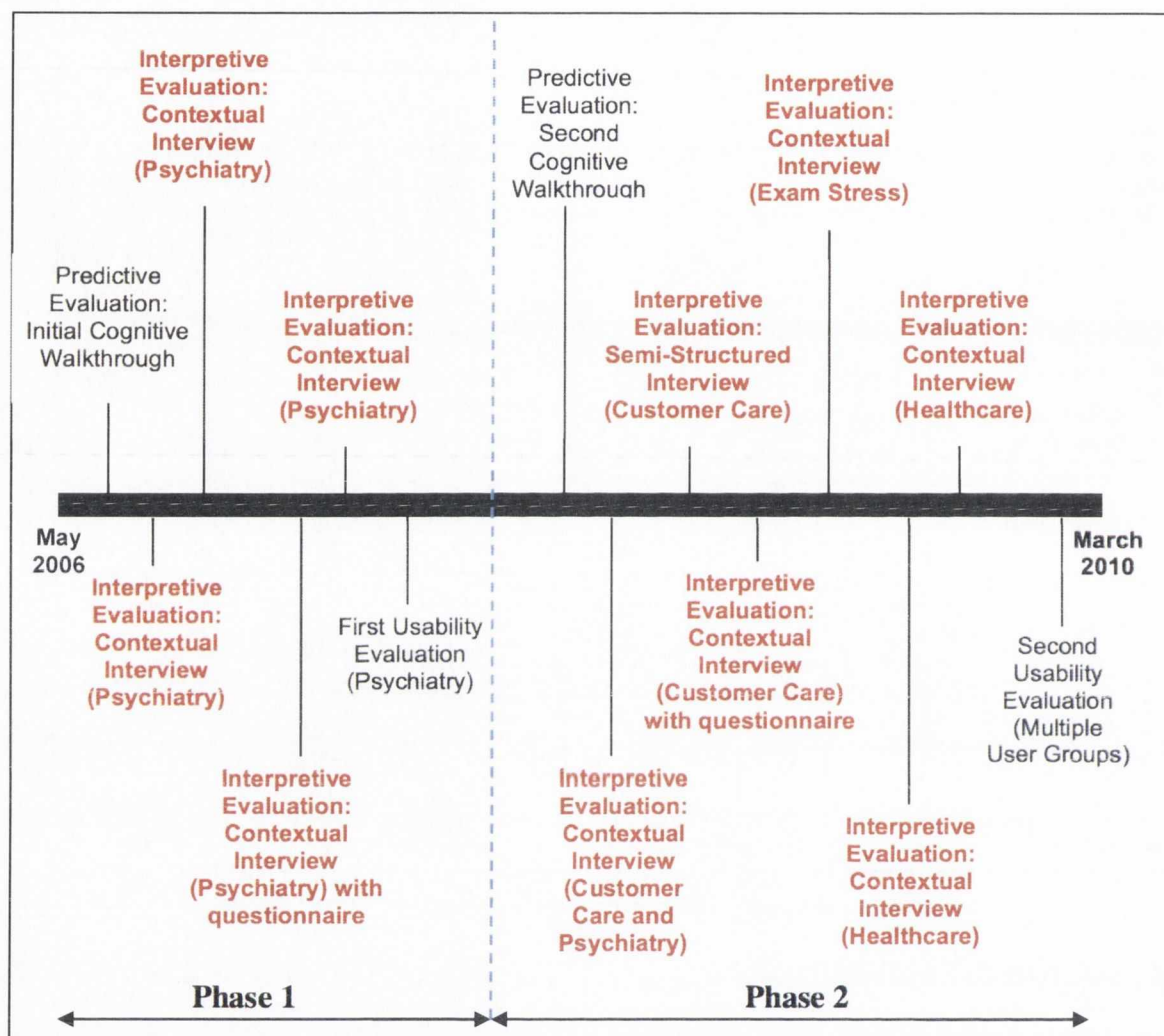


Figure 6.4: Evaluation Timeline – Interpretive Evaluations Highlighted

The approach of interpretive evaluations is to allow users and evaluators to work closely together evaluating a system in its natural setting (Holtzblatt and Beyer 1993); this is exhibitiv of the action research approach employed. Interpretive evaluations were completed with different degrees of interpretation and user control. Some contextual interviews allowed the users a great deal of control and can be thought of as being participative and cooperative (Preece 1994) in nature. These types of interviews were less structured and the users were allowed to expand the scope of the discussion (Holtzblatt and Beyer 1993). Alternative methods of evaluations emphasised the statement of goals (Walsham 1993) and the use of more quantitative tests.

The interpretive evaluations objectives, particularly in the second phase of development, were not necessarily clearly defined at the beginning but were later negotiated collaboratively with users. This approach allowed users a greater degree of input into the evaluation, design and development of a system. The interviews were generally documented as audio recordings and in some cases video recordings. The primary means of data collection employed was audio recording and, in some instances, video recording. The approach taken to process the data collected was to review each of the audio or video recordings to correlate a set of design changes and to then implement these changes. The overall effects of the results of the interpretive evaluations can be seen as design changes in the system; described in sections 6.3.2 and 6.4.2.

A number of interpretive evaluations incorporated less user control than some of the more participative interviews, particularly at later stages of development. For example, elements of usability evaluations (Preece 1994), such as questionnaires and semi-structured interviews, were also included within the contextual interviews. These were used to further analyze feedback, make selections between alternative designs and to confirm evaluation metrics were being followed. They were designed to be short and less comprehensive than complete usability evaluations. For example, a contextual interview with Psychiatry users in the first phase of development incorporated a questionnaire which decreased the amount of user control. These types of experiments were less interpretive and allowed more objectivity. They required clear goals to be defined before the evaluation and data could be easily correlated and results accounted.

Furthermore, more than one type of objective interpretive evaluation was conducted with the same user group. For example, short questionnaires and directed discussions were conducted with the Customer Care users in the second phase of development. These were used to evaluate and examine several concepts including the newly incorporated personalisation features. These experiments were not intended to be as rigorous as complete usability evaluations but instead allowed aspects of ACTSim to be rapidly evaluated.

6.2.3 Usability Evaluations

Two usability evaluations (Tyldesley 1988) were completed during the development of the ACTSim composition tool; both at the end of the two main phases of development, as

highlighted in Figure 6.5. Usability evaluations were employed far less frequently than interpretive evaluations as they require a great deal of time and resources to design and complete. Usability evaluations are however necessary within the development of a user based application as they provide subjective quantitative and qualitative data (Preece 1994). Experimental evaluations such as these are the most powerful methods of evaluating a design or an aspect of design (Dix 2004). In particular they allow system requirements to be closely examined in an objective manner.

Both usability evaluations completed were designed on user based trials and were completed in both natural settings and controlled environments. Evaluation metrics were derived from application requirements (Preece 1994). These metrics were then used to develop tasks (Macaulay 1995) for the users complete. Data for the first usability evaluation was gathered with questionnaires. Data for the second usability evaluation employed several techniques including questionnaires, video analysis and an examination of the user's comprehension of the ACTSim system by testing their memory retention and understanding.

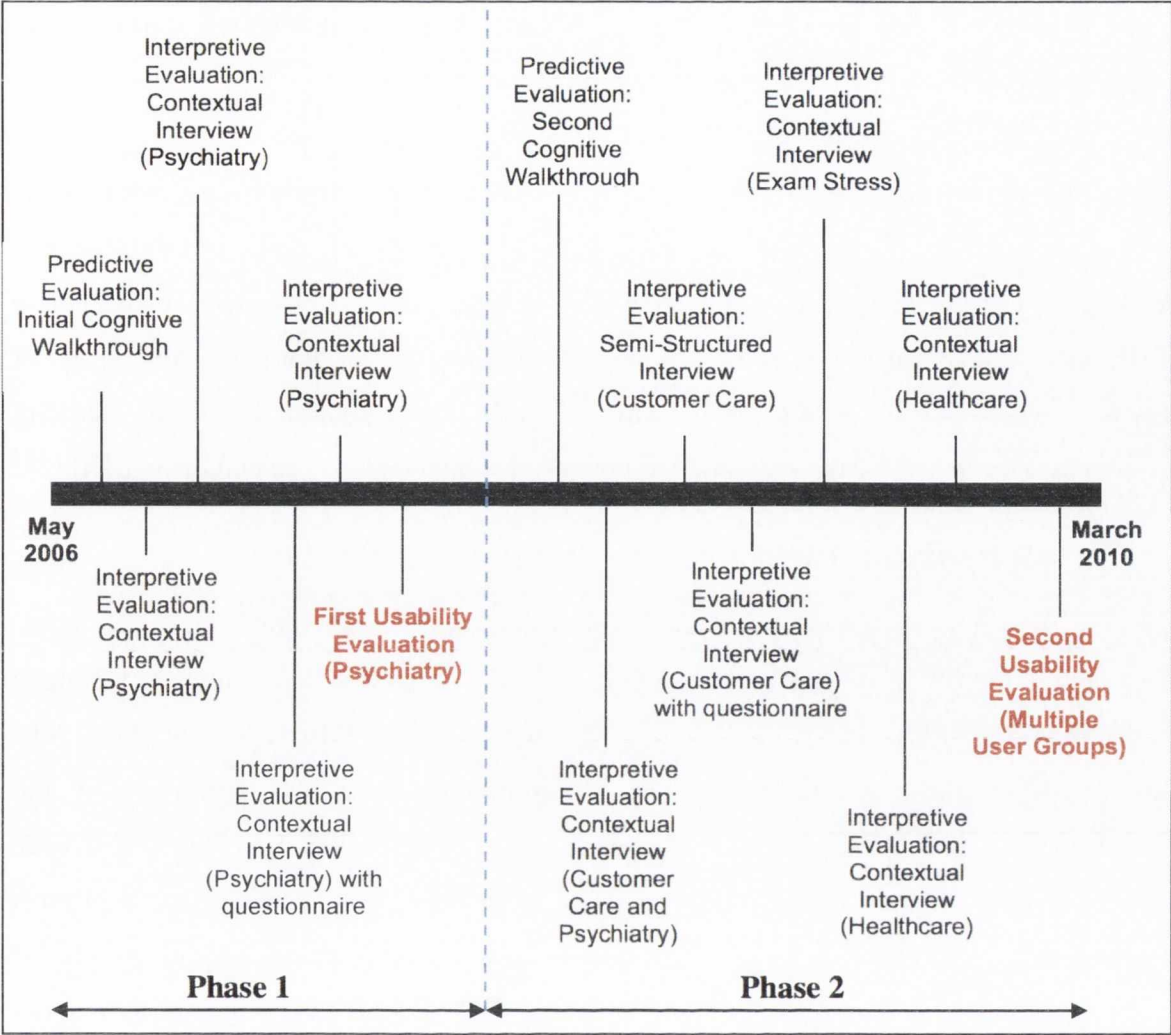


Figure 6.5: Evaluation Timeline – Usability Evaluations Highlighted

Tasks were carried out by each user in order to achieve a work goal which closely examined an aspect of ACTSims design. Once the users had completed all the tasks the data was collected and could be examined to ensure planned levels of usability were achieved.

One of the difficulties in conducting a usability study for a niche area, such as authoring, is that the number of people who compose eLearning is far smaller than the number of people that might consume eLearning. Specifying general eLearning composition to the domain of soft skills and the number becomes lower. Refining this again to authoring soft skill simulations and the number of potential users becomes even lower. Therefore attempting to obtain a large population of users for this research’s usability evaluations was beyond the scope of a PhD thesis. It was however possible to identify and gather five users for each of the usability evaluations that were completed; numerous studies have shown that little more

is learnt from user based applications when evaluation participant numbers exceeds five (Nielsen and Molich 1990; Nielsen and Landauer 1993).

Although five participants may not be statistically significant (Woolrych and Cockton 2001) they are representative of the small number of users at which ACTSim is targeted. The evaluations have also been based on the experience of the users after prolonged use and deep engagement. The experiments were further strengthened by the fact that a diverse set of users was employed in the second usability evolution. The results from the usability evaluations are indicative of the potential strengths and weaknesses of the authoring tool.

6.2.4 Development Approach

As previously described, the design of the composition tool was separated into two phases, the first addressing the requirements for composing Dialogue Models; these include Dialogue Representation, the Dialogue Model Authoring Process and the Navigational Aids. The second phase of design addressed the incorporation of Adaptivity Representation and Application as well as Verification, the Pedagogical Framework and the Simulation Authoring Process. As the second phase of design is so reliant on the success of the first it was necessary to initially evaluate the requirements for composing the dialogue models. If the composition tool cannot be successfully used to compose soft skill simulations without adaptivity, the addition of adaptivity and more advanced functionality would certainly have resulted in failure. This approach also allowed the design decisions regarding adaptivity to be made in a more informed manner.

6.2.5 Reflections at the End of Each Phase of Development

The two phases of development have followed an action based approach composed of reflection, action and evaluation. Reflection components of this cycle are complex and important as they must incorporate information and data from the state of the art survey and evaluation results to form and structure software design. Reflection points connect each cycle, both major and minor, within development so are important components of the action based research approach.

There are numerous minor cycles within each of the two phases of development. While these minor cycles incorporate reflection points they are of a fine granularity relative to the overall development process. There are also two major points of reflection at the end of each of the

two phases of development. These are the two *key* instances within the development process which consider the comprehensive results of the two usability evaluations. They are the two milestone points where requirements had been addressed, functionality had been fully implemented and objective thorough examinations had been completed. The following two sections, 6.3 and 6.4, present a detailed account of the two phases of development. At the end of each of these sections are included a description of the two major reflection points.

6.3 Phase One of Development

This section presents the evaluation approaches and results associated with the first phase of development. It is divided into three subsections; the first subsection describes the first predictive evaluation completed, the following subsection examines the four interpretive evaluations completed in the first phase of development and the final subsection presents the usability evaluation completed at the end of the first phase of development.

6.3.1 Phase One Predictive Evaluation

The predictive evaluations in this research were realised as cognitive walkthroughs and were an intricate part of the initial composition tools design process. The first cognitive walkthrough was completed at the beginning of the first phase of development. The objective of this evaluation was to gather feedback from users so as to incrementally improve storyboards. As such, the evaluation process involved both the feedback from the storyboard and the iterative improvement of outlined functionality. This cognitive walkthrough evaluated the initial interface layout and associated functionality of the ACTSim composition tool. Measurements were qualitative in nature and based on users subjective feedback. The technique for gathering data was to question users as they were guided through a sequence of steps, using storyboards, which defined individual system operations. System functions were deemed to have acquired an acceptable level of usability once no issues could be predicted by participants. An outline of the evaluation method and results of the predictive evaluation completed in phase one of development is described below.

Phase One Predictive Evaluation Method

Storyboarding of the ACTSim system was initially mapped out using paper and pencil. The iterative designs incorporated the systems phase one requirement as dictated by the state of the art survey. These included Dialogue Representation, the Dialogue Model Authoring Process and the Navigational Aids. Basic system functionality, such as file operations, was not explicitly stated at this stage of design as it was considered a standard requirement. The

paper and pencil designs were modified and amended through multiple iterations. Feedback of the different iterations was gathered using cognitive walkthroughs. Once no usability issues could be predicted by participants the predicative evaluation was complete. For clarity, the paper and pencil storyboards were transcribed to an electronic medium using Microsoft PowerPoint (PowerPoint 2011) and Visual Basic (Visual Basic 2011). The cognitive walkthroughs conducted during this phase of development were completed with two Psychiatry authors who possessed previous experience in developing soft skill simulations as well as experts in area of eLearning.

Phase One Predictive Evaluation Results

The combination of evaluation and development between evaluations within the first phase predictive evaluation produced detailed storyboards. While there were numerous cycles of development and evaluation it would not be practical to describe each iteration. The feedback of the evaluation participants is captured in the resultant storyboards.

Figure 6.6 and Figure 6.7 illustrate two of the many final storyboards (mockups used to illustrate the applications interface and usage). These storyboards were used to describe the non-adaptive requirements of the composition tool such as the basic functionality for authoring dialogue and the related authoring process. The storyboard presented in Figure 6.6 displays how the initial design of the ACTSim authoring tool might look. It includes a Dialogue Space, palette and an example of how a Dialogue Model might be constructed. The storyboard displayed in Figure 6.7 was used to demonstrate how the zoom functionality and the map feature might be incorporated into the design. Additional storyboards composed at this stage of development can be found in the appendix of this thesis, see Appendix VIII – Storyboards.

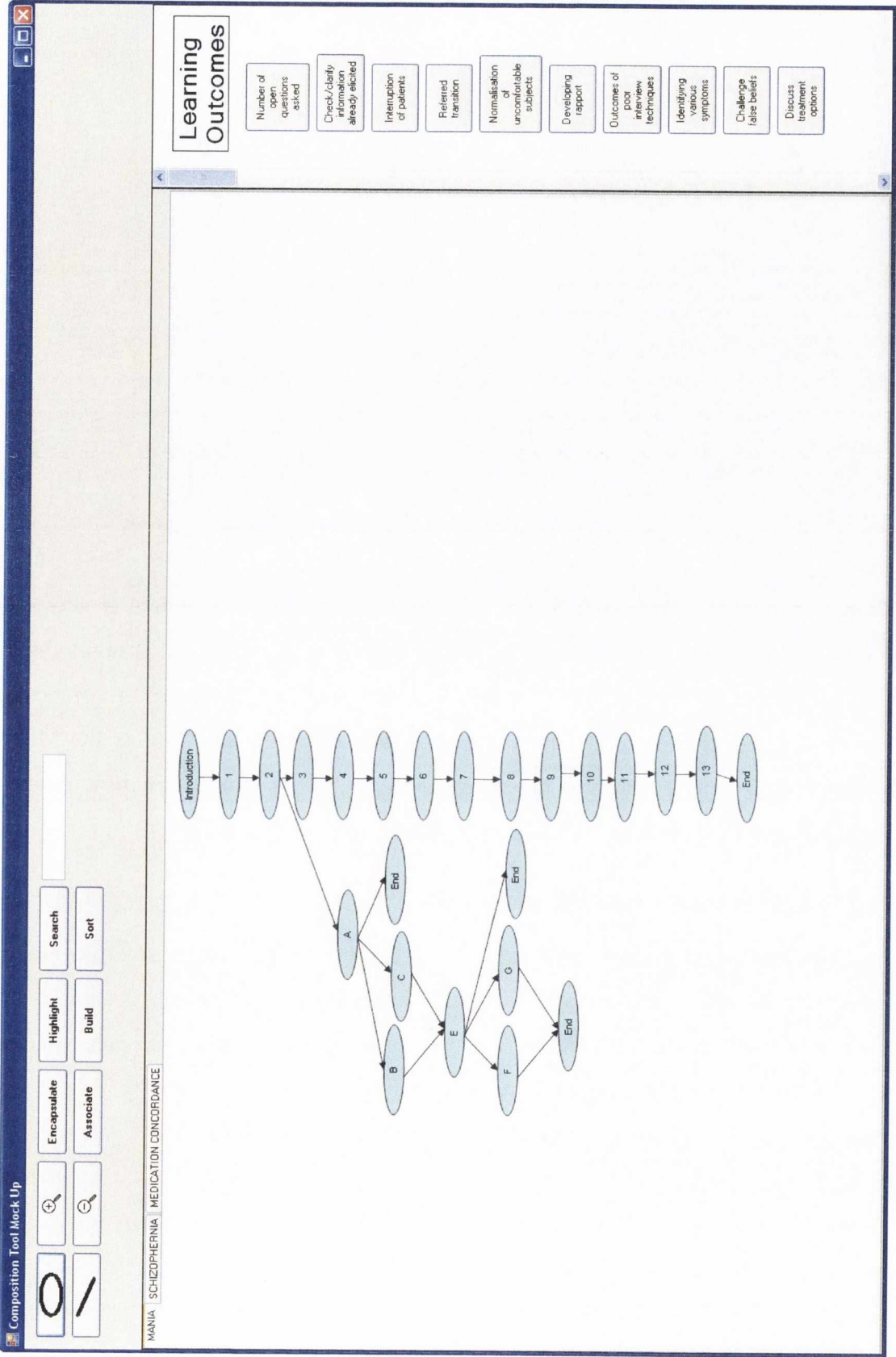


Figure 6.6: First Cognitive Walkthrough Story board – Mockup of Initial Design of ACTSim Authoring Tool

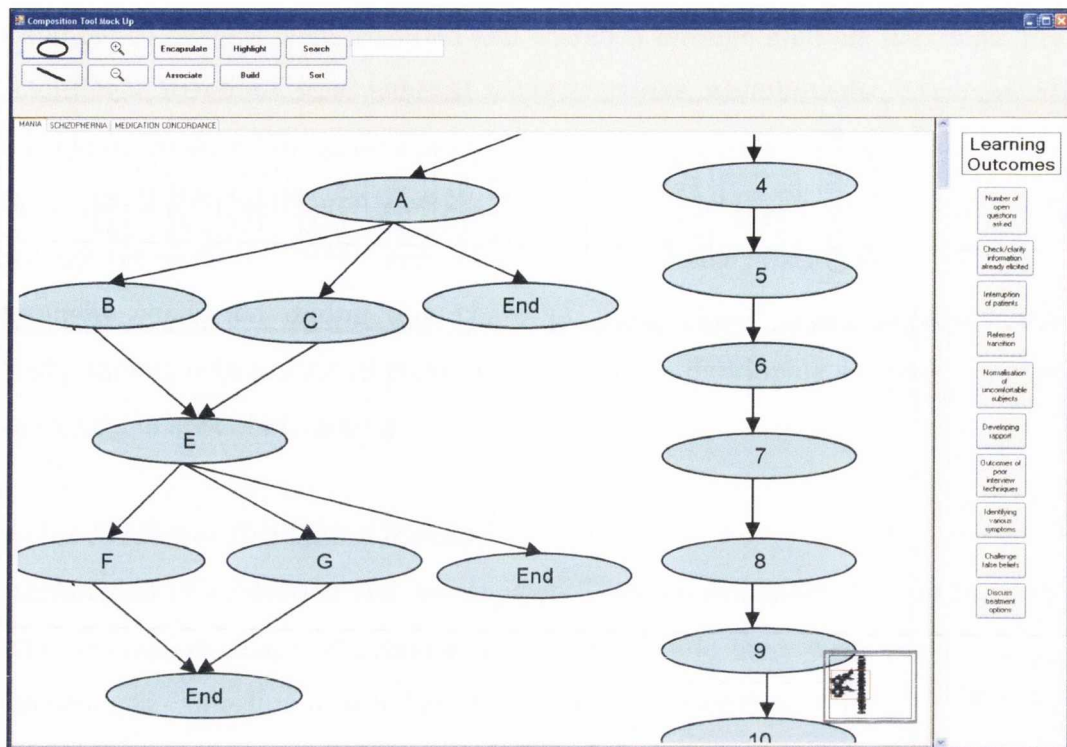


Figure 6.7: First Cognitive Walkthrough Storyboard - Mockup of Use of the Zoom Function with Map

The initial cognitive walkthrough focused on the Dialogue Representation. It allowed the system to be developed with the Dialogue Space central to its interface. It highlighted the need for a palette which incorporated functionality that was essential to composing a Dialogue Model. It also highlighted the need for Navigational Aids such as the Map and Zoom functionality. The initial component interfaces, the Dialogue Element Properties window for example, were also designed at this stage of development.

6.3.2 Phase One Interpretive Evaluations

A number of interpretive evaluations were completed during the first phase of development using the ACTSim prototype which was initially based on the storyboards developed in the first cognitive walkthrough. The objective of the interpretive evaluations was to closely examine and evaluate ACTSim functionality with respect to system usability. The feedback from the users allowed the composition tool to be incrementally improved and augmented with functionality as dictated by system requirements and storyboards. Evaluation metrics were based on qualitative feedback from participants and examined their assessment of effectiveness, efficiency and user satisfaction. The phase one interpretive evaluations employed three contextual interviews and a contextual interview which also included a questionnaire; the contextual interviews were participative and cooperative in nature. The

method and results of the interpretive evaluations completed in phase one of development are described below.

Phase One Interpretive Evaluations Method

During phase one of the development process there were four interpretive evaluations conducted with the psychiatry user group. The interpretive evaluations were each aligned with the following pattern:

- Install initial/updated ACTSim software on users computers
- Allow users time to compose models and operate the different functionality of the ACTSim system
- Conduct the interpretive evaluation
- Review the evaluation
- Design updates to the ACTSim system
- Implement the updates to the ACTSim system
- Prepare for the next software install

The initial sequence of the pattern described above began with installing the first prototype of the ACTSim system which was built using the cognitive walkthrough storyboards as design schematics; not all of the design elements were immediately incorporated into the prototype. The focus of the first prototype and interpretive evaluation was Dialogue Representation and basic system functionality. The second interpretive evaluation examined the functionality of the navigational aids such as the map feature. The third interpretive evaluation, which included a short questionnaire as well as a contextual interview, focused on sub-dialogue representation in the modelling tool. The fourth and final interpretive evaluation examined the main stem branching approach employed by the Dialogue Modelling Process.

Phase One Interpretive Evaluations Results

This section describes the results of each of the interpretive evaluations that were completed during the first phase of development.

First Contextual Interview: Dialogue Representation and Basic Functionality

The initial design focused on dialogue modelling; this included adding Dialogue Elements to the Canvas, connecting Dialogue Elements, accessing properties and design of associated

interfaces. The initial design of the ACTSim prototype did not include Triggered events; as such the initial GMF diagram model did not contain the trigger classes that are presented in Figure 5.6.

There were no major design issues detected with the first prototypes approach to Dialogue Representation or basic functionality. The most notable adjustment to the design of the ACTSim prototype was the Dialogue Elements property interface. Users commented that the initial storyboard designs of the 'Node Editing Box', as presented in Figure 6.8, did not define and highlight the learner option and simulation response in an effective and user friendly manner; learner option and simulation response is previously discussed in section 4.2.2, Dialogue Representation. The initial design of the storyboard interface described both of these properties, 'Name' and 'Script', but their characterisation and layout in the initial interface was found by users to be unintuitive and abstract. Discussions in the first contextual interview resulted in these properties being promoted within the Dialogue Element (previously called the Bubble) interface to a more prominent position. The properties were renamed 'Question Transcript' and 'Answer Transcript', terms that are identifiable with psychiatric interviewing. Later these properties were renamed again to the more generic terms of 'Statement' and 'Response'.

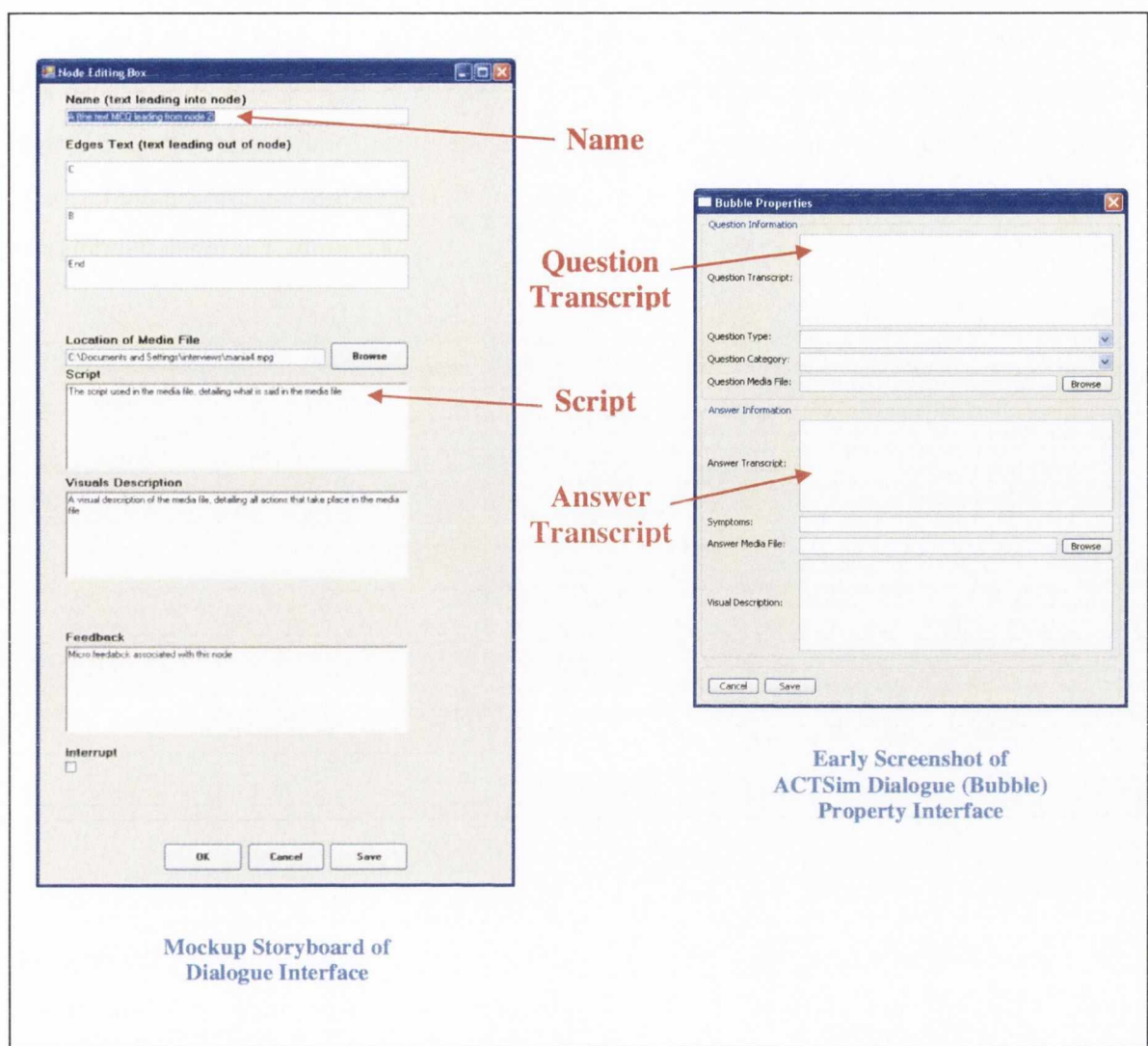


Figure 6.8: Comparison of Node Editing Box Storyboard and Early Screenshot of ACTSim Dialogue (Bubble) Property Interface

Second Contextual Interview: Navigational Aids

There were several navigational aids implemented at this stage of development; including zoom, map, list, arrange features. Users found the usability of each of these features to be acceptable. Users commented that the map feature was the most used and useful of the navigational aids.

As described in chapter five, the ACTSim system was implemented with the eclipse GMF plug-in. This resulted in the map feature, a default GMF feature, being promoted to a more pronounced position in the design layout of the ACTSim system. This is somewhat apparent when comparing the original storyboards to a screenshot of ACTSim taken from the author's 2007 publication (Gaffney, Dagger et al. 2007) as presented in Figure 6.9. GMF also afforded

the map feature with much more functionality than originally envisaged in the first phase cognitive walkthroughs. It was originally thought of as being a simple point of reference for the author. The GMF implementation resulted in allowing a user to alter the Dialogue Space (Canvas) view through manipulation of the map. Based on user comments, this created a very effective, efficient and user friendly method of navigating large complex Dialogue Models.

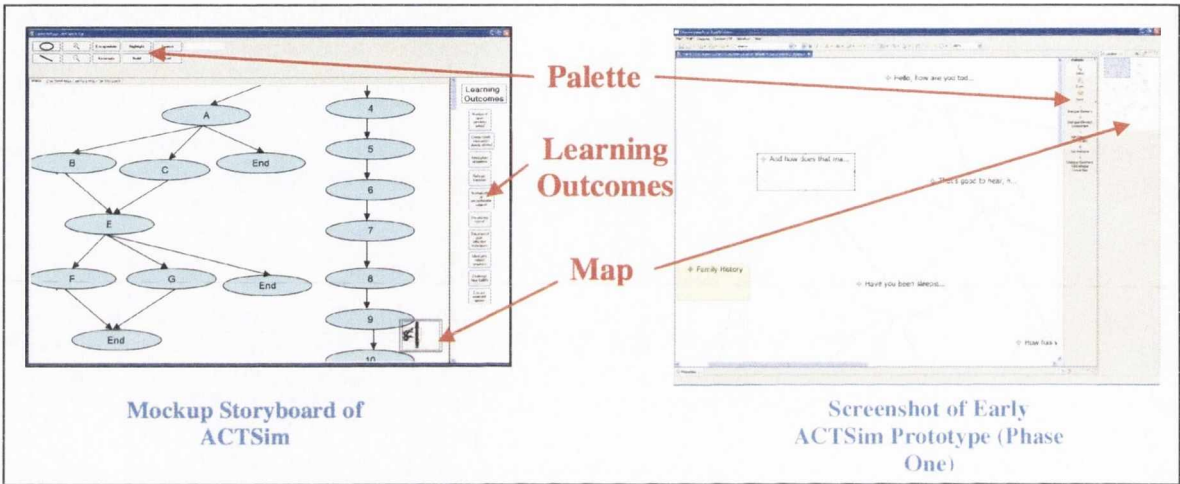


Figure 6.9: Comparison of Mockup Storyboard (Figure 6.7) and Screenshot of Early ACTSim Prototype (Gaffney, Dagger et al. 2007)

Further adjustments to the ACTSim prototype are also apparent when comparing the original storyboards from the initial cognitive walkthrough to the early screenshot of ACTSim. They illustrate how some of the features have changed over the course of the initial development due to the approach to implementation and as a result of contextual interviews. For example, the Learning Outcomes were originally to be constantly displayed within a window but this was deemed to be unnecessary with the Map feature taking a more prominent position. The comparison in Figure 6.9 also highlights some of the design aspects that have remained in ACTSim such as the approach to Dialogue Representation and Zoom functionality.

Some default settings related to the GMF were also found to be improvements to the initial storyboarding designs. For example, the location of the palette in Figure 6.9 has moved from the top of the interface to the side. The users, the same employed for the initial cognitive walkthrough, commented that they preferred this design as they could move the mouse cursor horizontally rather than vertically.

It should also be noted that at this stage of development there were a number of features yet to be implemented. For example, Dialogue Elements (Bubbles) do not have an assigned id property and the palette feature does not include icons for the different functionality.

Third Contextual Interview and Questionnaire: Sub-Dialogue Representation

The original storyboarding generated during the predictive evaluations included a feature for encapsulating sections of the dialogue (sub-dialogues). The objective of this feature was to reduce the complexity of the Dialogue Models and enhance its ease of comprehension. The implementation of this additional feature can be viewed in the diagram of the ACTSim Domain Model, Figure 6.10. The model includes a ‘SubDia’ component (Sub-Dialogue Node) which was designed to contain Dialogue Elements (Bubble Nodes).

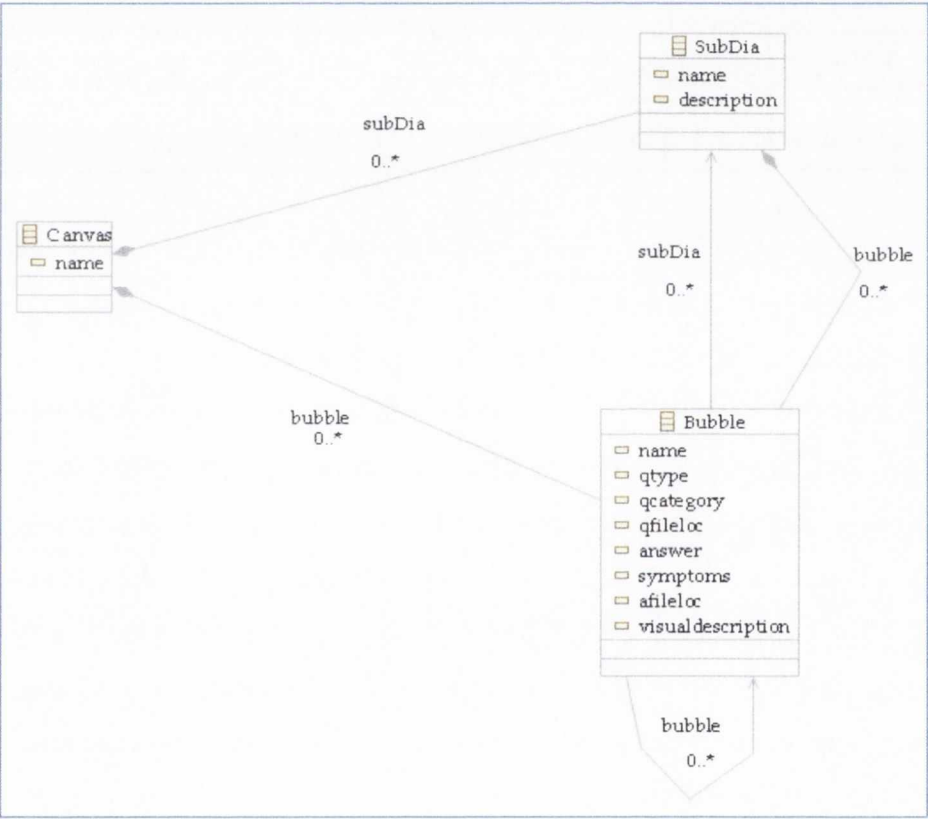


Figure 6.10: Phase One ACTSim Prototype Domain Model

The objective of the third interpretive evaluation was to examine and evaluate the Sub-Dialogue Representation. Two designs were developed and implemented, a tabbed approach and a compartment approach. Two users within psychiatry were asked to complete several short tasks with each design and their opinions were collected with a short questionnaire; see Appendix IX - Composition Tool Sub-Dialogue Feature Comparison. Both users found the

compartment approach more intuitive and user friendly than the tabbed approach. However, this initial evaluation was only completed with two users; both of whom had experience with authoring soft skill simulations so easily understood the concept of encapsulation. Subsequent evaluations found the concept of a Sub-Dialogue feature to be impractical when incorporated with large complex graphs (Dialogue Models). Evaluations completed later again, with a larger set of less experienced users, were inconclusive and discussions with users indicated that the concept was confusing. The following (fourth) contextual interview describes further why the Sub-Dialogue feature was not incorporated in the final design of the ACTSim composition tool. As such, this feature is not discussed in the design and implementation chapters.

Although the primary goal of the questionnaire associated with this evaluation was to investigate Sub-Dialogue Representation, one user also used it as an opportunity to add additional comments relating to the ACTSim system; copies of the resultant questionnaires can be accessed online (Gaffney 2006). Some suggestions for improving the ACTSim system included: the use of colour to separate sections of the Dialogue Model; a search feature to assist in locating Dialogue Elements; and a unique identifier for each Dialogue Element.

Fourth Contextual Interview: Dialogue Modelling Process

The Dialogue Modelling Process was only introduced formally to users at a late stage of the first phase of development; although, as the same users had participated in the cognitive walkthrough, they were aware of the process. The screenshot of ACTSim taken from the author's 2007 publication (Gaffney, Dagger et al. 2007) presents a model composed without the Dialogue Modelling Process where there is no clear main stem evident, as presented in Figure 6.11. Initial models composed by the users were relatively simple and so did not require a comprehensive modelling strategy; again evident in the same diagram. The formal introduction of the Dialogue Modelling Process, as described in 4.2.4 in the Dialogue Model Authoring Process subsection, was found by users to be very beneficial. Users commented that it assisted them in composing larger and more complex Dialogue Models.

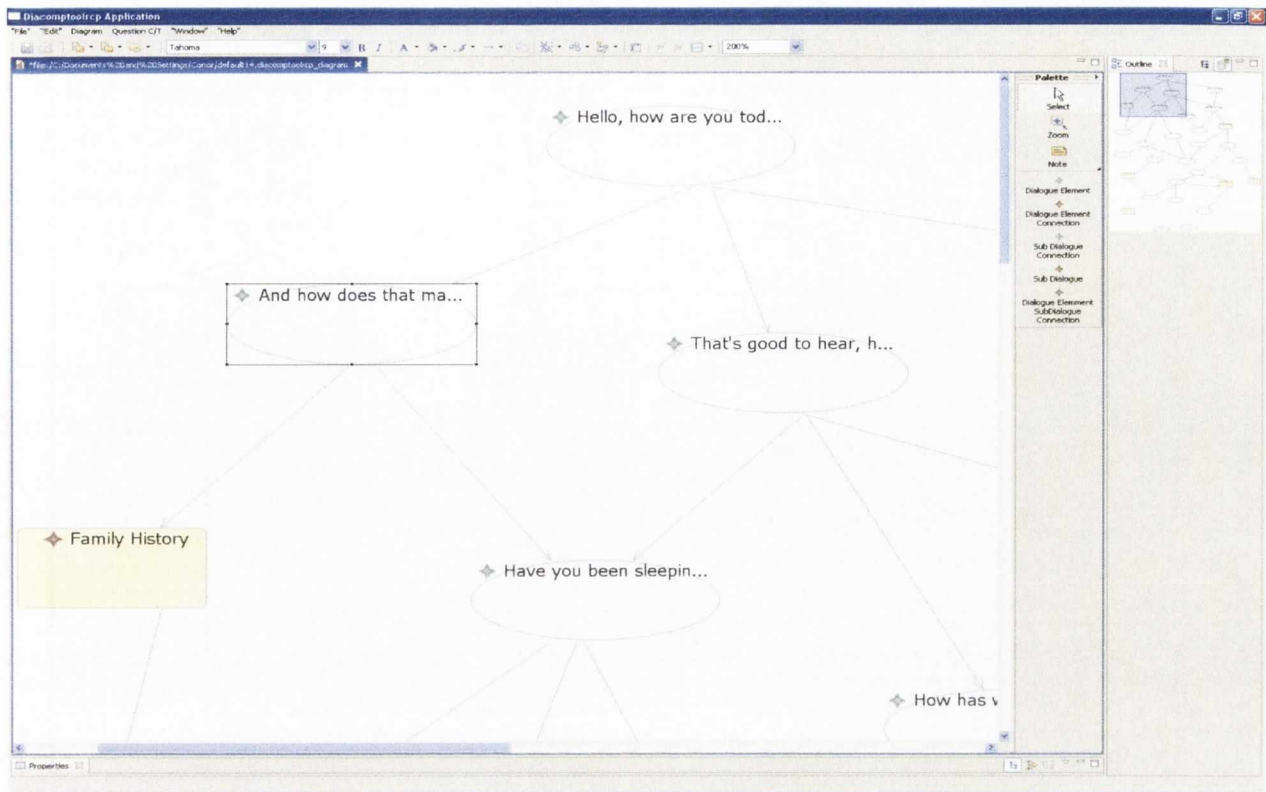


Figure 6.11: Screenshot of Early ACTSim Prototype (Gaffney, Dagger et al. 2007)

The introduction of the Dialogue Modelling Process coupled with the development of larger more complex Dialogue Models also resulted in a change in the ACTSim design. As described in the previous interpretive evaluation, a Sub-Dialogue feature was initially incorporated in the ACTSim prototype. The initial designs of how a Dialogue Model might be constructed, as outlined in the cognitive walkthrough storyboard in Figure 6.12, were somewhat naive in their expectation. The approach did not expect branches to reintegrate with the main stem. As such, the initial design allowed the concept of Sub-Dialogues to be possible. However, with complex models, branches would connect back into the main stem and interlink with one another. Encapsulating such branches proved to be impractical with such interconnectivity.

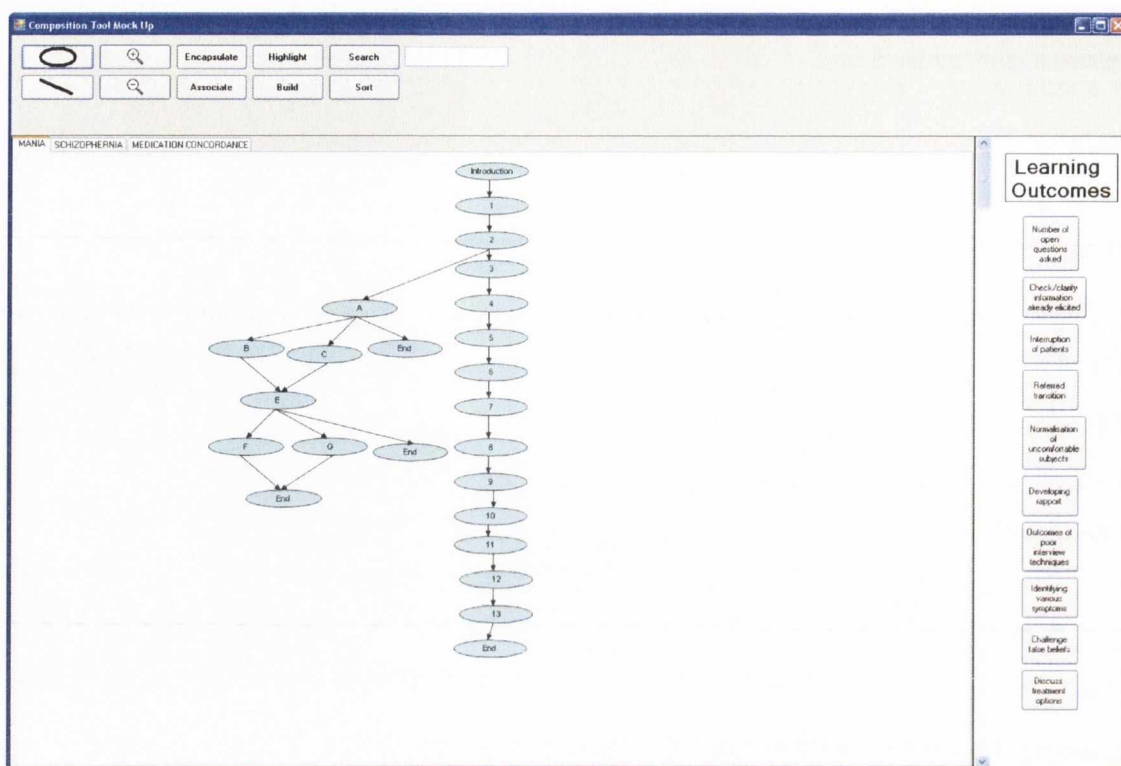


Figure 6.12: Storyboard Mockup of Dialogue Modelling Process

The first phase of interpretive evaluations resulted in an effective prototype which allowed subject matter experts to model dialogue and compose non-adaptive soft skill simulations. Although the system had at this point only been evaluated with a small number of users and limited to one domain the testing was in depth and rigorous. The interpretive evaluations prepared ACTSim for its first usability evaluation which is described in the following section.

6.3.3 Phase One Usability Evaluation

The first usability evaluation was completed at the end of the first phase of development. The ACTSim composition tool evaluated was a mature system and had already been through multiple evaluations. The approach and results of the first phase usability evaluation is described below.

Phase One Usability Evaluation Method

The first phase usability evaluation experiment was completed by five users from the Psychiatry group; psychiatry academics involved in lecturing and responsible for teaching soft skills, see Appendix VII – Evaluation User Groups for a more complete description. Each user was allowed a week to finish the assigned task and complete the evaluation

questionnaire; see Appendix X – First Usability Evaluation. The objective of this evaluation was to determine the ability of the ACTSim composition tool to address the key issues for authoring non-adaptive soft skill simulations; specifically examining Dialogue Representation, the Dialogue Model Authoring Process and the initially implemented Navigational Aids. The goal of this usability evaluation was to examine these areas with respect to their effectiveness, efficiency and usability (Hassenzahl, Beau et al. 2001). The following two sections describe the objectives and techniques associated with the first phase usability evaluation.

First Phase Usability Evaluation Objectives

1. Establish that acceptable levels of effectiveness and efficiency of basic functionality have been acquired. This was to be achieved by examining the **basic functions** such as save, open, close, undo/redo, minimizing/maximizing and insuring that they were all operating within a high standard.
2. Investigate if the users found the approach to **Dialogue Representation** was effective. This was to be accomplished by collecting data which illustrated if the users found the approach to Dialogue Representation that was incorporated was effective, efficient and user friendly. Specifically, determine if the users could fully understand the representation and create the dialogue models they wanted using the Dialogue Space and Palette.
3. Establish that basic operations used to create a **Dialogue Model** were efficient and user friendly. This was accomplished by measuring how efficiently the users could add Dialogue Elements, move/delete them, connect them and access/update their properties.
4. Investigate if the users found the **Dialogue Model Authoring Process** to be effective, efficient and user friendly. Could the users initially create the main stem and then add branches to it to create the dialogue model. Could the users complete these processes quickly and comfortably; could users manage multiple Dialogue Elements (moving and deleting branches); were all associated operation and features efficient and user friendly.

5. Investigate if the users found the **Navigational Aids** to be effective, efficient and user friendly. Were the Navigational Aids helpful in moving about the Dialogue Space and did they assist the users in finding particular Dialogue Elements. Did the features operate quickly and did the users find them comfortable to use. Zoom, Map, Search and Arrange features were examined during this experiment.

First Phase Usability Evaluation Techniques

As previously described, ACTSim is a user based application, with the author central to its development so a trial based user study was determined to be the most appropriate approach to evaluation. The evaluation process included the selection of five authors from a pool of psychiatry users and a specific series of stepwise tasks that the authors would complete in order to examine the composition tools different functionality. The authors were then to complete a questionnaire in order to evaluate each of the objectives previously outlined.

The evaluation objectives required authors to compose a Dialogue Model from beginning to completion to insure all relevant operations were executed and associated processes were followed. Such development could only be completed over several authoring sessions which required the evaluation task being completed by each user in their own time. To accomplish this ACTSim was installed locally on each user's computer (PC or laptop) and the evaluation task was completed in a real world setting. Authors developed Dialogue Models over a series of sittings as they would do in the real world. This approach not only insured that ACTSim was authentically evaluated but also highlighted issues that may not have been detected in a laboratory based evaluation scenario.

While the ACTSim composition tool was designed to be flexible enough to allow authors to compose a wide variety of soft skill simulations, a single domain was chosen to carry out this usability evaluation. The first pilot of the tool was used to support medical students in the department of psychiatry as part of the ADAPT (ADAPT 2010) project. Five authors were selected for the evaluation. Each author was a practicing psychiatric doctor, involved in the training of medical students with respect to their communication skills. This is an example of an authentic evaluation which incorporates the target audience of the composition tool. Each author was given the task of developing an initial interview that would take place between a

doctor and a patient. The objective of this dialogue was to create a simulation which would be used to simulate an initial introductory interview with a patient.

The task required an author to create a Dialogue Model which included eight topics and consisted of at least forty Dialogue Elements. The task also required the author to use the Main Stem branching approach for authoring and that the authors should utilise the different features implemented in the composition tool. Once the task was concluded the authors completed a questionnaire designed to gather data regarding the key objectives of the evaluation as previously outlined. An informal interview was also conducted with each author after the questionnaires were collected as part of the interpretive evaluation process.

Phase One Usability Evaluation Results

The first phase usability evaluation was completed by authors over the space of a week on their personal computers. They were required to compose a soft skill simulation model and complete a questionnaire. With the task completed and questionnaires collected the data was examined. The data was grouped so as to address the respective issues, the results of which are detailed below. As only five users were employed to complete this experiment all data associated with this set of results, while indicative, cannot be taken as statistically significant. This is similarly true for the second phase usability evaluation.

Basic System Functionality

All but one of the basic system functions behaved as expected. The Save function was found not to operate correctly with 75% of the users disagreeing with the statement "*Saving files was an efficient process*". Further testing discovered the fault was with the version of the GMF platform being employed; an issue which was easily corrected. An additional fault with the system install was also discovered with during the usability evaluation; this issue was also a fault with the GMF platform and was also rectified with a software update. Either of these issues may not have been discovered if the usability evaluation had taken place in a laboratory setting as they were dependent on the users' Windows operating system versions.

Dialogue Representation

The feedback regarding the Dialogue Representation was generally very positive with all authors successfully composing a Dialogue Model and utilising the associated functionality.

In terms of efficiency and user satisfaction the dialogue representation was very well received with one author commenting *“It represents flow of dialogue from one exchange to the next”*. The same user also commented that it *“does allow complex options to be logically represented”*. Another user commented *“[ACTSim] allows for multiple connections enabling the scripting of multiple choice interviews”*. Additional comments were that the tool allowed users to model the questions they wanted with *“no problem”* and that there was *“lots of scope”*.

All but one of the operations associated with the Dialogue Representation efficiency and user satisfaction returned an average between three and four in a Likert scale of one to four where four is the most favourable result⁴. Figure 6.13 displays a mean view of the efficiency and user satisfaction of the Dialogue Representation. The x-axis of the graph in Figure 6.13 describes areas related to the efficiency and user satisfaction of Dialogue Representation within ACTSim; the y-axis of the graph in Figure 6.13 displays the associated Likert scale.

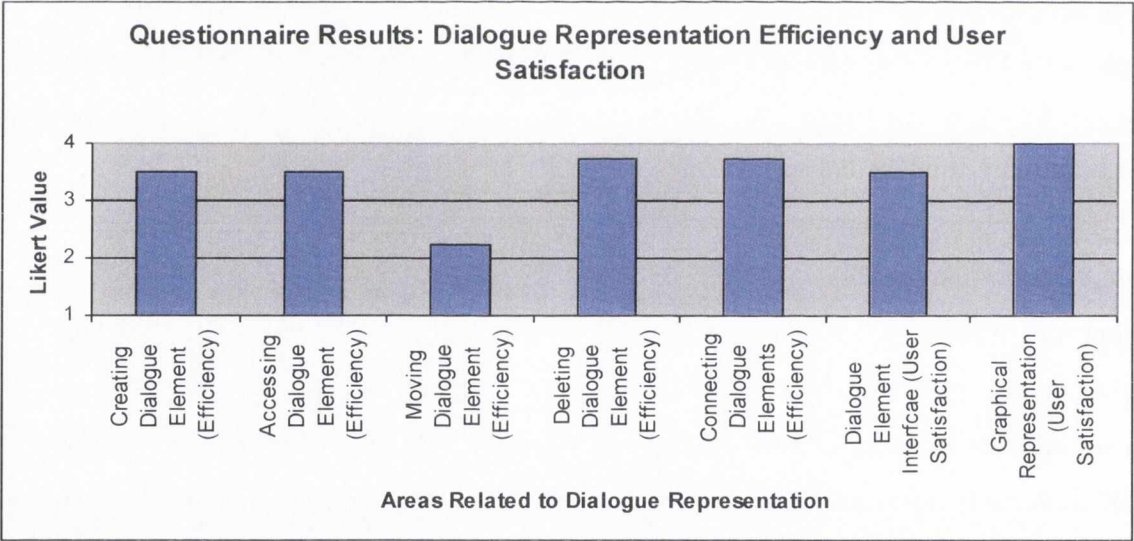


Figure 6.13: Correlated Mean Data of Efficiency and Usability of Dialogue Representation

The only negative response with respect to the user satisfaction and effectiveness of Dialogue Representation is highlighted by the third bar in the graph of Figure 6.13; Moving Dialogue Elements (Efficiency). Further investigation showed that this aspect of Dialogue Representation returned results lower than expected (mean Likert scale value below 3) due to

⁴ Four being the most favorable result and one being the least favorable result continued for all Likert scales for the remainder of this usability evaluation

some users' computer modality. Specifically, some users within the usability evaluation completed the task using only a laptop touchpad and found moving Dialogue Elements to be difficult. This is not an uncommon problem with systems that incorporate extensive drag and drop functionality. This is another issue that may not have been discovered within laboratory based evaluations; subsequent evaluations requested users to employ the use of a mouse.

While efficiency and user satisfaction feedback regarding the Dialogue Representation was very positive the effectiveness of results were inconclusive. This was not due to any failing of the composition tool but instead was a fault of the usability evaluation and its interpretation of 'effectiveness'. The initial interpretation of effectiveness, as related to Dialogue Representation, was viewed as a related tasks being 100% completed; with anything less considered to be non-effective. A much improved and comprehensive interpretation of Dialogue Representation effectiveness was used to re-evaluate the system in the second usability evaluation. The initial evaluation did however successfully test that it was possible to complete all associated Dialogue Representation operations; 100% of authors were successful in creating, editing, moving and connecting Dialogue Elements.

Dialogue Model Authoring Process

Feedback regarding the Dialogue Model Authoring Process was also very favourable. 100% of the users found the process to be effective. All users successfully followed the Dialogue Modelling Process initially creating the main stem and subsequently adding branches to expand the dialogue. One user commented that it was "*easy to create scenarios*" employing the Dialogue Model Authoring Process while another wrote that there was "*logic to the authoring process*". There were however some operations that were associated with the process that were not utilised by all of the users. For example, three of the five users did not successfully delete multiple Dialogue Elements (a single branch) with one user commenting "*I'm sure deleting multiple elements would have been no problem – [I] just didn't do it*". Similarly, one of the users did not attempt to move multiple Dialogue Elements while creating their Dialogue Model. The users who did complete these operations found their operation to be effective. The reason for these procedures not being utilised by users may have been that the users were only creating relatively small dialogue models (20 to 30 Dialogue Elements) and so were not often required to delete or move an entire branch of their Dialogue Model, although they were instructed to as part of their task.

Users also found the Dialogue Model Authoring Process to be very efficient; 100% of users either agreed or strongly agreed that creating the main stem and branches “*was an efficient process*”. All functionality associated with this process also scored an average between three and four on the on the Likert scale identical to the scale previously described; as presented in Figure 6.14. It should be noted that any user that did not complete the associated functionality were removed from the mean calculation. Users commented that the “*main stem was completed quickly*”.

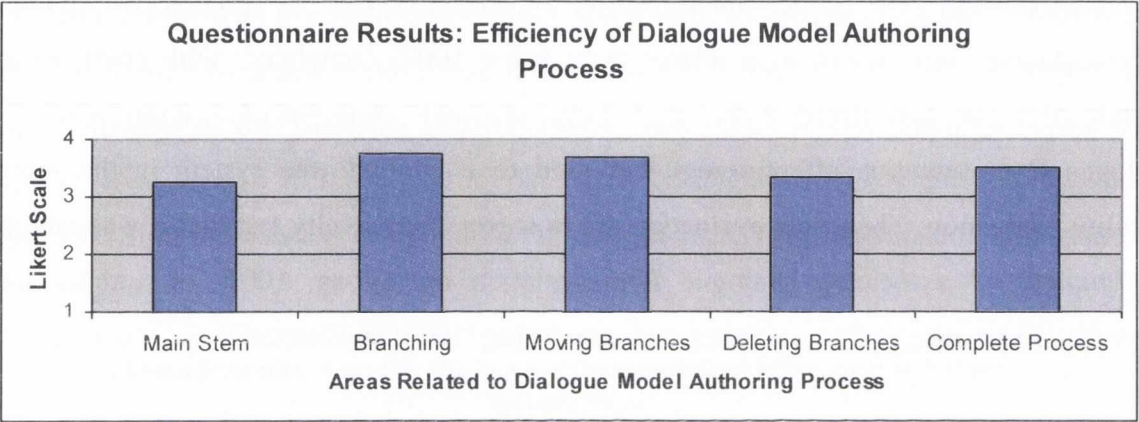


Figure 6.14: Correlated Mean View of Efficiency of Dialogue Model Authoring Process

The Dialogue Model Authoring Process was also found to be very user friendly. Figure 6.15 presents a mean view of the data correlated which described the user satisfaction associated with the Dialogue Model Authoring Process. The graph presented in Figure 6.15 follows a similar structure to previous graphs whereby the areas related to the Dialogue Model Authoring Process in the x-axis and the Likert scale is displayed in the y-axis. One user commenting that it was the Dialogue Model Authoring Process was “*easy to follow*”.

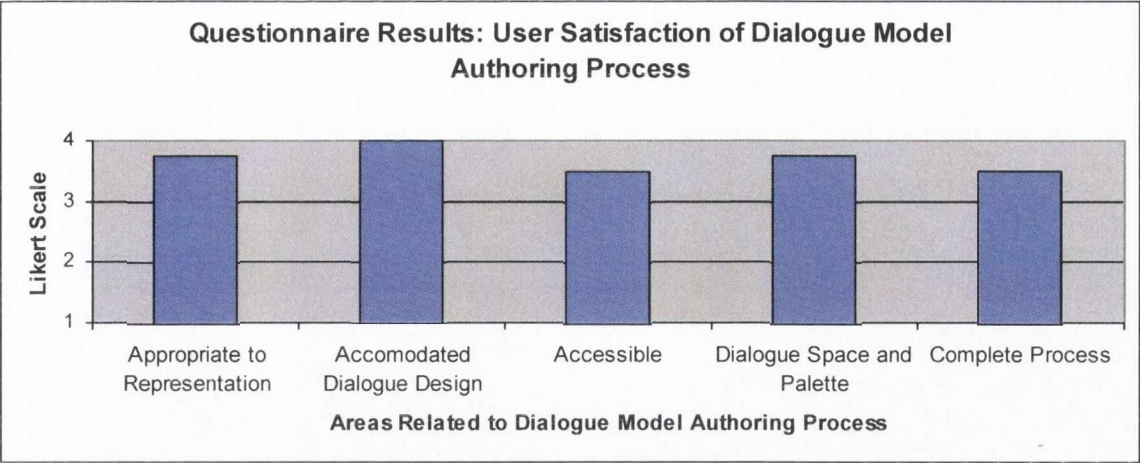


Figure 6.15: Correlated Mean View of User Satisfaction of Dialogue Model Authoring Process

Navigational Aids

Of the several aspects of ACTSim examined in this evaluation the assessment of the Navigational Aids⁵ returned the most inconclusive results. This was due to the relatively small number of users who utilised the Navigational Aids during the course of completing their allocated task. Although instructed to use each of the four Navigational Aids being evaluated at least once, only the Map feature was utilised by all of the users. The remainder of the Navigational Aids were only used by two or three of the users. The reason for the poor uptake may have been due to the small nature of the Dialogue Model being composed by the users. While Navigational Aids are most certainly needed for large Dialogue Models (>100 Dialogue Elements) they are not as essential if navigating a relatively small composite. The results accumulated for the navigational aids that were utilised were very positive. Although the data does not convey any indicative conclusions, with the exception of the Map feature, Figure 6.16 presents the correlated data of the efficiency of each Navigational Aid evaluated. Each of the Navigational Aids scored an average between three and four in the Likert scale in terms of their efficiency. The Navigational Aids were rigorously re-evaluated in the second usability evaluation which incorporated a different approach to ensure each Navigational Aid was utilised by the users.

⁵ Navigational Aids evaluated include Map, Zoom, List, Arrange

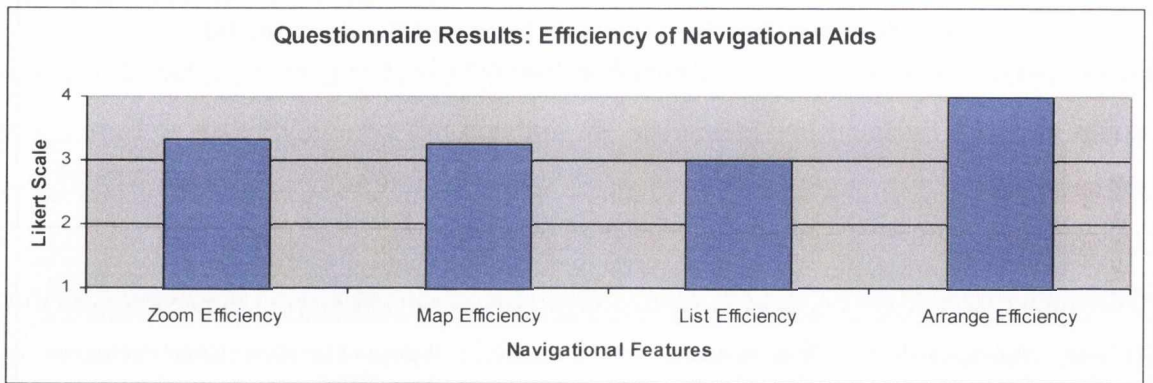


Figure 6.16: Correlated Mean View of Efficiency of the Navigational Aids

Although not all of the Navigational Aids were effectively examined the Map feature was utilised and its associated data correlated. 100% of the users thought the map feature was effective and user friendly. As presented in the Figure 6.16, all the users agreed or strongly agreed that the Map feature was efficient with one of the users commenting that it *“allows you to move quickly over the Dialogue Space as more and more questions are added”* and that it made it *“easy to move around”*.

This usability evaluation also incorporated a question regarding how the users might view encapsulation of Sub-Dialogues; a feature described in the phase one interpretive evaluations but which had been removed for the first usability evaluation. Users were asked if they would Disagree, Somewhat Disagree, Somewhat Agree or Agree with the following statement: *“The ability to hide sections of the dialogue model would aid the navigation of the dialogue space”*. The results were inconclusive with two users Somewhat Disagreeing, two users Somewhat Agreeing and one user Agreeing with the statement; one user commented *“don’t see why [encapsulation] would”* assist navigation.

6.3.4 End of Phase One Reflections

The results of the first usability evaluation were very positive. Comments from the evaluation participants described the tool as *“intuitive”* and *“flexible and intuitive”*. Evaluation users also said it was *“easy to use”* and that the *“end result is excellent”*. As expected the dialogue representation was key in the completed initial prototype. What was not initially realised by the author at the beginning of the prototypes development was that the dialogue model authoring process and navigational aids would become more prominent because of the selected dialogue model representation. By employing a graph based approach to represent

dialogue, which allowed authors to freely place Dialogue Elements and construct the model without restriction, the dialogue model authoring process became more important than initially expected. As outlined previously, the evaluation of the main-stem branching approach in the usability evaluation were very good.

The graph based dialogue representation approach allows large complex models to be composed. This is very beneficial as intricate and realistic/challenging soft skill simulations can be composed. It was not initially realised by the author just how elaborate and expansive these models would become. The navigational aids were originally thought to be for simple ease of use, a convenience that would only be used occasionally. However, during development, the navigational aids took a more prominent role in the ACTSim composition tool. The navigational aids, particularly the map feature, were used a great deal. The map feature allowed authors to not only navigate the models they created but to also assist them in managing the overall structure as it became a point of reference. With the exception of the map feature, the results of the navigational aids in the usability evaluation were somewhat inconclusive; future evaluations would be more decisive.

The initial design of the second phase prototype was dependent of several contributing factors: the prototype developed in first phase of development, the first usability evaluation, the state of the art survey and the resultant design requirements. The approach to the application and representation of adaptivity was key to this research. It needed to be effective, graphical and accessible to the authors. Furthermore, adaptive dimensions incorporated needed to be suited to a soft skill simulation learning environment. Additional functionality was also to be included so as to extend the existing design with a complete authoring process, educational framework and model validation functionality. Details of the initial design of the second phase prototype were captured in storyboarding and second predictive evaluation.

6.4 Phase Two of Development

This section presents the evaluation goals, methods and results associated with the second phase of development. The system employed at the beginning of phase two is the same system that was evaluated during the first usability evaluation; i.e. no additional functionality or features were added between the two phases. As part of the first evaluation in this phase of

development, a predictive evaluation, new features and functionality were described in a storyboard format so as to identify how they could be offered to users.

This section is divided into three subsections; the first subsection describes the second predictive evaluation completed, the following subsection examines the interpretive evaluations completed in the second phase of development and the final subsection presents the second usability evaluation which was completed at the end of the second phase of development.

6.4.1 Phase Two Predictive Evaluation

Similarly to the predictive evaluations completed at the beginning of the first phase of development, cognitive walkthroughs were employed at the beginning of the second phase of development. The objective of the phase two predictive evaluation was to gather feedback from users so as to incrementally improve storyboards which augmented the system. The focus of this evaluation was the incorporation of features used for applying and representing adaptivity. The cognitive walkthroughs needed to capture, examine and evaluate how these features would be incorporated into the existing ACTSim prototype produced and evaluated in the first phase of development. Similarly to the predictive evaluations completed in the first phase of development the evaluation process involved both the feedback from the storyboards and the iterative improvement of the outlined functionality. Evaluation metrics and techniques for gathering data were also similar to the first predictive evaluations. An outline of the evaluation method and results of the predictive evaluation completed in phase two of development are described below.

Phase Two Predictive Evaluation Method

The cognitive walkthrough completed at the beginning of the second phase of evaluation followed a very similar pattern to that of the cognitive walkthrough completed in the first phase of development. The first step was to initially develop paper and pencil storyboards which were modified and augmented through multiple iterations of cognitive walkthroughs based on users' feedback. Once no usability issues could be predicted by participants the predictive evaluation was complete. For clarity, important interface components of the pencil and paper storyboards were transcribed to an electronic medium using Microsoft PowerPoint (PowerPoint 2011).

The major difference between the first and second phase cognitive walkthroughs was that the initial pencil and paper storyboards were designed on top of a facsimile (printed screenshots) of the ACTSim prototype produced in the first phase of development; a sample collection of the initial phase two storyboards can be found online (Gaffney 2007).

The main focus of this cognitive walkthrough experiment was to examine how adaptivity might be represented and applied within the authoring tool. Some of the functions and redesigns to the initial prototype included:

- Adaptivity property interface designs
- Dialogue Element nodes redeveloped to include a compartment for Triggers
- Trigger interface designs
- Dialogue Element interface redesigned to include adaptivity tagging properties
- A Highlight function cognitive walkthrough

The second set of walkthrough evaluations were initially completed with experts in the area of eLearning adaptive hypermedia composition ensuring that the design concepts of personalisation were effective. This was followed by cognitive walkthroughs with authors that had no previous experience with adaptivity and personalisation; specifically two authors from the Psychiatry user group. This was to insure that the approach to adaptivity was comprehensive to those without AH experience.

Phase Two Predictive Evaluation Results

Similarly to the first cognitive walkthrough, the combination of evaluation and development between evaluations within the second phase predictive evaluation produced detailed storyboards. Figure 6.17 and Figure 6.18 display two of the storyboards (mockups used to illustrate interface functionality) that were derived at the beginning of the second phase of development. Figure 6.17 presents a storyboard mockup of the interface used for adding and editing the list of roles (adaptive dimension) that can be used for tagging. Figure 6.18 presents a mockup of the Assessment Trigger Properties interface accessible by double clicking a Trigger node in the Dialogue Element compartment.

Simulation Roles

	Description
Doctor	_____
Nurse	_____
Administrator	_____

Figure 6.17: Second Cognitive Walkthrough Storyboard - Soft Skill Simulation Roles Interface

Assessment Trigger Properties

Name:

Description:

URL:

Learning Outcomes:

Figure 6.18: Second Cognitive Walkthrough Storyboard - Assessment Trigger Properties Interface

Two approaches to personalisation had been postulated and it was the main objective of the second cognitive walkthrough to incorporate both of these approaches into the design of the existing authoring tool. The approach for Tagging Dialogue Elements so as to implement adaptivity was relatively straight forward to incorporate as very little of the existing composition tool design needed to be adjusted. The Trigger concept however required extensive thought as it needed to be intuitive, graphical, unobtrusive and simple to use. It required a redesign of the Dialogue Elements so they contained a compartment which could contain the different Triggers; the design of the associated component interfaces attempted to follow a similar pattern to the existing interfaces and were easily designed.

6.4.2 Phase Two Interpretive Evaluations

A number of interpretive evaluations were completed during the second phase of development using the ACTSim prototype which incorporated designs and redesigns initially

based on the storyboards developed in the second cognitive walkthrough. The objective of the interpretive evaluations was to closely examine and evaluate new ACTSim functionality with respect to system usability. The new functionality and features not only needed to address the system requirements but also needed to be incorporated into the existing system. The feedback from the users allowed the composition tool to be incrementally improved and augmented. The evaluations metrics employed in this phase of development were similar to those employed in the first phase of development focusing on qualitative feedback based on effectiveness, efficiency and user satisfaction. The phase two interpretive evaluations employed four contextual interviews, a contextual interview which also included a questionnaire and a semi-structured interview. The method and results of the interpretive evaluations completed in phase two of development are described below.

Phase Two Interpretive Evaluations Method

During phase two of the development process there were multiple interpretive evaluations conducted with several different user groups. Similarly to the interpretive evaluation conducted in the first phase of development these evaluations were participative and cooperative in nature. They also followed a similar approach and incorporated a pattern which included software install, allowing the user time to engage with the software, evaluation of the system and software redesign/update.

As there were multiple iterations of interpretive evaluation with multiple user groups which cross examined and re-examined ACTSim features it would be impractical to present a detailed account of each evaluation. However, the underlying objectives of these evaluations were to integrate and evaluate the application and representation of adaptivity in the ACTSim composition tool; as outlined previously in this thesis. In addition, the phase two interpretive evaluations were used to examine the pedagogical framework, the authoring process and the verification functionality.

Phase Two Interpretive Evaluation Results

Adaptivity application and representation, as described in 4.2.3, was the first feature set to be implemented and integrated with the ACTSim prototype developed in the first phase of development. To some degree, all of the contextual interviews employed discussion relating to these features and concepts. Furthermore, the second interpretive evaluation, with the

customer care user group, included a semi-structured interview which examined both Tagging and Triggers, Appendix XI – Customer Care Directed Discussion. The results of the semi-structured interview regarding the approach to tagging were positive with users commenting that the approach was accessible and user friendly. However, the users found that the design of the Triggers was somewhat ineffective.

The initial approach to implementing Triggers, at the beginning of the second phase of development, was with a ‘graph’ based approach. This resulted in Triggers being represented as free floating nodes within the Dialogue Element compartment; as described in the cognitive walkthrough storyboards and section 5.3.3 ACTSim Graphical Editor. However, prior to the interpretive evaluation that included the semi-structured interview (the second interpretive evaluation in the second phase of evaluation), the compartment which stored the Trigger was redesigned and implemented with a ‘list’ approach as a possible alternative; this resulted in Triggers being realised as a list within a Dialogue Element compartment. The ‘list’ approach allowed for a ‘neater’ structure but resulted in Triggers being fixed and non-transferable to other Dialogue Element compartments. The ‘list’ approach was evaluated with the semi-structured interview with rather negative results; users commented that they became frustrated with the fixed structuring of the list (not being allowed to move Triggers from compartment to compartment) and the small icons which resulted with the approach (double clicking the Triggers to access their properties was less user friendly). The same user group (customer care) had previously used the ‘graph’ approach to Triggers which was preferred and so was reinstated. The following interpretive evaluation, completed with the same user group, included a questionnaire, see Appendix XII – Customer Care Short Questionnaire, which re-examined Triggers within the system. The results of the evaluation were positive and the completed questionnaires can be found online (Gaffney 2008).

Both the semi-structured interview and questionnaire were also used to re-evaluate features and concepts that had been initially implemented and evaluated in the first phase of development with a different user group. Areas re-evaluated include Dialogue Representation, Navigational Aids and the Dialogue Modelling Process. The results of these re-examinations were also very good.

The overall results of the interpretive evaluations with regard to adaptivity can be seen as design changes in the system; as illustrated in Figure 6.19 and Figure 6.20. The storyboard mockup in Figure 6.19 presents a comparison of a screenshot of the ACTSim prototype from phase one of development to a screenshot of the ACTSim prototype from phase two of development. The screenshot of the ACTSim prototype from phase two includes a compartment for Triggers (with the ‘graph’ based approach) as well as an Assessment Trigger. Figure 6.20 presents a comparison of the Roles (adaptive dimension) interface storyboard and its implementation in the ACTSim system. The mockup storyboard in Figure 6.20 had been created at the beginning of phase two of development so its design was much more informed than the previous initial storyboarding and therefore very little adjustment was required during the interpretive evaluations.

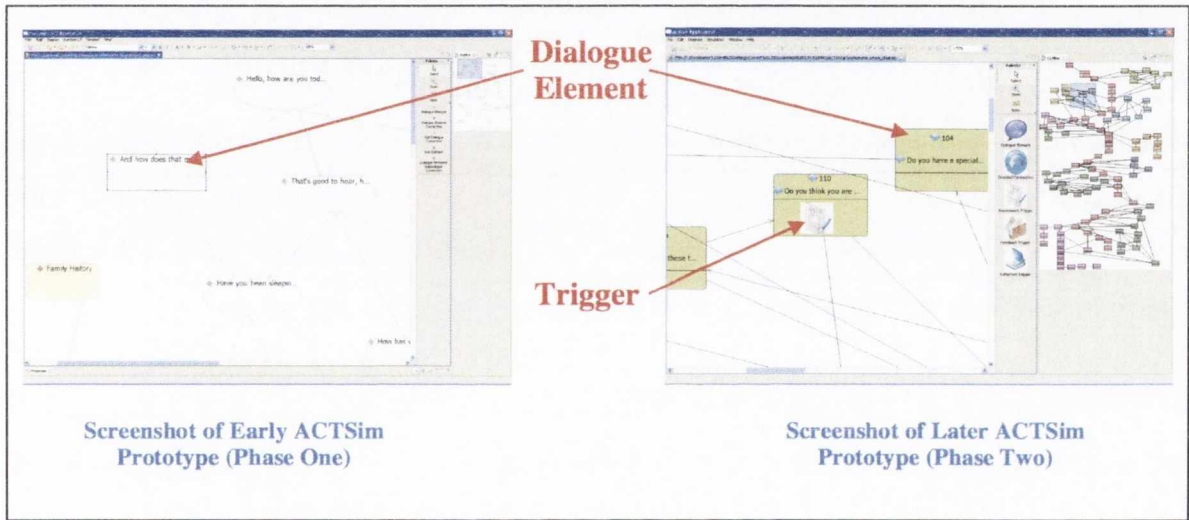


Figure 6.19: Comparison of Screenshot of ACTSim Prototype from Phase One to Screenshot of ACTSim Prototype from Phase Two

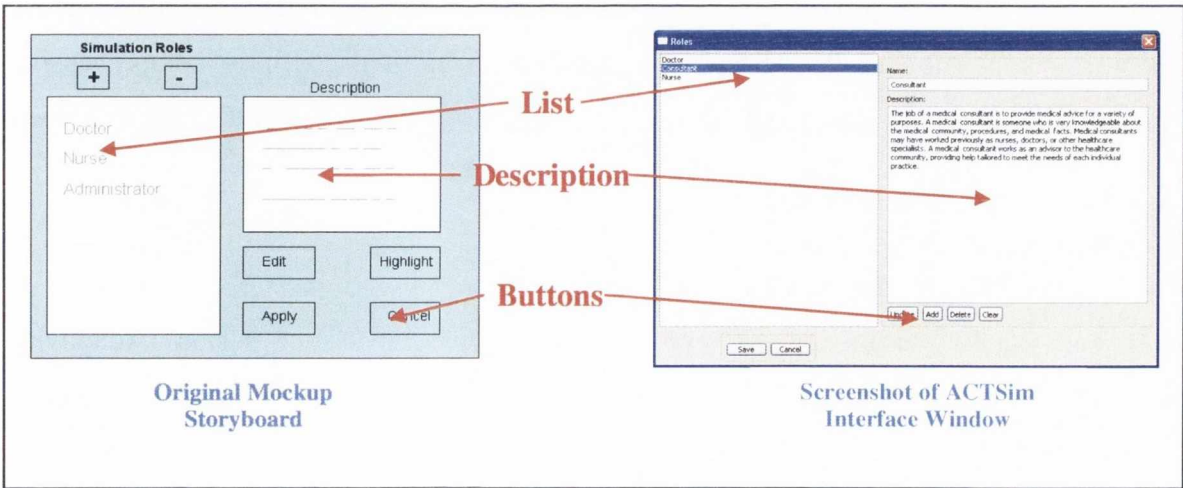


Figure 6.20: Comparison of Roles Interface Storyboard (Figure 6.8) and Roles Interface Implementation (Appendix IV – Component Interfaces - Roles)

Additional features introduced and examined with second phase interpretive evaluations include the pedagogical framework (section 4.1.4), complete authoring process (section 4.1.3) and verification functionality (section 4.1.6). Implementation of these features was rather straightforward and the corresponding evaluation results were very positive. Integration was assisted by the fact that these features were introduced at a later stage of development and so were much more informed as they could be based on previous feature design and implementation. For example, applying tags for adaptivity was developed and evaluated earlier in the second development phase. The verification Script feature followed a process of selecting Dialogue Elements which is very similar to that of applying tags. Similarly, the function for highlighting adaptivity tags, which was also developed and evaluated in the earlier stages of second development process, informed the design of the Search function which was introduced later.

The second phase of interpretive evaluations resulted in a robust system that addressed all the design requirements outlined in chapter four. The system had been tested and evaluated with multiple user groups and was prepared for its second usability evaluation which is described in the following section.

6.4.3 Phase Two Usability Evaluation

The second usability evaluation was completed at the end of the second phase of development. All of the design requirements described in chapters four and five had been incorporated into the system. The ACTSim composition tool had already been through multiple evaluations in two phases of evaluation and trialled with multiple users across different domains. As such the system was prepared for its final and most comprehensive evaluation. The approach and results of the second phase usability evaluation are described below.

Phase Two Usability Evaluation Method

The second usability evaluation focused on the adaptive aspects of design (Adaptivity Application and Representation) such as Tagging the Dialogue Model and adding Triggers. This evaluation also examined the complete Authoring Process (used to compose the entire simulation), Pedagogical Framework, Model Verification and re-examined Navigational Aids that had not been adequately evaluated in the previous usability evaluation. Although

examined in the first usability evaluation the effectiveness of the Dialogue Representation was also re-evaluated.

The second usability evaluation was also used as an opportunity to evaluate the flexibility of ACTSim. The composition tool had been successfully examined in several domains during predictive and interpretive evaluations but the first usability evaluation had only employed users from the Psychiatry group. The second usability evaluation employed five users with a single user from the following domains: Psychiatry, Exam Stress, Healthcare (Social Interaction for Adolescent with Autistic Spectrum Disorders), Healthcare (Informative Adolescent Instruction) and Education.

In order to evaluate adaptivity within ACTSim it was necessary for the users from these domains to have previously developed a Dialogue Model that contained no adaptivity; i.e. the models were not tagged and no Triggers had been added. Employing a pre-developed Dialogue Model in the usability evaluation allowed users to apply adaptivity to a model that they were familiar with. All participants completed the second usability experiment within two hours.

The following two sections describe the objectives and techniques of the second phase usability evaluation.

Second Phase Usability Evaluation Objectives

1. Determine the effectiveness of the **Dialogue Representation** within the composition tool. This was to be accomplished by collecting data which illustrated if the users found the approach incorporated to Dialogue Representation was effective. Specifically, could the users fully understand the representation and create the dialogue models they wanted using the Dialogue Space and Palette.
2. Investigate, in terms of effectiveness, efficiency and user friendliness, the **Adaptivity Application and Representation** of ACTSim. Specifically, could the users fully understand how Tagging a Dialogue Model and adding Triggers would affect the adapted model; could the users execute associated functions in a timely fashion; were the users able to easily execute the associated functionality. These objectives were

addressed with respect to both Tagging (creating Tag properties, Tagging single Dialogue Elements, Tagging multiple Dialogue Elements, etc.) and Triggers (easily and quickly add Triggers, move Triggers, delete Triggers access/edit Triggers properties, etc.).

3. Investigate the **Authoring Process** with respect to its effectiveness, efficiency and user satisfaction. Specifically, could the users understand and easily follow the process; was each step of the process clear and concise; could the users complete an iteration of the process in a timely fashion; and could the users easily access the Authoring Process within the composition tool.
4. Investigate the **Pedagogical Framework** with respect to its effectiveness, efficiency and user satisfaction. Specifically, could the users easily understand the purpose of both the Pedagogical Approach and the Learning Outcomes; could the user create and populate these aspects of their composition efficiently; and did the users find their interaction with this part of the user friendly and easy to interact with.
5. Investigate the **Navigational Aids** (with the exception of the Map feature) with respect to effectiveness, efficiency and user satisfaction. Specifically, examine under these criteria the Zoom, Search, List and Arrange features; the Search feature had been added to ACTSim since the first usability evaluation.
6. Investigate the **Model Verification** functionality with respect to their effectiveness, efficiency and user satisfaction. Specifically, Script, Highlight and Validate features.

Second Phase Usability Evaluation Techniques

Similarly to the first usability evaluation, the second usability evaluation was completed as a trial based user study. There were however a number of differences in the techniques employed by the second usability evaluation. Unlike the first usability evaluation, users were not required to compose a complete Dialogue Model from beginning to end. This allowed the second usability evaluation to be conducted within a laboratory environment. While conducting the initial usability evaluation in the real world was both necessary and beneficial it was also important to conduct an experiment within in a formal setting:

- A laboratory setting allows data to be accumulated through several different means including video capturing and question recording as the users interact with the interface. Data can be collected in a more objective manner than simply using questionnaires which can be subjective.
- It is possible to ensure that a user completes all parts of the task they have been assigned; such as executing all relevant operations (as was an issue with the first evaluation).
- It is possible to ensure that all users complete their task within an identical environment; there is no external interference and each user utilises identical equipment such as their form of modality (mouse versus touchpad for example).

There were three factors to be considered when selecting what methods of data gathering should be used in the second usability evaluation. The first factor was the suitability of the method of data gathering in relation to the type of data that was to be gathered. For some system requirements that were being examined video capture was the most suitable option; for others questionnaires were the best approach. The second factor to be considered was ethical restrictions, an overview of which is presented in Appendix VI – Ethical Approval. The third factor was the cost (time and resources) of using the data gathering methods in relation to the importance of the requirement being evaluated; some aspects of the design within the second usability evaluation were identified as being of greater importance than others.

The ACTSim requirements identified as being key to the systems operation within the second usability evaluation were the Dialogue Representation effectiveness and the Adaptivity Application and Representation usability. Additional evaluation tools were employed for analysing both of these requirements. While other requirements are also important within the composition tool, ACTSim could feasibly still allow adaptive soft skill simulations to be composed without these features; Dialogue Representation effectiveness and the Adaptivity Application and Representation are essential components within the system.

A short test which examined a user's understanding and memory retention was used to aid the evaluation of the effectiveness of both the Dialogue Representation and Adaptivity

Application and Representation. Furthermore, video capturing of the computer screen was used to aid the evaluation of the Adaptivity Application and Representation efficiency and user satisfaction. A questionnaire was also used to examine these and the other system requirements being evaluated. Additionally, the laboratory setting allowed questions asked by the users to be recorded during short training session at the beginning of the evaluation and throughout the experiment; this approach could have highlighted design flaws or effectiveness issues within ACTSim.

Best efforts were made to approximate an authentic usability evaluation laboratory; the room used to conduct the experiment was designed to be sound proof and had no windows from which distractions might be caused. While it was not possible to video record users due to ethical approval issues, a video capture of their computer screen was allowed as it insured anonymity.

Appendix XIII - Second Usability Evaluation - Design Document contains a complete account of the second usability evaluation; the experiment consisted of the following steps:

- Participant Information Sheet: each user was presented with a document that explained the purpose of the experiment, their participation and associated legislation.
- Consent Form: each user was required to sign a consent form agreeing to participate in the experiment.
- Pre-Evaluation Questionnaire: each user was required to complete a short questionnaire which outlined their background and experience; Appendix XIII - Second Usability Evaluation - Pre-Questionnaire.
- Training: each user completed a short training session which detailed ACTSim's design principles and covered every aspect of its operation including all associated processes.
- Proficiency Task: each user was required to complete a short test which demonstrated that they had reached an adequate level of proficiency; this ensured that they had reached a sufficient level of competency to complete the main evaluation task.
- Main Task: each user was required to complete the main task (details of which can be found below). As a user completed each step of the task they were required to tick a box indicating they had completed the step; this was to insure that all parts of the task were completed by the user.

- Test: each user was required to complete a short test which examined their understanding of aspects of ACTSim's purpose and principles. It focused on the effectiveness of the Dialogue Representation and the adaptivity features; Appendix XIII - Second Usability Evaluation – Test.
- Post-Evaluation Questionnaire: each user was required to complete a questionnaire. The questionnaire was used to gather data which described all areas of the system but particularly focused on user satisfaction; Appendix XIII - Second Usability Evaluation - Post-Questionnaire.

The main task was divided into two sub-tasks. The first sub-task required the users to complete several iterations of the Authoring Process with the Dialogue Models that they had previously developed. This task included the following steps: examine and evaluate the model; create suitable tagging properties (users were allowed to choose from role, subject or category; whichever property they felt was most suitable to their simulation); apply the tagging properties to sections of their Dialogue Model; add at least one of each of the Trigger types to the model; edit each of the Trigger's properties appropriately. Employing a step-by-step approach to the task allowed users to evaluate their model, apply adaptivity and ensured they followed ACTSim's authoring process. A final step in the first task required users to independently complete a full iteration of the authoring process.

The purpose of the second sub-task was to evaluate the Navigational Aids and Validation features used in ACTSim. The second sub-task employed a different Dialogue Model to the first sub-task. The Dialogue Model employed in the second sub-task was a model that was unfamiliar to the users; it was complex and very large (>100 Dialogue Elements). The users were required to locate specific Dialogue Elements based on determination within the model, specific tags and Statement properties. The second sub-task was presented in series of steps, each step associated with a separate Navigational Aid or Validation feature. Employing a Dialogue Model that the users were unfamiliar with ensured that users were not predisposed to the location of any of the Dialogue Elements so requiring them to correctly use each Navigational Aid or Validation feature.

Phase Two Usability Evaluation Results

The second phase usability evaluation was completed by users in a laboratory setting. The evaluation comprised of a number of components including a test, evaluation task and questionnaire. The evaluation was completed by five users from diverse backgrounds. Once all five users had completed the usability evaluation the data was collected from the questionnaires, tests, recorded questions/comments and video screen capture. The data was grouped so as to address the respective issues, the results of which are detailed below.

It should be noted that the Likert scale used in the second usability evaluation incorporated a five point scale compared to a four point scale in the first usability evaluation. The scale was increased in the second usability evaluation to allow a greater range to be captured. The associated questionnaire was designed so five was the most favourable result and one the least favourable result.

During the course of the experiment any questions or comments asserted by the users were recorded. These were used in conjunction with comments left in the evaluation questionnaire to collect qualitative data which was used to assist in explaining system frailties or failings. While a vast majority of comments made during the evaluation were reaffirmation questions there was some pertinent data collected.

Dialogue Representation

Only the effectiveness of the Dialogue Representation was examined in this experiment as other aspects of its usability were addressed in the previous usability evaluation. Two methods of analysis were incorporated into the evaluation; questionnaire questions and test questions.

During the training session only one question was asked by a user that was relevant to the Dialogue Representation effectiveness. While completing the Proficiency Task a user asked if there were two different types of nodes, one for Statements and one for Responses. While this is a substantial error the user had simply not used ACTSim in a number of months; having successfully developed a Dialogue Model previously.

Figure 6.21 presents the mean score of the users' associated test. It shows the objective results of the users understanding with respect to individual areas which are related to the effectiveness of Dialogue Representation. The only aspect of Dialogue Representation not to score over 80% was related to how well users understood the translation of the graph to the soft skill simulation produced; the fourth bar in Figure 6.21. This is conceptually the most difficult and challenging correlation for users to visualise. It was for this reason that the Script feature had been previously added to ACTSim; to aid the author in following the dialogue they have created and see how it might play out in the simulation.

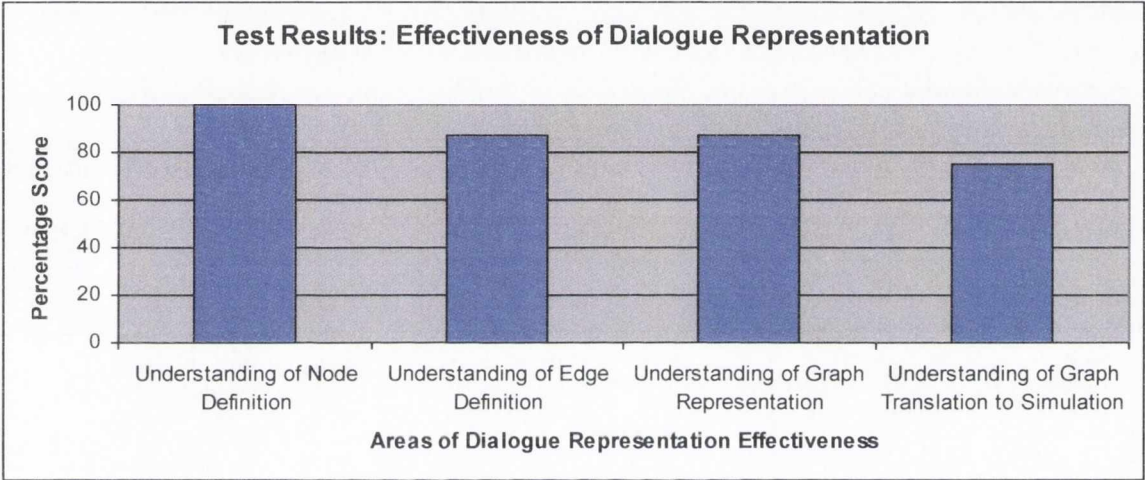


Figure 6.21: Dialogue Representation Effectiveness Test Results

Figure 6.22 presents the questionnaire results related to the Dialogue Representation effectiveness. All areas related to Dialogue Representation effectiveness returned an average between four and five in a Likert scale of one to five where five is the most favourable result. While these results were very positive one user did comment in the questionnaire that they felt the graph approach may not have entirely suited their dialogue. This was most probably due to the nature of the users' Dialogue Model, which they also acknowledged. The user had composed a Dialogue Model that was highly informative that used a very low level of interaction. The resultant Dialogue Model did conform with what can be thought of as '*natural dialogue*' and Dialogue Elements became heavily interlinked and closely compounded.

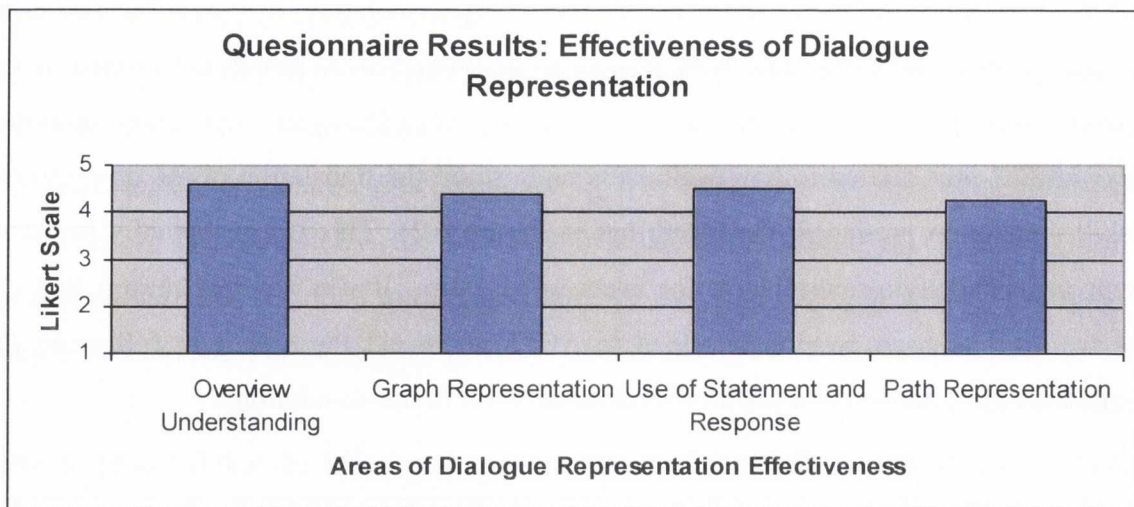


Figure 6.22: Dialogue Representation Effectiveness Questionnaire Results

Adaptivity Application and Representation

Adaptivity Application and Representation was examined with respect to its effectiveness, efficiency and usability. Due to the importance of adaptivity design within ACTSim several methods of evaluation were employed to gather data. These included test questions, video screen capture analysis and questionnaire questions.

Figure 6.23 presents the mean test results related to the adaptivity effectiveness. Figure 6.24 displays the associated questionnaire results which uses a Likert scale similar to the one previously described in the Dialogue Representation. Most of the test results are positive with users scoring an average between 80% and 100%. The first result of note not to score highly relates to Understanding the Result and Effect of Adaptivity within the Simulation. The other areas of Tagging, the first two bars in the graph, deal with the concept of Tagging within the composition tool. The users can conceptualise how inclusion and exclusion will work when dealing with their concrete model; this is confirmed in the questionnaire results graph, Figure 6.24, the first bar indicating the users understood the effects of Tagging. It seems however that users have a difficulty when they attempt to relate how their adapted model will be realised as an end personalised simulation. One user commented in the questionnaire that the “*effect of tagging not obvious until the end*”. This is a similar issue, and possibly an extension of, the Understanding of Graph Translation to the Simulation area described previously in Dialogue Representation analysis. Similarly to the Script function designed to aid authors’ visualisation of dialogue, the Highlight function developed for ACTSim was designed so authors could visualise Tagging and see how it might affect their Dialogue Model.

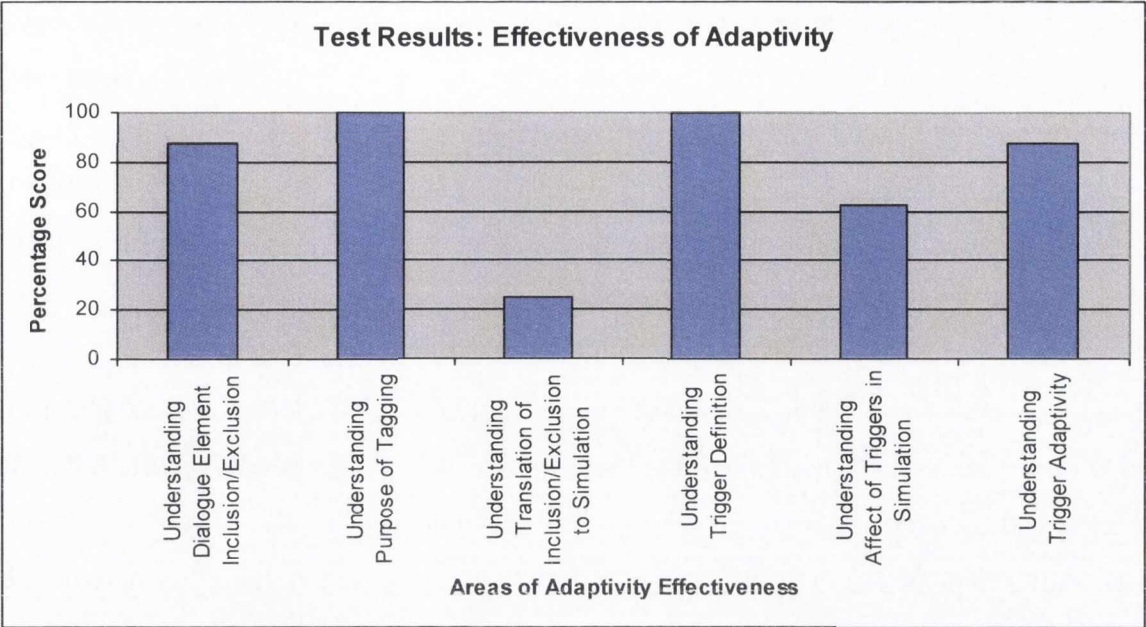


Figure 6.23: Adaptivity Effectiveness Test Results

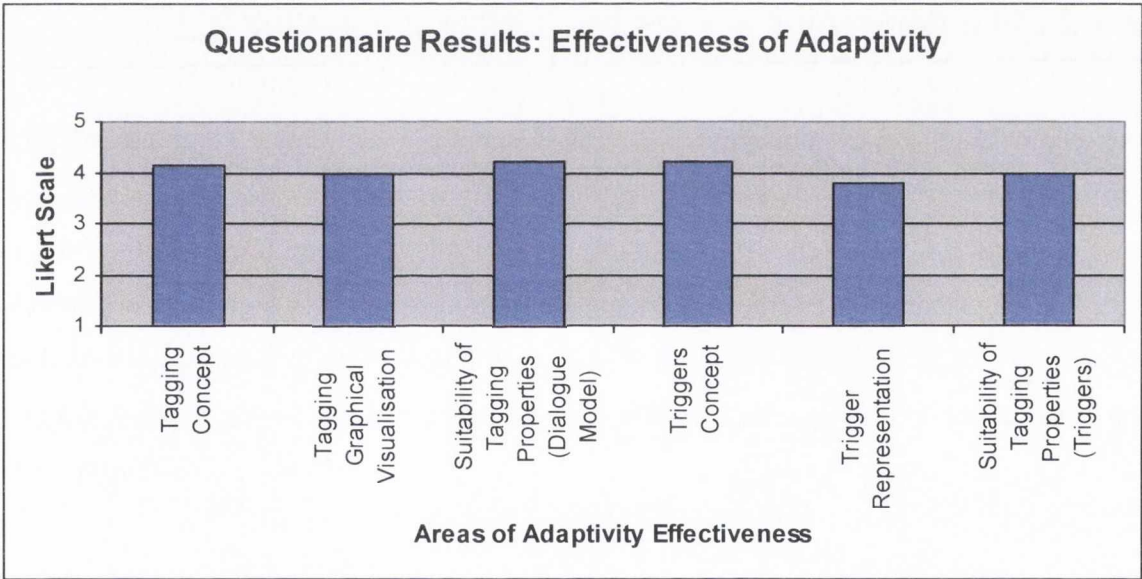


Figure 6.24: Adaptivity Effectiveness Questionnaire Results

The effectiveness of Triggers within the composition tool generally scored very well. There were however two areas in the evaluation where Triggers did not score as well as anticipated. The first of these, Understanding the Affect of Triggers in the Simulation, is highlighted in the effectiveness test results; the fifth bar in Figure 6.23. The test question that this result is associated with asked “*Briefly (one or two sentences) explain how a trigger in the dialogue model will effect the final simulation*”. A correct answer required the users to give a general

description of what happens when a learner encounters a Trigger within a simulation; a Trigger firing if its tag matched that of a learner's user model. While some users gave a correct answer there were generally only partially correct answers given by most users. Answers were either specific to a particular type of Trigger or did not mention that Triggers will not fire if they are not suited to an individual learner. It is the author's view that the evaluation users did not fully understand the question and what was required as a fully correct answer.

The second area where Triggers effectiveness did not score as expected was related to the Trigger Representation; highlighted by the fifth bar in the graph of Figure 6.24. The statement that users were asked to consider was "*I like the way triggers were represented in the Dialogue Model*". Four of the five user indicated that they agreed with this statement but one user neither agreed nor disagreed, meaning the result for the Trigger Representation was just below the 4 on the Likert scale.

In order to evaluate the efficiency of functions associated with adaptivity an analysis of the video screen capture was conducted. The time taken for each user to complete adaptivity operations was recorded along with the number and types of errors that were made. Errors were divided into three categories; slip, lapse and mistake (Reason 1990). Both a slip and lapse are unintended actions which are acceptable in low numbers. In high numbers however, these types of errors can affect the efficiency of an application. A mistake is associated with an intended action and can indicate a more serious usability issue with a system; details of errors classification can be found in Appendix XIII - Second Usability Evaluation - Error Classification.

Figure 6.25 displays a table which summarises the mean time expended by each user to complete the associated adaptivity operation. The table also includes an overview of the number and type of errors made by the users while completing these operations. The majority of operations are performed within an acceptable amount of time. For example, Tag properties were, on average, created in 15.76 seconds (well below the 30-45 seconds acceptable limit) and all Triggers were created and updated in less than a minute. The two areas that do not meet expectations were the 'Highlight Tags' operation and the 'Delete Triggers from Compartment' action. The reason the 'Highlight Tags' action took so long may

be due to the fact that this was a relatively new function that most users would not have previously encountered. While no errors occurred when users attempted to use the function the associated user interface and selection was deliberated over for some time. The task required users to highlight Tags initially using a single Tag, followed by a combination of Tags using the AND operation and finally a combination of Tags using the OR operation. The list of Tags used in this part of the task was also unfamiliar to the users and they spent considerable time simply locating the tag instances in the lists provided. Users may also have needed more time when considering the Boolean operation.

The other area of concern was deleting a Trigger from a Dialogue Element Compartment. This is a relatively simple and quick procedure to execute; select the Trigger in the Dialogue Model and press the Delete or Backspace button. Several users attempted to delete the Trigger through a popup menu accessed by right clicking the Trigger. The function that the users were attempting to use is a GMF feature which allows operations to be completed on the background data model and was not designed to be used to delete graphical objects. Users made several attempts to complete the operation using this function before completing the operation correctly. This system issue is further highlighted by the number and type of errors made by users, as seen in the table of Figure 6.25.

In terms of efficiency however the total number of errors committed by users is very low. Although familiar with the application, having previously composed Dialogue Models and completed the proficiency test at the beginning of the experiment, some users may not have used the composition tool in a number of months or may not have used some of the new features. Even without this as a consideration the number of errors that occurred during the evaluation was acceptable.

Operation Related to Adaptivity	Average Time (seconds)	Slip	Lapse	Mistake	Total Errors
Creating Tag Properties	15.76	1	1		2
Tag Individual Dialogue Element	5.75				
Tag Groups of Dialogue Elements	14.7	2	1		3
Highlight Tags	19.9				
Remove Tag(s): Single Dialogue Element	11.25		1		1
Remove Tag(s): Multiple Dialogue Elements	12.3				
Add and Update Assessment Trigger	34.6	1			1
Add and Update Feedback Trigger	28.2	1			1
Add and Update Reflection Trigger	39.2				
Move Triggers in the Dialogue Space	5		2		1
Delete Triggers from Compartment	13.8	1		2	3
Access Triggers Properties	<1	1	1		2

Figure 6.25: Mean Time and Error Count for Operations Related to Adaptivity

The final area to be examined with respect to Adaptivity Application and Representation was user satisfaction (functionality user friendliness); the evaluations post-questionnaire was used to gather this data. A summary of user satisfaction related to adaptivity can be found in Figure 6.26; the x-axis of the graph displays the different areas of user satisfaction while the y-axis displays a Likert scale from one to five where five is the most favourable result. As evident, users found all areas of functionality related to adaptivity to be user friendly; all areas score between four and five in the Likert scale. Even the ‘Delete Triggers’ area, which caused users to make several errors (as shown previously), achieves an adequate level of user satisfaction. This may be because that, although the system was at fault for allowing a confusing function to be available, users were still satisfied with the correct functions operation eventually executed.

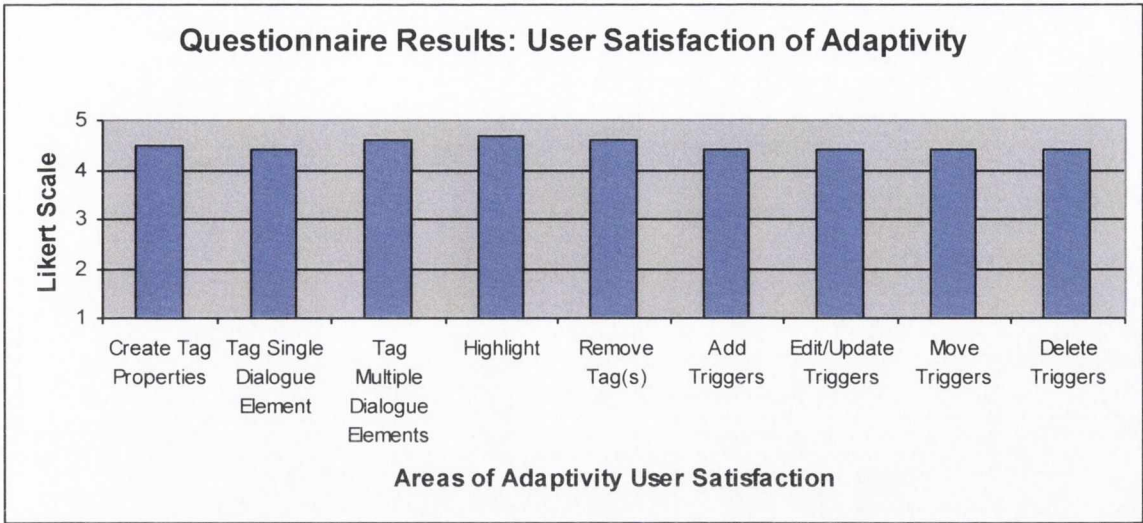


Figure 6.26: Adaptivity User Satisfaction Questionnaire Results

Authoring Process

ACTSim Authoring Process was examined with respect to its effectiveness, efficiency and usability. These areas were examined using the usability evaluations post-questionnaire. Figure 6.27 presents an overview of each area being examined; unlike previous data correlations which displayed each sub area being analysed an overview of the authoring process sufficiently describes the data. Effectiveness and efficiency score slightly below the expected value (3.96 and 3.9 respectively), while user satisfaction has scored between four and five on the Likert scale.

The results of the Authoring Process effectiveness may have been affected by the stages not being explained adequately; one user commented that they would have liked ‘*more explanation of each stage*’. Furthermore, another user commented that they “*found the triangle within the circle slightly confusing*”. The ‘*triangle*’ simply demonstrated constructive alignment. This was used during its development and is of no benefit to the users so should have been removed from the final Authoring Process. The Authoring Process effectiveness, however, did score relatively high despite these slight faults and one user commented in the questionnaire that it was “*easy to go through*”.

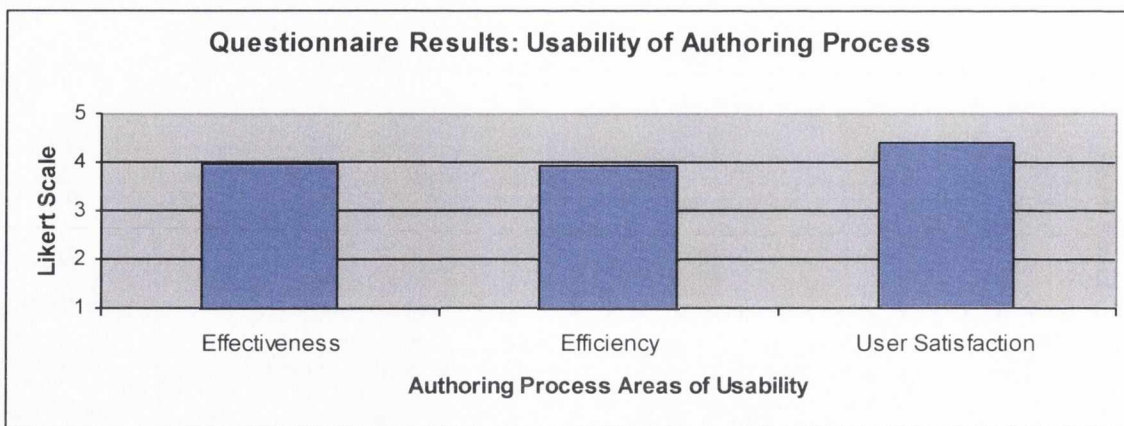


Figure 6.27: Authoring Process Usability Questionnaire Results

Efficiency seems to have been affected by the necessity of having to complete several iterations of the Authoring Process; one user commented that a single iteration “*would still take me a while*”. While these can be time consuming, iterations are necessary to effectively complete a Dialogue Model. Users were also required to attempt completion of all steps in a single iteration of the Authoring Process during the evaluation; this would not typically be necessary.

Pedagogical Framework

The Pedagogical Framework developed for ACTSim was examined with respect to its effectiveness, efficiency and user satisfaction. The data from both the Pedagogical Approach and Learning Outcomes were correlated and are displayed in Figure 6.28. As shown, users found the Pedagogical Framework to be accessible, quick and easy to use. One user did require further explanation of the Pedagogical Approach during training. They had composed their Dialogue Models very early in ACTSims development and had not previously explicitly described their Pedagogical Approach within the tool.

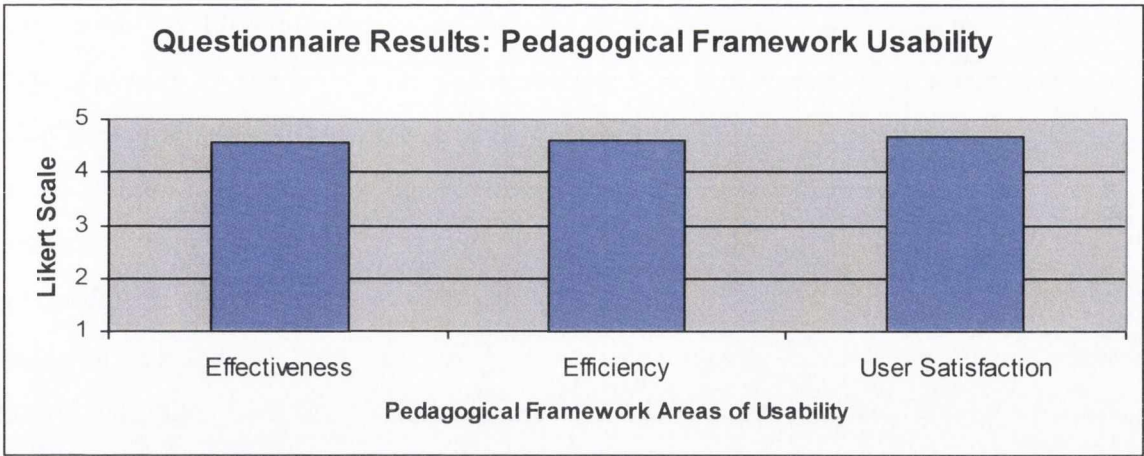


Figure 6.28: Pedagogical Framework Usability Questionnaire Results

Model Validation

The Model Validation features developed for ACTSim (Highlight, Verification and Script) were examined with respect to their effectiveness, efficiency and user satisfaction. The data from all three of these features were correlated and are displayed in Appendix XIV – Evaluation Results - Model Validation. Users found the Pedagogical Framework to be accessible, quick and easy to use.

Navigational Aids

All the Navigational Aids, with the exception of the Map feature, were examined with respect to effectiveness, efficiency and user satisfaction. Figure 6.29 shows a correlation of each Navigational Aids performance measured using the evaluations questionnaire. The graphs display the usability areas in the x-axis and the y-axis employs a Likert scale of one to five, where five is the most optimal result. The Zoom and Search features both perform very well and each area scores between four and five on the Likert scale. The efficiency of the List feature, which is circled in the bottom left graph of Figure 6.29, only achieves 3.6 in the Likert scale. Users found with a very large Dialogue Model there would be an extensive list for them to explore. One user commented in the questionnaire that ‘*you still have to look at all of the [Statements] to find the one you want*’. The List feature only seems efficient when the number of Dialogue Elements falls between a particular range; too few Dialogue Elements and it is easier to search the Dialogue Space, too many Dialogue Elements and the List becomes inefficient and it would be best to use another Navigational Aid such as the Search feature.

The only other area of the Navigational Aids not to score between four and five on the Likert scale was the effectiveness of the Arrange feature; circled in the bottom right graph of Figure 6.29. The results regarding the effectiveness were very mixed due to the situations in which it had been used. Users found that when they executed Arrange with their own models (which would have occurred while initially developing their models before the usability evaluation) the function rearranged the models in such a way that they became difficult to follow; one user commented *“I was used to knowing the layout of the [Dialogue Elements] so would not like them rearranged”*. However, the evaluation task required the users to apply the Arrange feature to an unfamiliar Dialogue Model; this produced more favourable results within the questionnaire. One user commented on the distinction of the two situations where the Arrange feature could be applied by saying *“not necessary for my model but worked well for demo model”*. These comments highlight the situation in which the Arrange feature is really effective. The evaluation questionnaire needed to be clearer in separating the two circumstances which was left up to the users’ interpretation.

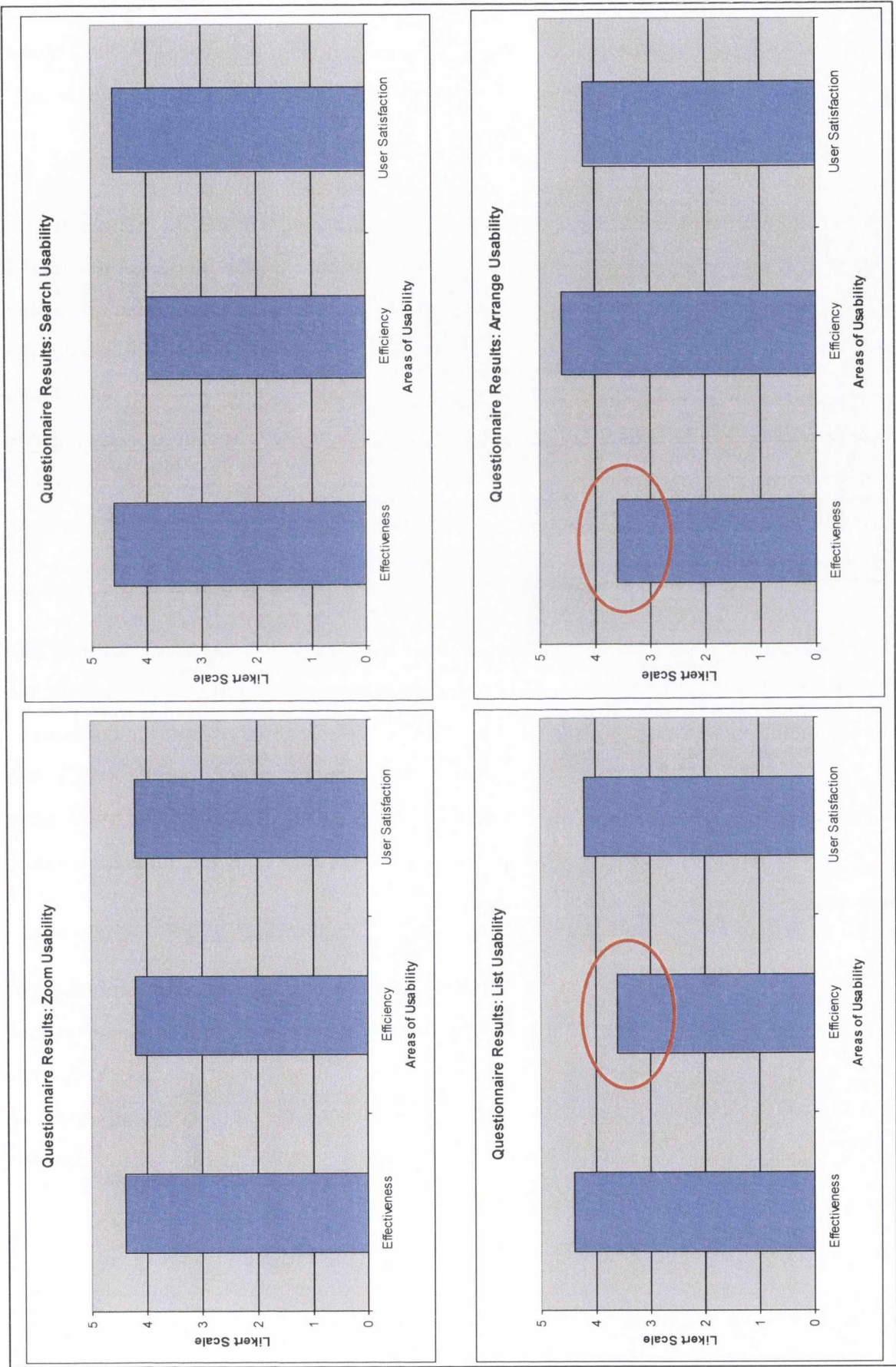


Figure 6.29 Navigational Aids Usability Data

6.4.4 End of Phase Two Reflections

The results of the second usability evaluation were generally very positive. The evaluation comprehensively examined the representation and application of adaptivity in the second usability evaluation. Furthermore, additional functionality and some design aspects that were not conclusively examined in the first usability evaluation were also evaluated.

The second usability evaluation suggests that the approach to adaptivity in ACTSim employs techniques that are accessible to the authors. Authors had little trouble tagging their model appropriately and adding Adaptive Triggers. The approaches incorporated into the composition tool resulted in models that did not become significantly more complicated; the separation of adaptivity from the simulation model allows adaptivity to be easily edited and altered. Furthermore, the adaptive features, while also being extensible, were easily understood by the authors and addressed different aspects of the system: Role (learner); Subjects (simulation model); Categories (communication/interaction types); and Learning Outcomes (pedagogical goals).

The only criticism of adaptivity application and representation within in the ACTSim is that there perhaps still lacked a full comprehension of how the adaptivity is realised in the delivered simulation on the part of the authors. As it is possible to have multiple dimensions being applied to the model, the resulting personalisation can be rather complex. While the highlight functionality attempts to assist the author with visualising resultant adaptivity in the model, it is a visual aid rather than a comprehensive resultant view of the simulation; some additional functionality could assist in this area.

The examination of the authoring process showed it to be effective and user friendly but possibly a little inefficient. As authoring adaptive systems and soft skill simulations is a rather complex task, there are many steps incorporated into the authoring process. An author, at least when first using the composition tool, may find they need to be constantly viewing the process as they progress. A more interactive integration of the authoring process into the composition tool may improve its efficiency.

The model validation functionality was found to be effective, efficient and easy to use. However, it is the author's opinion that this functionality could be more sophisticated. While

examining Dialogue Elements for the existence of Statement and Responses is important it is rather simplistic. An improved approach might examine the structure of the model to search for root nodes, end points or loops. Despite its shortcomings the validation functionality does demonstrate validation as a proof of concept which could be improved upon in the future.

The results of other areas examined by the second usability evaluation were also positive. As expected the more comprehensive examination of the Dialogue Representation was very good. Similarly, most of the navigational aids performed very well. An examination of the pedagogical framework employed was also successful in its design and implementation.

The final aspect of the reflections at the end of both the first phase and the second phase of development was to compare ACTSim with related work. The following section presents a brief comparison of ACTSim with systems described in the state of the art survey.

6.5 Comparison of ACTSim to Related Research

This section briefly compares and contrasts ACTSim with the composition tools described in the state of the art survey contained in chapter 3; the review is presented in relation to ACTSims requirements.

Dialogue Representation

There are many authoring tools that employ a simple text interface to describe the simulation model such as Experience Builder and Knowledge Dynamics KDSimStudio (KDSimStudio 2011). While this is adequate for describing simple simulations it does not allow the author to develop large authentic simulations and is insufficient to capture the complexities involved in describing the various paths available to the learner. The graph based approach to representation is far more intuitive and user friendly than the use of text to describe the simulation models.

The manner in which the ACTSim composition tool represents dialogue and allows the author to construct the dialogue models is similar to the ACCT's approach to representing learning content. The use of nodes and edges to represent dialogue and speech can also be found in many dialogue management systems (Churcher, Atwell et al. 1997) such as the CSLU toolkit (Sutton, Cole et al. 1998) and GULAN (Gustafson, Elmberg et al. 1998). However, while these systems incorporate an intuitive representation of dialogue they are not

used for composing adaptive educational simulations. They do not incorporate the same functionality as ACTSim such as determining a pedagogical framework, applying adaptivity or include an authoring process to support the author in creating the simulation and Dialogue Model.

Adaptivity: Application

Unlike the ACCT authoring tool, ACTSim allows the author complete control over the personalisation that can take place within the simulation. The MOT authoring tool allows the author such control but requires the author to have a technical background in adaptive hypermedia to incorporate the adaptivity. The ACTSim composition tool incorporates the best of both the ACCT and MOT; the author completely controls the adaptivity but does not need to create complex rules to define the adaptivity.

Adaptivity: Trigger Representation

The approach of incorporating additional components in a simulation model is not unusual. For example, SimWriter incorporates many different types of nodes within its model representation. These non-adaptive nodes not only represent the dialogue that occurs within the simulation but also indicate the occurrence of such things as learner reports and directed feedback. SimWriter and other soft skill simulation authoring tools link these components directly with the components that are used to describe the dialogue. This approach immediately adds to the complexity of authoring the simulation and also makes the flow of the dialogue difficult to follow. The approach employed by ACTSim allows Trigger components to be placed in Dialogue Element compartments. Such separation from the Dialogue Model allows Triggers to be easily moved or deleted from a model.

Authoring Process

The Dialogue Model development process established in this research is based on a concept employed by Experience Builder. The approach Experience Builder incorporates initially requires an author to develop the most optimal and direct route through a dialogue that a learner might navigate. Once this has been completed an author adds additional options and routes to the optimal route thus expanding the simulation. There are several advantages to incorporating such an approach:

- This approach captures structuring principles which are conducive to creating dialogue models. It initially instigates a directed theme that runs through a dialogue from which additional aspects can be established.
- Experience Builder's approach is easily accessible to authors who may not be experienced with composing eLearning courses or designing interactive dialogues.
- The execution of this approach is simple and straight forward. Training an author in using this technique requires little instruction.
- One of the main challenges for an author in composing and developing an ACTSim Dialogue Model is knowing where to begin. This approach creates a clear and concise starting point from which an author can expand and develop their ideas branching from a central theme.

While the concept employed by Experience Builder to develop a soft skill simulation's dialogue model is effective its implementation is poorly executed. Its limitation is due to the restrictive representation of dialogue within the tool. The idea of creating an optimal route and expanding it is good but as an author cannot view their model graphically the process and model become difficult to follow; this is particularly true with large scale simulation development.

The ACTSim Dialogue Model development process is based on principles employed by Experience Builder but it is incorporated into the process which also creates a graph representation of the dialogue model. The Dialogue Model development process also becomes a much more explicitly iterative step in this research, while this methodology is implied by the Experience Builder process it is not clearly defined.

Pedagogical Framework

Generally soft skill simulation authoring tools do not incorporate any kind of pedagogical framework; although when done so it is typically implemented in a simplistic accessible manner. This is contrary to composition tools designed for authoring more traditional eLearning content such as CourseLab (CourseLab 2012) and DialogPLUS (Davis and Fill

2007; Bailey, Fill, et al. 2006). They are more likely to employ such a pedagogical framework but the implementation is typically more technical and less accessible.

Experience Builder, SimWriter and the ACCT allow authors to simply define Learning Outcomes. ACTSim employs a similar strategy but also permits authors to easily associate Learning Outcomes with Dialogue Elements. This approach allows authors to ensure that all Learning Outcomes are addressed within the simulation model which assists in the simulation design. While MOT employs elements of a pedagogical framework the goals and constraints model that authors are allowed to compose is a rather technical approach and may be somewhat inaccessible to non-technical subject matter experts.

The ACCT also supports pedagogical and instructional theories; although these are expressed implicitly by the author. ACTSim refines this concept as the Pedagogical Approach to allow authors to explicitly state the simulations instructional strategy. This is particularly suited to soft skill simulations as an author must decide the direction of the dialogue upon which the Dialogue Model is based; the Pedagogical Approach allows an author to define the style of the dialogue.

Navigational Aids

The approach employed by most composition tools to express and represent dialogue or content means they do not require navigational aids. Typically, traditional eLearning models, such as those created in the ACCT, are not as large and expansive as soft skill simulation models due to the fine granularity needed to express dialogue. While SimWriter employs a graphical representation of dialogue, it does so in a restrictive manner so as not realistically allow large complex models making navigational aids unnecessary. Although Captivate 5 is similarly restrictive it does employ an interactive map. ACTSim incorporates a similar map navigational aid; such a feature would benefit the VISION composition tool. Other navigational aids incorporated into ACTSim are typical of eLearning authoring tools, the zoom and search features for example are common.

Model Verification

The only authoring tool surveyed to incorporate any kind of model verification was SimWriter; specifically a scripting feature. ACTSim incorporates a similar feature but also

employs a highlight feature to allow adaptivity tags to be viewed and a validation feature which allows an author to verify that essential data has been entered.

6.6 Summary

This chapter has described the numerous evaluations that have taken place during the development of ACTSim. Specifically three different approaches were used to fully evaluate the authoring tool at different stages of development; predictive, interpretive and usability evaluations. Predictive evaluations allowed evaluations to be completed before a prototype was developed allowing designs to be quickly and easily assessed. Interpretive evaluations allowed real authentic end users to effectively and efficiently influence the design of ACTSim. The usability evaluations insured that each aspect of the authoring tools design was rigorously re-examined in a completely subjective manner.

The final usability evaluations results were very positive. There were however a few small issues highlighted by this experiment. For example; the authoring process effectiveness and efficiency could be improved; the Arrange feature was not as effective as hoped; and the Trigger Representation could be improved. How these minor issues might be addressed, along with other possible design innovations, are discussed in the following final Conclusions chapter.

7 Conclusions

This thesis has presented research in the area of authoring adaptive soft skill simulations. It has detailed innovative approaches, features and processes incorporated in a composition tool designed to be used by non-technical subject matter experts. ACTSim is a unique authoring tool that not only allows soft skills to be composed rapidly without the need to write code but also allows adaptivity to be implemented effectively and efficiently. The system developed as part of this research was extensively and rigorously evaluated with genuine real world users using several different evaluation approaches during its development.

This chapter initially discusses the objectives and achievements of this research. It also identifies the key contributions the research has made to the state of the art in both the areas of authoring Adaptive Educational Hypermedia systems and educational simulations. The final section of this chapter discusses possible future work that could take place which would advance and improve the research already completed.

7.1 Achievements

The research question first posed by this thesis in chapter one was:

“What are models, mechanisms and authoring processes required for adaptive soft skill simulations and how can adaptive features for soft skill simulations be authored by non-technical designers?”

To fully analyse this research question there are three areas that must be examined; the models, mechanisms and authoring processes. Each of these areas are discussed in the following sections.

What are the Models? Within Adaptive Hypermedia there are typically several models employed; for example, the user model, concept model and adaptivity model. The key model within soft skill simulations is the Dialogue Model. The objective of the Dialogue Model (content model) developed by this research was that it be accessible to non-technical subject matter experts while also being robust and descriptive enough to define dialogue that would occur within a simulation. The use of a graph design (nodes and edges) achieves this objective as it is visually descriptive yet easily understood. While this approach is used by

many soft skill simulation authoring tools and dialogue management systems, the design used in this research is far more user friendly, effective and efficient compared to other systems. It allows users to freely place Dialogue Elements with the Dialogue Space and it does not restrict an author's composition. Furthermore, the design does not incorporate many different types of node and thus reduces complexity and prevents confusion. The few additional types of nodes that do exist cannot be attached directly to the Dialogue Model allowing them to be easily moved and deleted. Dialogue Models in ACTSim are easy to compose, read and comprehend.

What are the Mechanisms? The objective of this research was to allow authors to initially develop soft skill simulation models (described above) which can be formatted in such a way as to become adaptive. There are two approaches designed to allow users to easily and effectively apply adaptivity; Tagging and Triggers. Both mechanisms have been designed to be accessible by non-technical subject matter experts. Implementation is accomplished graphically and resultant adaptivity can be very effective. The Tagging mechanism is applied easily through multiple selection or editing an individual Dialogue Elements tagging component. The system also allows Tagging properties to be easily emphasised visually through a highlight feature which can be manipulated by authors so as to see different combinations of tag properties across their model.

The Triggers mechanism developed as part of this research is also easily adopted by authors and can produce very effective personalised pedagogical events. Triggers are easily created, moved and edited within the Dialogue Model. Tagging them for personalisation is easily accomplished and approach to their adaptivity is simply an extension of the inclusion and exclusion of Dialogue Elements and so easily understood by authors. Trigger types (Assessment, Feedback and Reflection) are also all very approachable for non-technical users and even without any formal pedagogical training authors would find these educational concepts easily understood.

What are the Authoring Processes? The objective of this research was to develop a rigorous authoring process that was easily accessible to non-technical authors which incorporated good practices in the development of soft skill simulations and personalisation.

The authoring process also needed to be pedagogically informed, thus insuring the developed simulations were educationally effective.

The developed authoring process is iterative in nature so as to ensure that authors would allow assessment and reflection of their work to feed into further development of their Dialogue Model. This approach ensured rigor in the authors approach to development.

The authoring process was also ensured to be pedagogically informed by designing it about constructive alignment. Authors are required to develop learning outcomes and a pedagogical approach as part of the authoring process. Both of these features within the authoring tool were designed to be accessible and user friendly.

All previously mentioned steps within the authoring process were designed to give authors adequate support in creating a Dialogue Model, the development of which is realised itself as an iterative step within the authoring process. The author is again supported within this step as they are instructed and direct how to go about creating their models with a Dialogue Model authoring process. The main stem branching approach is very effective in allowing authors to compose Dialogue Models quickly and easily. The final aspect of the authoring process is to allow authors to apply adaptivity by Tagging their models and adding Triggers.

7.2 Objectives

Chapter one of this thesis outlined five objectives, as presented in section 1.3. This section presents a brief analysis how successfully each of these objectives has been addressed.

1. *A state of the art survey of authoring tools used to compose adaptive soft skill simulations.* As previously described, to the authors knowledge no other such authoring tools currently exist. To successfully address this objective the state of the art analysis was divided into two. The first part of the analysis examined tools used to compose soft skill simulations, the second part of the analysis focused on educational AH composition tools. The state of the art survey, which can be found in chapter three of this thesis, was initially completed at the beginning of this research and was reviewed and updated regularly during the course of this research. The analysis allowed detailed system requirements for the adaptive models, authoring process and

authoring systems to be derived. These key findings were later realised in the implementation and design of the ACTSim composition tool.

2. *Refine and design the models required for composing adaptive soft skill simulations.*

The goal of this objective was to employ approaches to adaptivity which were both effective and easily accessible to non-technical subject matter experts within the domain of soft skill simulations. The objective was successfully addressed by combining techniques from Adaptive Hypermedia and Soft Skill Simulation research with new and unique interpretations and mechanisms which could be easily understood by subject matter experts. Specifically, a graph representation of dialogue (Dialogue Model) which allowed adaptive flow to be easily defined, adaptive mechanisms (Triggers) which allowed educational interventions to be personalised and adaptive dimensions which were easily accessible and related to soft skill simulations. A detailed description of these designs was described in chapter four of this thesis.

3. *Research and develop an authoring process for composing adaptive soft skill simulations.* This research successfully addressed this objective by developing an authoring process which incorporates constructive alignment and components related to adaptivity application; as described in section 4.2.4 of this thesis. The authoring process is iterative to insure rigorous composition and is designed to be accessible to non-technical subject matter experts.

The process can be accessed visually within the ACTSim composition tool but is not embedded in the software. A loose integration with the process and software was chosen as not every step of the process will be completed with each iteration. A user will instead customise the process during each iteration to include the steps that they require. The loose integration also means that the process, although designed for applying adaptivity to soft skill simulation models, is realised as a somewhat separate entity and could be applied to other authoring situations or composition tools.

4. *Research and develop a tool to enable non-technical authors to create the relevant models.* The ACTSim composition tool was designed and implemented to support the development of Dialogue Models which could be tagged for adaptive flow and which

would allow Adaptive Triggers to be incorporated. Furthermore, the tool supported pedagogical components which support for the authoring of the educationally sound soft skill simulations. The ACTSim system also employed many features which assisted the subject matter experts in navigating and validating the models that they composed.

The ACTSim system was built using the Eclipses Graphical Modelling Framework plug-in. While this technology was cutting edge at the time of development it can be identified today as having some limitations; specifically, it is not a web based graphical editor technology. New technologies, such as HTML5 and Adobe's Flex (not available when the ACTSim composition tool was being built), allow graph editors to be developed for access through web browsers. If ACTSim was to be redeveloped it might be done so as a web based system to allow easier access, software updates and group collaboration. Although it should also be noted that even with the technological advancements that have been made, a web based system may not support all the functionality presently implemented in the ACTSim system.

5. *Validate and evaluate the adaptive models, the authoring process and the simulation composition system developed.* The ACTSim system and its associated processes, mechanisms and features have been rigorously evaluated throughout the course of this research. Several different types of evaluations have been employed and evaluations have been completed as part of user based trials with real world users in several domains. In particular, this research has evaluated the approaches for applying adaptivity to Dialogue Models and Triggers with relevant Adaptive Dimensions.

The evaluations that were conducted were very in-depth, however, the total number of users could be considered relatively low; a total of eleven users were employed during the course of this research. The results of the evaluations are therefore probably more indicative than statistically significant. Future research should perform similar experiments but with larger populations. Furthermore, a large majority of the subjects employed in the experiments were highly qualified and educated to a postgraduate level. Future experiments might broaden the user pool and employ subject experts that are not necessarily experts in the area of soft skills but who *do* have an interest in the area.

7.3 Contribution to the State of the Art

The primary contribution to the state of the art of this research is an authoring tool that allows non-technical subject matter experts to apply and implement adaptivity in eLearning simulations that they have developed. This was accomplished by incorporating unique and innovative approaches to describe and graphically represent adaptivity within simulation models while employing adaptive dimensions suited to soft skills.

There are two approaches to authoring adaptivity incorporated in the ACTSim composition tool: Adapting Dialogue Models and Adapting Triggers. The first of these approaches, **Adapting Dialogue Models**, is based on the inclusion and exclusion of Dialogue Elements within a simulation. While this concept is widely used in many Adaptive Hypermedia eLearning systems this research is the *first* to apply such adaptivity to soft skill simulations. The particular innovation to authoring Adaptive Hypermedia eLearning experiences is that personalisation can be implemented by subject matter experts without the need to create rules or program code. ACTSim authors are allowed complete control over how inclusion and exclusion will occur.

The second approach to authoring personalisation is based on **Adapting Triggers** within the simulation. Unlike a traditional eLearning course, soft skill simulations employ natural narratives which are reactive to learners' choices and decisions. Within soft skill simulations a learner triggers events by arriving at a particular point in a simulation; such components are also evident in procedural simulations and eLearning games. This research's innovation is to interpret events within the simulation as being adaptive; they may or may not occur within a simulation. The visualisation of such events within the authoring tool are graphically realised as separate nodes to the rest of the model which can be placed within Dialogue Element compartments. Trigger nodes can be tagged within the composition tool to indicate their adaptivity. To the author's knowledge there are no other soft skill simulations which incorporate truly adaptive Triggers. Furthermore, there are no soft skills simulation authoring tools which allow such adaptive Triggers to be implemented graphically or otherwise.

The final aspect of adaptivity within this research which contributes to the state of the art is the approach to **Adaptive Dimensions**. The four dimension⁶ employed to define adaptivity were selected as they characterise *key features* relating to soft skill simulation. The Adaptive Dimensions are easily understood for those without a detailed knowledge of adaptive hypermedia and so are accessible for non-technical subject matter experts. To the authors knowledge there are no other composition tool which define adaptive dimensions specifically for soft skill training environments.

The three elements, Adapting Dialogue Models, Adaptive Triggers and Adaptive Dimensions, are combined in the ACTSim authoring tool to allow a non-technical person to develop highly adaptive learning experiences. This contribution to the state of the art allows non-technical subject matter experts to deliver sophisticated personalisation. Such composition by non-technical authors has hitherto not been possible in the area of soft skill simulations.

The additional contribution to the state of the art is an innovative authoring process. The authoring process is pedagogically informed and incorporates a Dialogue Model development process specifically designed for developing graph based Dialogue Models. The authoring process is also the first to incorporate the concept of personalisation design and implementation into soft skill simulations. ACTSim also implements a number of functional features which also support the composition of adaptive soft skill simulations.

As previously stated in this thesis, the research described here has been recognised for its innovation and contribution to the state of the art by being published in several international conferences and an important special issue journal in the area of authoring adaptive hypermedia.

The research described in this thesis has also been recognised nationally and internationally by different means. In 2008 the author was awarded the Irish Software Association Student Medal for this research being “*the most innovative and commercially viable software product or service*”. Furthermore, the author’s work was also recognised in an international student competition and was short listed for presentation at the Hypertext 2008 conference. Finally,

⁶ Adaptive Dimensions specified for Soft Skill Simulations include Role, Learning Outcomes, Relevant Subjects, Interactivity Categories

the ACTSim composition tool and its associated processes are currently being used to develop adaptive soft skill simulations in a campus start-up company, Empower the User (ETU 2012). As part of Empower the User's continued research and development, it is a partner in the ImREAL project (ImREAL 2011) where ACTSim is also being used in developing "*immersive reflective experience based adaptive learning*" soft skill simulations.

7.4 Future Work

This section describes two areas of future work, the first subsection outlines ACTSim authoring tool improvements while the second subsection presents possible future areas of research.

7.4.1 Authoring Tool Improvements

There are a number of areas which could be addressed so as to improve the accessibility and effectiveness of the developed authoring system. There are however two areas in particular that the author has identified which could make a significant contribution to the state of the art; feature enhancements (related to adaptivity) and workflow integration. Both of these areas are discussed in the following paragraphs.

Feature Enhancements

The two feature enhancements described below are specifically related to the improvement of viewing and applying adaptivity within the composition tool.

Improved Visualisation of Tagging Effects: The highlight feature allows authors to view Tags and different combinations of Tags within a Dialogue Model. This could be expanded to allow an author to input possible learner user models and view the resultant adapted model. The system would temporarily remove all irrelevant Dialogue Elements and Triggers so the author could see exactly how a simulation would be adapted. This feature could be combined with the above feature so an author could also experience an entire adapted simulation within the composition tool.

Advance Settings: While the goal of this research has always been to develop an authoring tool for non-technical inexperienced users it is possible that some users will have previous experience in Adaptive hypermedia or eLearning composition. A possible improvement to

ACTSim may be a feature that allows additional more advanced users to apply advanced operations such as implementing adaptive rules or creating their own adaptive dimensions.

Workflow Integration

There are some soft skill areas that explicitly involve an underlying process. For example, a soft skill sales or customer service simulation typically follows a defined procedural workflow. While these processes are explicitly stated they are only implicitly integrated into a soft skill simulation; an author will create a simulation based on the process but is not allowed to express it in a composition tool. It is the author's view that integrating such a procedural workflow would be a beneficial addition to ACTSim.

There would be a number of advantages to integrating procedural soft skill components. Such logic would allow an author to create a much more sophisticated simulation. The integrated process would also add to the existing pedagogy scaffolding and allow for better authoring practices and documentation. Finally, the workflow could be enhanced so as to become adaptive thus creating even more effective and efficient learning environments.

7.4.2 Future Research

This research has investigated and examined several different domains including training simulations, eLearning authoring and personalisation. As such, there are a number of directions that could be explored in order to expand this research. For example, advancing eLearning authoring techniques, applying different approaches to simulation composition and improving user experience interfaces. This section briefly describes two areas that the author views as being exciting and offer a strong possibility of adding to the state of the art within the domain of authoring adaptive educational simulations.

Adaptivity

This research has described a new and innovative approach to authoring adaptive soft skill simulations. However, while the approach to personalisation that has been described is intuitive and accessible, it is somewhat limited. Even with the addition of a feature described in the previous section, allowing an SME (Subject Matter Expert) to add adaptive dimensions, there is still limited scope for the type of personalisation that can be defined. The

nature of the personalisation described in this research is somewhat limited by inclusion/exclusion of model components.

One possible approach to expanding and improving the personalisation of soft skill simulations is to allow additional services to control or implement the adaptivity. This research has assumed that an adaptive engine executes adaptivity using the defined soft skill simulation models and user models. An approach that might allow for more sophisticated personalisation could be implemented with additional adaptive services. Such services would be specialized and capable of implementing very specific types of personalisation but would allow for different approaches to that described in this thesis. Such services would package the complexities of the personalisation so as to remove it from the scope of a designer or SME while still delivering an improvement to the learner experience.

An example of such personalisation has been implemented using ACTSim within the ImREAL project (ImREAL 2011). A metacognition service delivers personalised metacognitive prompts to a learner at points in a simulation indicated by an SME via the ACTSim Reflection Trigger. An author does not need to compose the reflective prompts they simply add Reflection Triggers at points where reflections should occur. The service then defines and delivers the prompts at run-time based on more complex algorithms than inclusion and exclusion.

A benefit of such services is that they can be plugged into different eLearning environments as they are designed to be integrated with existing architectures. The challenge of such integration is that the SME or simulation designer will still need to be allowed to define, to some degree, how the adaptivity will be realised. Such adaptive services will not only need to integrate with delivery platforms but also with design and authoring tools.

New ACTSim Applications

The ACTSim authoring tool was designed specifically for creating adaptive soft skill simulations. However, there is a possibility of broadening its application and implementing different types of adaptive and non-adaptive educational courses as well as other online scenarios. The strongest synergy for such applications is the possibility of employing ACTSim to build and compose eLearning games. As described in chapter two of this thesis

there is a strong link between simulations and games. The similarity between the two becomes even stronger when specifically considering soft skill simulations. Both soft skill simulation and games imply a narrative or flow in their operating models. However, there are several challenges that would need to be addressed to allow ACTSim to be broadened into this area. For example, there are number of different dimensions that also need to be considered when designing a game. These dimensions would need to be incorporated into the model during the model development and accommodated in the functionality of the tool. Furthermore, the ACTSim authoring process would need to be updated to include game design and ensure the enjoyment of game be preserved.

The ACTSim platform could also be used to develop scenarios that are neither simulation nor game based. One strong possibility would be to use the ACTSim authoring tool to develop dynamic storytelling scenarios. This would involve paths through the simulation being selected for users so as to generate a new story each time they accesses a scenario. The story path selection could be implemented with different types of approaches. A very simple approach might be based on simple randomisation while the more interesting research could examine sophisticated algorithms. The selection algorithms could use some form of personalisation so as to expand that area of research into a different domain. The authoring platform would not need significant alteration or redesign. The research challenge would be how to implement the selection of the story routes. There are numerous research questions to be answered with such an application. What algorithms would be suitable and effective for users for their enjoyment or education? Would there be a need to define a new type of personalisation or interpretation of the tagging? Would it still be educationally effective to select incorrect paths for learners? What additional feedback would be required?

Other possible areas of new applications include multiple choice story selection (story paths chosen by the users) or employing the ACTSim tool as an aid for developing movie scripts. Two objectives of the ACTSim authoring tool were 1) it could be used in multiple domains and 2) be suitable for people of a wide variety of technical backgrounds. The effect of these realised goals is a tool that is easy to use and very flexible. As such, the author believes there many areas where the ACTSim authoring tool could be employed.

Bibliography

(Abdullah and Davis 2003) Abdullah, N.A., H. Davis (2003) *"Is Simple Sequencing Simple Adaptive Hypermedia?"* 14th Conference on Hypertext and Hypermedia, Nottingham, UK, 26-30 August. ACM. Available online at <http://eprints.soton.ac.uk/259681>. Last Checked: 22 September 2011.

(Adams, Reid et al. 2008) Adams, W. K. , S. Reid, R. LeMaster, S. B. McKagan, K. K. Perkins, M. Dubson, C. E. Wieman (2008) *"A Study of Educational Simulations Part I - Engagement and Learning."* Journal of Interactive Learning Research, Vol. 19 Issue 3, 551-577, October 2008. JILR.

(ADAPT 2007) ADAPT (2007) *"Adaptive Plug-in for Run-time Composition of Personalised eLearning and Adaptive Simulations."* Available online at <http://kdeg.cs.tcd.ie/adapt>. Last Checked: 22 September 2011.

(Aebersold and Tschannen 2012) Aebersold, M., D. Tschannen (2012) *"Engaging the Avatar: New Frontiers in Immersive Education: Chapter 11 Using Virtual Simulators in Second Life for Teaching and Learning in Nursing Education"* Edited by Randy Hinrichs. IAP.

(AICC 2010) AICC (2010) *"AICC Compliant."* Available online at http://www.aicc.org/joomla/dev/index.php?option=com_content&view=article&id=63&Itemid=21. Last Checked: 22 September 2011.

(Akpan 2001) Akpan, J. P. (2001) *"Issues Associated with Inserting Computer Simulations into Biology Instruction: A Review of the Literature."* Electronic Journal of Science Education, Vol. 5 Issue 3, 1-35. EJSE.

(Alden 1998) Alden, J. (1998) *"A Trainer's Guide to Web-based Instruction."* American Society for Training and Development. ASTD.

(Aldrich 2004) Aldrich, C. (2004) *"Simulations and the future of learning : an innovative (and perhaps revolutionary) approach to e-learning"* San Francisco: Pfeiffer.

(Aldrich 2005) Aldrich, C. (2005) *"Learning by Doing. A Comprehensive Guide to Simulations, Computer Games, and Pedagogy in e-Learning and Other Educational Experiences."* San Francisco: Pfeiffer / John Wiley & Sons Inc.

(Alessi 1988) Alessi, S. M. (1988) *"Fidelity in the design of instructional simulations."* Journal of Computer Based Instruction, Vol. 15 Issue 2, 40-47. Springer.

(Alessi and Trollip 2001) Alessi, S. M., S. R. Trollip (2001) *"Multimedia for learning : methods and development."* Chapter 7: Simulations, 213-268 . Allyn and Bacon.

(Anderson, Reder, et al 1996) Anderson, J. R., L. M. Reder, H. A. Simon (1996) *"Situated learning and education."* Educational Researcher, Vol 25 Issue 4, 5-11, May 1996. Educational Researcher.

(Anderson and Sosniak 1994) Anderson, L. W., L. A. Sosniak (1994) *"Bloom's Taxonomy: A forty-year retrospective."* National Society for the Study of Education, Yearbook of the National Society for the Study of Education, Vol. 93 Issue 2. University of Chicago Press.

(Apple 2011) Apple (2011) *"The Apple iPad."* Available online at <http://www.apple.com/ipad/>. Last Checked: 22 September 2011.

(Armitage and Bowerman 2005) Armitage, N., C. Bowerman (2005) *"The LOM Approach-A CALL for Concern?"* Computer Assisted Language Learning, Vol.18 Issue 1, 109-118. Taylor & Francis Group.

(Aroyo and Kommers 1999) Aroyo, L., P. Kommers (1999) *"Intelligent Agents for Educational Computer-Aided Systems."* Journal of Interactive Learning Research, Vol. 10 Issue 3-4, December 1999. Association for the Advancement of Computing in Education.

(Arthorne and Laffra 2004) Arthorne, J., C. Laffra (2004) *"Official Eclipse 3.0 FAQs."* June 2004. Addison-Wesley Professional.

(Ashcraft 2006) Ashcraft, H. M. (2006) *"Cognition."* 4th Edition. Upper Saddle River, Prentice-Hall.

(Atkins and Murphy 1983) Atkins, S., K. Murphy (1983) *"Reflective learning: key to learning from experience."* Journal of Humanistic Psychology, Vol. 23 Issue 2, 199-217, April 1983. Wiley Online Library

(Bailey, Fill, et al. 2006) Bailey, C., K. Fill, M. Zalfan, H. Davis, G. Conole (2006) *"Panning for gold: designing pedagogically-inspired learning nuggets."* Educational Technology & Society: Special Issue Learning Design, Vol. 9 Issue 1, 113-122. International Forum Educational Technology & Society.

(Bajraktarevic, Hall et al. 2003) Bajraktarevic, N., W. Hall, P. Fullick (2003) *"Incorporating learning styles in hypermedia environment: Empirical evaluation."* Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems, Adaptive Hypermedia 2003, Budapest, Hungary, 41-53, May 2003.

(Barab, Hay et al. 1998) Barab, S. A., K. E. Hay, T. M. Duffy (1998) *"Grounded Constructions and How Technology Can Help."* TechTrends, Vol. 43 Issue 2, 15-23. Springer.

(Belkada, Cristea et al. 2001) Belkada, S., A. Cristea, T. Okamoto (2001) *"Discovernet: Adaptive Simulation-Based Learning Environment And Support System for Designing Neural Networks."* Journal of Educational Technology Research. JET.

(Bergin and Fors 2003) Bergin, R., U. Fors (2003) *"Interactive simulated patient—an advanced tool for student-activated learning in medicine and healthcare."* Computers and Education, Vol 40 Issue 4, 361-376, May 2003. Elsevier Science Ltd.

(Blaiwes and Regan 1986) Blaiwes, A. S., J. J. Regan (1986) *"Training devices: Concepts and progress."* Military contributions to instructional technology. New York: Praeger.

(Bohl, Schellhase et al. 2002) Bohl, O., J. Schellhase, R. Sengler, U. Winand (2002) *"The Sharable Content Object Reference Model (SCORM) - A Critical Review."* International Conference on Computers in Education. Auckland, New Zealand, 950 - 951, December 2002. IEEE.

(Boyd and Fales 1983) Boyd, E. M., A. W. Fales (1983) *"Reflective learning key to learning from experience."* Journal of Humanistic Psychology, Vol. 23 Issue 2, 99-117. SAGE.

(Brehmer and Dorner 1993) Brehmer, B., D. Dorner (1993) *"Experiments with computer-simulated microworlds: Escaping both the narrow straights of the laboratory and the deep blue sea of the field study."* Computers in Human Behavior, Vol. 9 Issue 2-3, 171-184. Elsevier.

(Brown 2007) Brown, E. (2007) *"The use of learning styles in adaptive hypermedia."* PhD Thesis, University of Nottingham. Available online at <http://etheses.nottingham.ac.uk/577/1/ejb-thesis-for-ethesis-archive.pdf>. Last Checked: 22 September 2011

(Brusilovsky 1996) Brusilovsky, P. (1996) *"Methods and Techniques of Adaptive Hypermedia."* User Modeling and User-Adapted Interaction, Vol. 6 Issue 2-3, 87-129. Springer.

(Brusilovsky 1998) Brusilovsky, P. (1998) *"Adaptive Educational Systems on the World-Wide-Web: A Review of Available Technologies."* Workshop WWW-Based Tutoring, 4th International Conference on Intelligent Tutoring Systems, San Antonio, Texas, USA, August 1998. Springer.

- (Brusilovsky 2001) Brusilovsky, P. (2001) *"Adaptive educational hypermedia."* 10th International PEG Conference, Tampere, Finland, June 2001, 8-12. Kluwer Academic Publishers.
- (Brusilovsky 2001a) Brusilovsky, P. (2001) *"Adaptive hypermedia."* User Modelling and User Adapted Interaction, Vol. 11 Issue 1-2, 87-110. Kluwer Academic Publishers.
- (Brusilovsky 2003) Brusilovsky, P. (2003) *"Adaptive navigation support in educational hypermedia: the role of student knowledge level and the case for meta-adaptation."* British Journal of Educational Technology, Vol. 34 Issue 4, 487-497. Wiley.
- (Brusilovsky 2004) Brusilovsky, P. (2004) *"KnowledgeTree: A distributed architecture for adaptive e-learning."* The 13th International World Wide Web Conference (WWW), New York, USA, 104-113. ACM
- (Brusilovsky 2012) Brusilovsky, P. (2012) *"Adaptive Technologies for Training and Education: Chapter 3 Adaptive Hypermedia for Education and Training."* Edited by Paula J. Durlach, Alan M. Lesgold. Cambridge University Press.
- (Brusilovsky and De Bra 1999) Brusilovsky, P., P. De Bra (1999) 2nd Workshop on Adaptive Systems and User Modeling on the World Wide Web, 8th International World Wide Web Conference, Toronto, Canada, May 1999. Available online at <http://wwwis.win.tue.nl/asum99/>. Last Checked: 22 September 2011
- (Brusilovsky, Kobsa, et al. 1998) Brusilovsky, P., A. Kobsa, J. Vassileva (1998) Editors: Adaptive Hypertext and Hypermedia, Pittsburgh, USA, June 1998. Kluwer Academic Publishers.
- (Brusilovsky and Rizzo 2002) Brusilovsky, P., R. Rizzo (2002) *"Using maps and landmarks for navigation between closed and open corpus hyperspace in Web-based education."* The New Review of Hypermedia and Multimedia, Vol. 9 Issue 1, 59-82. Taylor & Francis, Inc.
- (Brusilovsky, Schwarz et al. 1996) Brusilovsky, P., E. Schwarz, G. Weber (1996) *"ELM-ART: An Intelligent Tutoring System on the World Wide Web."* 3rd International Conference on Intelligent Tutoring System, Montreal, Canada, 261-269. Springer.
- (Brusilovsky, Schwarz et al. 1996a) Brusilovsky, P., E. Schwarz, G. Weber (1996) *"A tool for developing adaptive electronic textbooks on WWW."* World Conference of the Web Society, California, USA, October 1996, 64-69. AACE

(Budinsky, Steinberg, et al. 2003) Budinsky, F., D. Steinberg, E. Merks, R. Ellersick, T. Grose (2003) *"Eclipse Modeling Framework."* Addison Wesley Professional.

(Calvi and De Bra 1997) Calvi, L., P. De Bra (1997) *"Proficiency Adapted Information Browsing and Filtering in Hypermedia Educational Systems."* User Modeling and UserAdapted Interaction, Vol. 7 Issue 4, 257–277. Springer.

(Calvi and Cristea 2002) Calvi, L., A. I. Cristea (2002) *"Towards Generic Adaptive Systems Analysis of a Case Study."* Adaptive Hypermedia and Adaptive Web-Based System, Vol. 2347, 79-89. Springer.

(CAPSL 2011) CAPSL (2011) *"Centre for Learning Technology."* Available online at www.tcd.ie/CAPSL. Last Checked: 22 September 2011

(Captive5 2010) Captive5 (2010) *"Captive 5 Authoring Tool."* Available online at <http://www.adobe.com/products/captivate>. Last Checked: 22 September 2011

(Carro 2002) Carro, M. R. (2002) *"Adaptive Hypermedia in Education: New Considerations and Trends."* 6th World Multiconference on Systemics, Cybernetics and Informatics, Orlando, Florida, USA, July 2002. IIS.

(Carver, Hill, et al. 1999) Carver, C., J. Hill, U. Pooch (1999) *"Third Generation Adaptive Hypermedia Systems."* WebNet World Conference on the WWW and Internet, Honolulu, Hawaii, USA, October 1999, Vol. 1/2. AACE

(Casti 1996) Casti, J. (1996) *"Would-be Worlds: How Simulation is Changing the Frontiers of Science."* New York: John Wiley.

(Catlab 2011) *"Catlab"* Available online at <http://www.emescience.com/bio-software-catlab.html>. Last Checked: 22 September 2011.

(Chou 2010) Chou, C. (2010) *"Investigating the effects of incorporating collaborative action research into an in-service teacher training program."* Procedia Social and Behavioral Sciences, Vol. 2, 2728–2734. Science Direct.

(Churcher, Atwell et al. 1997) Churcher, G., E. Atwell, C. Souter (1997) *"Dialogue Management Systems: a Survey and Overview."* Report 97.6, School of Computer Studies, University of Leeds. Available online at <http://www.cs.cmu.edu/~dod/papers/churcher97.pdf>. Last Checked: 22 September 2011

(Conati and Zhao 2004) Conati, C., X. Zhao (2004) *"Building and Evaluating an Intelligent Pedagogical Agent to Improve the Effectiveness of an Educational Game."* 9th International Conference on Intelligent User Interfaces, Funchal, France, January 2004. ACM.

(Conklin 1987) Conklin, J. (1987) *"Hypertext: An Introduction and Survey."* IEEE Computer, Vol. 20 Issue 9, 17-41. IEEE.

(Conlan 2005) Conlan, O. (2005) *"The Multi-Model, Metadata Driven Approach to Personalised eLearning Services."* PhD Thesis, School of Computer Science and Statistics, Trinity College Dublin, Dublin. Available online at http://www.scss.tcd.ie/Owen.Conlan/publications/Conlan_Thesis.pdf. Last Checked: 22 September 2011.

(Conlan, Dagger, et al. 2002) Conlan, O., D. Dagger, V. Wade (2002) *"Towards a Standards-based Approach to e-Learning Personalization using Reusable Learning Objects."* 7th World Conference on E-Learning in Corporate, Government, Healthcare, & Higher Education, Montreal, Canada October 2002. AACE.

(Conlan and Wade 2004) Conlan, O., V. Wade (2004) *"Evaluation of APeLS-An adaptive eLearning service based on the multi-model, metadata-driven approach."* 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Eindhoven, Netherlands, August 2004. Springer-Verlag.

(Conlan, Wade et al. 2002) Conlan, O., V. Wade, C. Bruen, M. Gargan (2002) *"Multi-model, Metadata Driven Approach to Adaptive Hypermedia Services for Personalized eLearning."* 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Malaga, Spain, May 2002. Springer-Verlag.

(Connolly, Johnson, et al. 1998) Connolly, C., J. Johnson, C. Lexa (1998) *"AVATAR: An Intelligent Air Traffic Control Simulator and Trainer. Intelligent Tutoring Systems."* Lecture Notes on Computer Science. Vol. 1452, 534-543 Springer

(Conrick 1998) Conrick, M. (1998) *"Computer Based Education: more than just a package."* The Australian Electronic Journal of Nursing Education, Vol. 4 Issue 1. AEJN. Available online at http://www.scu.edu.au/schools/nhcp/aejne/archive/vol4-1/mconrickvol4_1.html. Last Checked: 1 April 2012.

(Corno and Snow 1986) Corno, L., E. Snow (1986) *"Adapting Teaching to Individual Differences among Learners."* Handbook of Research on Teaching. Edited by M. Wittrock. Macmillan.

(Cordova and Lepper) Cordova, D., M. Lepper (1996) *"Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice."* Journal of Educational Psychology, Vol. 88 Issue 4, 715-730. APA Psyc NET.

(Costaridou, Panayiotakis, et al. 1996) Costaridou, L., G. Panayiotakis, N. Pallikarakis, B. Proimos (1996) *"Radiographic skills learning: procedure simulation using adaptive hypermedia."* British Institute of Radiology, Vol. 69, 938-945. BJR. Available online at <http://bjr.birjournals.org/content/69/826/938.full.pdf+html>. Last Checked: 1 April 2012.

(CourseLab 2012) CourseLab (2012) Available online at <http://courselab.com>. Last Checked: 1 April 2012.

(Cristea and Calvi 2003) Cristea, A., L. Calvi (2003) *"The three Layers of Adaptation Granularity."* User Modelling Conference, June 22-26, 2003, Johnstown, PA, USA. Springer.

(Cristea and de Mooij 2003) Cristea, A., A. de Mooij (2003) *"Adaptive Course Authoring: My Online Teacher."* 10th International Conference on Telecommunications, Feb. 23 - March 1 2003, Tahiti, French Polynesia. IEEE.

(Cristea and de Mooij 2003a) Cristea, A., A. de Mooij (2003) *"LAOS: Layered WWW AHS Authoring Model and their corresponding Algebraic Operators."* 12th International World Wide Web Conference, 12-24 May, Budapest, Hungary. Available online at <http://www.ra.ethz.ch/cdstore/www2003/papers/alternate/P301/p301-cristea.pdf>. Last Checked: 1 April 2012.

(Cristea and Stewart 2004) Cristea, A., C. Stewart (2004) *"Authoring of Adaptive Hypermedia."* Advances in Web-Based Education: Personalized Learning Environments. Information Science Publishing.

(Cumming, Okamoto, et al. 1999) Cumming, G., T. Okamoto, L. Gomez (1999) *"Advanced Research in Computers and Communications in Education."* Amsterdam: IOS Press.

(Dagger 2006) Dagger, D. (2006) *"Personalised eLearning Development Environments."* PhD. Computer Science. Dublin, Trinity College Dublin. Available online at https://www.scss.tcd.ie/Declan.Dagger/publications/thesis/Declan_Dagger_PhD_Thesis_2006.pdf. Last Checked: 1 April 2012

(Dagger, Rogers, et al. 2007) Dagger, D., C. Rogers, V. Wade, C. Gaffney, K. Armstrong, B. Fitzmaurice, M. Gill, E. Walsh, E. *"Adaptive Simulations for Communication Skills Training in Healthcare."* 2nd International Workshop on Personalisation for E-Health at the 11th International Conference on User Modelling, Corfu,

Greece, June 2007. Springer. Available online at <http://www.csc.liv.ac.uk/~floriana/Pers4eHealth/proceedings-um/W7.pdf#page=21>. Last Checked: 1 April 2012.

(Dagger and Wade 2005) Dagger, D., V. Wade (2005) *"Evaluation of adaptive course construction toolkit (ACCT)."* 3rd International Workshop on Authoring of Adaptive and Adaptable Educational Hypermedia, 12th International Conference on Artificial Intelligence in Education, 18-22 July 2005, Amsterdam, Netherlands. Available online at http://www.dcs.warwick.ac.uk/~acristea/AAAEH05/cameraReady/6-11-a3eh_daggerd_IOS_format_v1.1.doc. Last Checked: 1 April 2012.

(Dagger, Wade, et al. 2004) Dagger, D., V. Wade, O. Conlan (2004) *"Developing Adaptive Pedagogy with the Adaptive Course Construction Toolkit."* Conference of Authoring of Adaptive and Adaptable Hypermedia, Eindhoven, Netherlands, August 23 - 26. Available online at https://www.scss.tcd.ie/Owen.Conlan/publications/ah2004ws_daggerd_v1.pdf. Last Checked: 1 April 2012.

(Dalgarno 2001) Dalgarno, B. (2001) *"Interpretations of constructivism and consequences for Computer Assisted Learning."* British Journal of Educational Technology, Vol. 32 Issue 2, 183-194. Wiley.

(Davis and Fill 2007) Davis, H., K. Fill (2007) *"Embedding blended learning in a university's teaching culture: Experience and reflections."* British Journal of Educational Technology, Vol. 33 Issue 5, 817-828. Wiley.

(Davis and White 2005) Davis, H., S. White (2005) *"A Research-Led Curriculum in Multimedia: Learning about Convergence."* 10th Annual Conference on Innovation and Technology in Computer Science Education, 27-29 June, Lisbon. ACM.

(De Bra, Aerts, et al. 2003) De Bra, P., A. Aerts, B. Berden, B. de Lange, B. Rousseau, T. Santic, D. Smits, N. Stash (2003) *"AHA! The adaptive hypermedia architecture."* 14th Conference on Hypertext and Hypermedia, August 26-30, Nottingham, UK . ACM.

(De Bra and Calvi 1998) De Bra, P., L. Calvi (1998) *"AHA: a Generic Adaptive Hypermedia System."* 2nd Workshop on Adaptive Hypertext and Hypermedia, Hypertext Conference, 20-24 June, Pittsburgh, USA. ACM.

(De Bra, Eklund, et al. 1999) De Bra, P., J. Eklund, P. Brusilovsky, W. Hall, A. Kobsa (1999) *"Adaptive hypermedia: Purpose, methods, and techniques.(Panel)"* 10th Conference on Hypertext, 21-25 February, Darmstadt, Germany. New York: ACM.

(De Bra, Houben, et al. 1999) De Bra, P., G.-J. Houben, H. Wu (1999) *"AHAM: a Dexter-based reference model for adaptive hypermedia."* 10th Conference on Hypertext and hypermedia, 21-25 February, Darmstadt, Germany. New York: ACM.

(De Bra, Stash, et al. 2007) De Bra, P., N. Stash, D. Smits, C. Romero, S. Ventura (2007) *"Authoring and management tools for adaptive educational hypermedia systems: The AHA! case study."* Studies in Computational Intelligence, Vol. 62, 285-308. Springer. Available online at <http://www.als-project.org/download.php?b65b958f576fbed844f51004eba23840>. Last Checked: 1 April 2012.

(De Jong and Van Joolingen 1998) De Jong, T., W. R. Van Joolingen (1998) *"Scientific discovery learning with computer simulations of conceptual domains."* Review of Educational Research, Vol. 68, 179-201. Sage.

(Dede 1996) Dede, C. (1996) *"The evolution of distance education: Emerging technologies and distributed learning."* The American Journal of Distance Education, Vol. 10 Issue 2, 4-36. Routledge.

(Dempsey 2009) Dempsey, J. (2009) *"Interactive Simulation in the Teaching of Web Programming."* Masters of Science in Technology & Learning, Computer Science. Dublin, Trinity College Dublin.

(Dewey 1938) Dewey, J. (1938) *"Experience and education."* New York: Collier Macmillan Publishers.

(Dix 2004) Dix, A. J. (2004) *"Human-computer interaction."* Harlow, Pearson.

(Dorin, Demmin, et al. 1990) Dorin, H., P. Demmin, D. Gabel (1990) *"Chemistry: The study of matter."* Englewood Cliffs, NJ: Prentice Hall, Inc.

(Drossos, Giotopoulos, et al. 2005) Drossos, L., K. Giotopoulos, B. Vassiliadis, E. Sakkopoulos, K. Votis, A. Stefani, A. Tsakalidis, M. Xenos, S. Likothanassis (2005) *"On the Personalization of an E-learning Environment in Higher Education using Graphs."* 10th Panhellenic Conference in Informatics, 11-13 November, Volos, Greece. Available online at <http://quality.eap.gr/Publications/XM/Conferences%20English/C40%20-%20Personalization%20of%20an%20E-learning%20Environment.pdf>. Last Checked: 1 April 2012.

(Duffy and Cunningham 1996) Duffy, T., D. Cunningham (1996) *"Constructivism: Implications for the Design and Delivery of Instruction."* Handbook of Research on Educational Communications and Technology. Edited by J. Spector, M. Merrill, J. van Merriënboer, M. Driscoll. New York: Simon & Schuster.

(Dwyer 1974) Dwyer, F. M. (1974) *"Strategies for improving visual learning."* University Park: Pennsylvania State University, Learning Services.

(Eastmond and Ziegahn 1996) Eastmond, D., L. Ziegahn (1996) *"Instructional design for the online classroom."* Computer mediated communication and the online classroom. Edited by Z. L. Berge, M. P. Collins, 59-80. Hampton Press.

(Eclipse 2012) Eclipse (2012) Available online at <http://www.eclipse.org/>. Last Checked: 1 April 2012.

(Ehrig, Ermel, et al. 2005) Ehrig, K., C. Ermel, S. Hansgen, G. Taentzer (2005) *"Generation of visual editors as eclipse plug-ins."* 20th IEEE/ACM International Conference on Automated software engineering, 7-11 Nov., Long Beach, CA, USA, 134-143. ACM.

(Eklund and Brusilovsky 1998) Eklund, J., P. Brusilovsky (1998) *"The value of adaptivity in hypermedia learning environments: A short review of empirical evidence."* Adaptive Hypertext and Hypermedia Workshop, 9th ACM International Hypertext Conference Hypertext, June 20-24, Pittsburgh, USA. ACM.

(ELEKTRA 2008) ELEKTRA (2008) *"Enhanced Learning Experience and Knowledge TRANSfer."* Available online at <http://www.elektra-project.org/>. Last Checked: 1 April 2012.

(Emmendorfer 2009) Emmendorfer, J. (2009) *"Teach Soft Skills With e-Learning."* Technicl Report. Available online at <http://www.ame-learning.com/revolution/wp-content/uploads/softskills.pdf>. Last Checked: 1 April 2012.

(ETU 2012) ETU (2012) *"Empower the User"* Available online at <http://etu.ie/site/>. Last Checked: 1 April 2012.

(ExperienceBuilder 2012) ExperienceBuilder (2012) *"Experience Builder Authoring Tool."* Available online at <http://www.experiencebuilders.com/>. Last Checked: 1 April 2012.

(Feinstein and Cannon 2002) Feinstein, A. H., H. M. Cannon (2002) *"Constructs of simulation evaluation."* Simulation & Gaming, Vol. 32, 425-440. Sage.

(Fish and Harter 1991) Fish, D., R. Harter (1991) *"Integrated Training and Reusable Simulations."* Integrated Training and Reusable Simulations, Florida, USA. I/ITSEC.

(Fitzmaurice, Armstrong, et al. 2007) Fitzmaurice, B., K. Armstrong, V. Carroll, D. Dagger, M. Gill (2007) *"Virtual Interviews for Students Interacting Online for Psychiatry (VISION): a novel resource for learning clinical interview skills."* Psychiatric Bulletin Vol. 31, 218-220. The Psychiatrist. Available online at <http://pb.rcpsych.org/content/31/6/218.full.pdf+html>. Last Checked: 1 April 2012.

(Forinash and Wisman 2001) Forinash, K., R. Wisman (2001) *"The viability of distance education science laboratories."* T H E Journal (Technological Horizons In Education), Vol. 29 Issue 2, 38-44.

(Fu 1989) Fu, L. (1989) *"Integration of neural heuristics into knowledge-based inference."* International Joint Conference on Neural Networks IJCNN, June 18-22, Washington DC, USA. IEEE. Available online at <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=118425>. Last Checked: 1 April 2012.

(Gaffney 2006) Gaffney, C. (2006) *"ADAPT Composition Tool Sub-Dialogue Feature Comparison Questionnaire."* Available online at <http://www.scribd.com/doc/133767751/Phase-1-i-e-Questionnaire>. Last Checked: 1 April 2013.

(Gaffney 2007) Gaffney, C. (2007) *"Sample of Initial Phase 2 Storyboards."* Available online at <http://www.scribd.com/doc/133767750/Sample-Phase-2-Storyboards>. Last Checked: 1 April 2012.

(Gaffney 2008) Gaffney, C. (2008) *"Telecommunication Simulation Evaluation - Questionnaire."* Available online at <http://www.scribd.com/doc/133767749/Phase-2-i-e-Questionnaire>. Last Checked: 1 April 2013.

(Gaffney 2013) Gaffney, C. (2013) ACTSim video screen capture demo. Available online at <http://www.youtube.com/watch?v=xTQGUotTOFg> . Last Checked: 1 April 2013.

(Gaffney, Dagger et al. 2010) Gaffney, C., D. Dagger, V. Wade (2010) *"Authoring and Delivering Personalised Simulations — an Innovative Approach to Adaptive eLearning for Soft Skills."* Journal of Universal Computer Science: Special Issue on Authoring of Adaptive and Adaptable Hypermedia, Vol. 16 Issue 19. Available online at http://www.jucs.org/jucs_16_19/authoring_and_delivering_personalised/ . Last Checked: 22 September 2011. J.UCS.

(Gaffney, Dagger et al. 2008) Gaffney, C., D. Dagger, V. Wade (2008) *"Evaluation of ACTSim: a Composition Tool for Authoring Adaptive Soft Skill Simulation."* 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Hannover, Germany, July 2008. Springer.

(Gaffney, Dagger et al. 2008) Gaffney, C., D. Dagger, V. Wade (2008) "*A State of the Art Survey of Soft Skill Simulation Authoring Tools.*" 19th ACM Conference on Hypertext and Hypermedia, Pittsburgh, USA, June 2008. ACM.

(Gaffney, Dagger et al. 2007a) Gaffney, C., D. Dagger, V. Wade (2007) "*Supporting personalised simulations: a pedagogic support framework for modelling and composing adaptive dialectic simulations.*" Summer Computer Simulation Conference, San Diego, USA, July 2007. Available online at <http://dl.acm.org/citation.cfm?id=1358093&bnc=1>. Last Checked: 22 September 2011. ACM.

(Gaffney, Dagger et al. 2007) Gaffney, C., D. Dagger, V. Wade (2007) "*Towards a Composition Tool for Personalised Simulations.*" A3H: Fifth International Workshop on Authoring of Adaptive and Adaptable Hypermedia, 11th International Conference on User Modeling, Corfu, Greece, June 2007. Springer.

(Gibbons, Fairweather, et al. 1997) Gibbons, A. S., P. G. Fairweather, T. Anderson, M. Merrill (1997) "*Simulation and Computer-Based Instruction: A Future View.*" Instructional Development Paradigms. Edited by C. R. Dills, A. J. Romiszowski. NJ: Educational Technology Publication.

(Goodyear, Njoo, et al. 1991) Goodyear, P., M. Njoo, J.J.A. van Berkum (1991) "*Learning processes, learner attributes and simulations.*" Education & Computing, Vol. 6, 263-304. Eric.

(Graesser, Person, et al. 1995) Graesser, A. C., N. Person, J. Magliano (1995) "*Collaborative dialog patterns in naturalistic one-to-one tutoring.*" Applied Cognitive Psychology, Vol. 9, 359-387. Available online at http://www.cs.pitt.edu/~chopin/references/tig/graesser_person.pdf.pdf. Last Checked: 1 April 2012.

(Granland, Bergland, et al. 2000) Granland, R., E. Bergland, H. Eriksson (2000) "*Designing web-based simulation for learning.*" Future Generation Computer Systems, Vol. 17, 171-185. Elsevier.

(GRAPPLE 2011) GRAPPLE (2011) "*The Generic Responsive Adaptive Personalized Learning Environment.*" Description. Available online at <http://www.grapple-project.org/summary>. Last Checked: 1 April 2012.

(Gruender and Tobin 1991) Gruender, C. D., K. G. Tobin (1991) "*Promise and prospect.*" Science Education, Vol. 75. Wiley.

(Gustafson, Elmberg et al. 1998) Gustafson, J., P. Elmberg, R. Carlson, A. Jonsson (1998) "*An Educational Dialogue System with a User Controllable Dialogue Manager.*" 5th International Conference on Spoken Language Processing, Sydney, Australia, November 1998. ICSLP.

(Harrigan, Kravcik, et al. 2009) Harrigan, M., M. Kravčik, C. Steiner, V. Wade (2009) *"What Do Academic Users Really Want from an Adaptive Learning System?"* User Modeling, Adaptation, and Personalization Lecture Notes in Computer Science, Vol. 5535, 454-460. Springer.

(Hassenzahl, Beau, et al. 2001) Hassenzahl, M., Beu, A., M. Burmester (2001) *"Engineering joy."* IEEE Software: Ergonomic requirements for office work with visual display terminals (VDTs): Guidance on usability, Vol. 18 Issue 1, 70-76. IEEE.

(Heffernan, Turner, et al. 2006) Heffernan, N., T. Turner, A. Lourenco, M. Macasek, G. Nuzzo-Jones, K. Koedigner (2006) *"The ASSISTment Builder: Towards an Analysis of Cost Effectiveness of ITS Creation."* International Florida Artificial Intelligence Research Society Conference (FLAIRS '06), Florida, USA, May 11-13, 2006. AAAI Press.

(Heinich, Molenda, et al. 1999) Heinich, R., M. Molenda, J. Russell, S. Smaldino (1999) *"Instructional Media and Technologies for Learning."* Upper Saddle River, NJ: Prentice-Hall.

(Hendricks 2006) Hendricks, C. (2006) *"Improving schools through action research: A comprehensive guide for educators."* Pearson.

(Henning 1998) Henning, P. (1998) *"Everyday Cognition and Situated Learning. Handbook of Research on Educational Communications and Technology."* Edited by D. H. Jonassen. New York: Simon & Schuster.

(Heron 1996) Heron, J. (1996) *"Co-operative inquiry : research into the human condition."* London, SAGE.

(Herrington, Oliver, et al. 2003) Herrington, J., R. Oliver, T. Reeves (2003) *"Patterns of engagement in authentic online learning environments."* Australian Journal of Educational Technology Vol. 19 Issue 1, 59-71. Australasian Society for Computers in Learning in Tertiary Education.

(Hertzner and Pannese 2011) Hertzner, S., L. Pannese (2011) *"Adaptive Simulation For Multicultural Dialogue Training"* International Conference The Future of Education, Milan, Italy, May 2011. Pixel. Available online at http://www.pixel-online.net/edu_future/common/download/Paper_pdf/ENT28-Hetzner.pdf. Last Checked: 1 April

(Holtzblatt and Beyer 1993) Holtzblatt, K., H. Beyer (1993) *"Making Customer-Centered Design Work for Teams."* Communications of the ACM, Vol. 36 Issue 10, 93-103. ACM.

(Holtzblatt and Jones 1993) Holtzblatt, K., S. Jones (1993) *"Contextual Inquiry: A participatory technique for system design."* Participatory Design: Principles and Practice. Edited by D. Schuler and A. Namioka. Routledge.

(Hubal, Kizakevich, et al. 2000) Hubal, R., P. Kizakevich, C. Guinn, K. Merino, S. West (2000) *"The virtual standardized patient."* Stud. Health Technol. Inform, Vol. 70, 133–138. ISO Press.

(Huisman and Vries 1991) Huisman, W., F. J. Vries (1991) *"Functions and design of educational simulation models."* Centre for educational production. H. Open Universiteit.

(Hung and Chen 2002) Hung, D., D. Chen (2002) *"Two kinds of scaffolding: The dialectical process within the authenticity-generalizability (A-G) continuum."* Education Technology & Society, Vol. 5 Issue 4, 148-153. International Forum of Educational Technology & Society.

(Hunicke and Chapman 2004) Hunicke, R., V. Chapman (2004) *"AI for dynamic difficulty adjustment in games."* Challenges in Game AI Workshop, 19th National Conference on Artificial Intelligence, 25-29 July, San Jose, California, USA. AAAI

(ISO 2012) ISO (2012) *"International Organization of Standardization."* Available online at www.iso.org. Last Checked: 1 April 2012.

(Jarz, Kainz, et al. 1997) Jarz, E. M., G. A. Kainz, G. Walpoth (1997) *"Multimedia-based case studies in education: Design, development, and evaluation of multimedia-based case studies."* Journal of Educational Multimedia and Hypermedia, Vol. 6 Issue 1, 23-46. AACE.

(Johnson 2003) Johnson, W. L. (2003) *"Interaction tactics for socially intelligent pedagogical agents."* 8th International Conference on Intelligent User Interfaces, 12-15 January, Miami, FL, USA. ACM Press, New York.

(Johnson, Wang, et al. 2007) Johnson, W. L., N. Wang, S. Wu (2007) *"Experience with serious games for learning foreign languages and cultures."* SimTecT Conference, 4-7 June, Brisbane, Australia. Mendeley.

(Jokela and Iivari 2003) Jokela, T., N. Iivari (2003) *"The standard of user-centered design and the standard definition of usability: analyzing ISO 13407 against ISO 9241-11."* Latin American conference on Human-computer interaction, 17-20 November, Rio de Janeiro, Brazil. ACM.

(Jonassen, Dyer, et al. 1997) Jonassen, D., D. Dyer, K. Peters, T. Robinson, D. Harvey, M. King, P. Loughner (1997) *"Cognitive flexibility hypertexts on the Web: Engaging learners in making meaning."* Web-based instruction. Edited by B. H. Khan, E. Cliffs, 119 -133. NJ: Educational Technology Publications.

(Jonassen 1999) Jonassen, D. H. (1999) *"Designing constructivist learning environments."* Instructional Design Theories and Models, Vol. 2: A New paradigm of instructional theory: Edit by C. M. Reigeluth. NJ: Lawrence Erlbaum Associates.

(Kaplan, Fenwick, et al. 1993) Kaplan, C., J. Fenwick, J. Chen (1993) *"Adaptive hypertext navigation based on user goals and context."* User Models and User Adapted Interaction, Vol. 3 Issue 3, 193-220. UMUAI.

(Kay and Goldberg 1977) Kay, A., A. Goldberg (1977) *"Personal dynamic media."* IEEE Computer Vol. 10 Issue 3, 31-41.IEEE.

(KDSimStudio 2011) " *KDSimStudio: Simulated Conversation Development Tool* " Available online at <http://www.kdsimstudio.com/DownloadKDSimStudio.html>. Last Checked: 22 September 2011.

(Kearsley and Shneiderman 1998) Kearsley, G., B. Shneiderman (1998) *"Engagement theory: A framework for technology-based teaching and learning."* Educational Technology, Vol. 38 Issue 5, 20-23. JTE.

(Keller and Suzuki 1988) Keller, J. M., K. Suzuki (1988) *"Use of the ARCS motivation model in courseware design."* Instructional designs for microcomputer courseware. Edited by D. Jonassen, 401-434. NJ: Lawrence Erlbaum Associates.

(Kember 2000) Kember, D. (2000) *"Action Learning and Action Research. Improving the Quality of Teaching and Learning."* Sterling, VA: Stylus Publishing.

(Kinshuk 2004) Kinshuk, L. T. (2004) *"Improving mobile learning environments by applying mobile agents technology."* Third Pan Commonwealth Forum on Open Learning, 4-8 July, Dunedin, New Zealand. Commonwealth of Learning.

(Kirschner and Huisman 1998) Kirschner, P., W. Huisman (1998) *"Dry laboratories in science education: computer-based practical work."* International Journal of Science Education, Vol. 20 Issue 6, 665-682. Mendley.

(Kuo and Chen 2001) Kuo, Y.-F., L.-S. Chen (2001) *"Personalization technology application to internet content provider."* Expert Systems with Applications, Vol.21 Issue 4, 203-215. Elseiver.

(Land and Hannafin 2000) Land, S., M. Hannafin (2000) *"Student-centered learning environments."* Theoretical foundations of learning environments. Edited by D. H. Jonassen and S. Land. Lawrence Erlbaum Associates.

(Langone, Clees, et al. 2005) Langone, J., T. J. Clees, L. Rieber, M. Matzko (2005) *"The future of computer-based interactive technology for teaching individuals with moderate to severe disabilities: Issues relating to research and practice."* Journal of Special Education Technology, Vol. 18 Issue 10, 5-16. JSET.

(Lave and Wenger 1991) Lave, J., E. Wenger (1991) *"Situated Learning. Legitimate peripheral participation."* Cambridge: University of Cambridge Press.

(Law and Kickmeier-Rust 2008) Law, E. L.-C., M. D. Kickmeier-Rust (2008) *"80Days: Immersive digital educational games with adaptive storytelling."* 1st International Workshop on Story-Telling and Educational Games, September 16, Maastricht, The Netherlands. CEUR-WS.

(Leemkuil, de Jong, et al. 2003) Leemkuil, H., T. de Jong, R. de Hoog, N. Christoph (2003) *"KM QUEST: A collaborative Internet-based simulation game."* Simulation & Gaming, Vol. 31 Issue 1, 89-111. Sage.

(Lepper and Chabay 1985) Lepper, M. R., R. W. Chabay (1985) *"Intrinsic motivation and instruction: Conflicting views on the role of the motivational processes in computer-based education."* Educational Psychologist, Vol 20, 217-231. Routledge.

(Leutner 1993) Leutner, D. (1993) *"Guided discovery learning with computer-based simulation games: Effects of adaptive and non-adaptive instructional support."* Learning and Instruction, Vol.3, 113-132. Elsevier.

(Lim 2004) Lim, C. (2004) *"Engaging Learners in Online learning Environments."* TechTrends, Vol. 48 Issue 4, 16-23. Springer.

(Lopes and Bidarra 2011) Lopes, R., R. Bidarra. (2011) *"Adaptivity challenges in games and simulations: a survey."* IEEE Transactions on Computational Intelligence and AI in Games, Vol. 3 Issue 2, 85-99. IEEE.

(Lunce 2006) Lunce, L. (2006) *"Simulations: Bringing the benefits of situated learning to the traditional classroom."* Journal of Applied Educational Technology, Vol.3 Issue 1, 37-45. JAET.

(Luz 2000) Luz, S. (2000) *"State-of-the-art survey of existing dialogue management tools."* Technical Report. Available online at <http://www.google.com/url?sa=t&rct=j&q=%22state-of-the-art%20survey%20of%20existing%20dialogue%20management%20tools.%22&source=web&cd=1&ved=0CC>

MQFjAA&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Bjsessionid%3D347FE3793AA6A66BDACD343083526FE7%3Fdoi%3D10.1.1.16.6297%26rep%3Drep1%26type%3Dpdf&ei=D7KBT462DsHMhAea2K3TBw&usg=AFQjCNHgSrksBUyDH9_CrqEEzBArDARRDQ&cad=rja. Last Checked: 1 April 2012.

(Macaulay 1995) Macaulay, L. (1995) *"Human - computer interaction for software designers."* London, International Thomson Computer Press.

(Malone 1981) Malone, T. W. (1981) *"Toward a theory of intrinsically motivating instruction."* Cognitive Sci, Vol.5 Issue 4, 333-369. Wiley.

(Malone and Lepper 1987) Malone, T. W., M. R. Lepper (1987) *"Making learning fun: a taxonomy of intrinsic motivations for learning."* Aptitude, Learning and Instruction: Cognitive and Affective Process Analysis. Edited by R. E. Snow and M. J. Farr, Vol. 3. Lawrence Erlbaum, Hillsdale, NJ.

(Margo, Swarz, et al. 2010) Magro, A., J. Swarz, A. Ousley (2010) *"CancerSPACE: An interactive e-learning tool aimed to improve cancer screening rates."* Journal of Computer-Mediated Communication, Vol. 15, Issue 3, 482-499. Wiley.

(Markulis and Strang 1985) Markulis, P. M., D. Strang (1985) *"Tire Use of Decision Support Systems (DSS) and Operations Research/Management Science (OR/MS) Techniques to Enhance the Learning Experience of Student: Participating in Computerized Simulations."* Developments in Business Simulation and Experiential Exercises. Edited by J. W. G. a. A. C. Burns. Vol. 12, 30-34. Wordpress.

(Martin and Waag 1978) Martin, E. L., W. L. Waag (1978) *"Contributions of platform motion to simulator training effectiveness: Study I-Basic contact."* Technical Report. Brooks Air Force Base, TX: Air Force Human Resources Laboratory.

(Mayes and de Freitas 2004) Mayes, T., S. de Freitas (2004) *"Review of e-learning theories, frameworks and models."* JISC e-Learning Models Desk Study. Available online at http://www.jisc.ac.uk/uploaded_documents/Stage%20%20Learning%20Models%20%28Version%201%29.pdf . Last Checked: 1 April 2012.

(McClelland 1965) McClelland, D. C. (1965) *"Toward a theory of motive acquisition."* American Psychologist, Vol. 33, 201-211. APA.

(McHaney 1991) McHaney, R. (1991) *"Computer Simulation: A Practical Perspective."* Academic Press, New York, NY.

(Mergel 1998) Mergel, B. (1998) *"Instructional Design and Learning Theory."* Professional and Theoretical Issues in Educational Technology Available online at http://members.iinet.net.au/~aamcarthur/11_March_2008_files/Learning_Theories_of_Instructional_Design.pdf. Last Checked: 1 April 2012.

(Merrill 1991) Merrill, M. D. (1991) *"Constructivism and instructional design."* Educational Technology, Vol. 31 Issue 5, 45-53. JTE.

(Miller 1974) Miller, G. E. (1974) *"Some considerations in the design and utilization of simulators for technical training."* Alexandria, VA: Air Force Human Resource Laboratory. ERIC. Available online at <http://www.eric.ed.gov/PDFS/ED097919.pdf>. Last Checked: 1 April 2012.

(Miller 1978) Miller, J. G. (1978) *"Living systems."* New York: McGraw-Hill.

(Milosavljevic 1997) Milosavljevic, M. (1997) *"Augmenting the user's knowledge via comparison."* 6th International Conference on User Modelling, 2-5 June, Chia Laguna, Sardinia. Springer Wien New York.

(Min 2001) Min, R. (2001) *"Designing dynamical learning environments for simulation: Micro-worlds applets on the World Wide Web."* 6th EARLI, SIG, 27-29 June, Erfurt, Germany. ACM.

(Mitchell, Chen, et al. 2005) Mitchell, T. J. F., S. Y. Chen, R. D. Macredie (2005) *"Cognitive styles and adaptive Web-based learning."* Psychology of Education Review, Vol. 29 Issue 1, 34-42. Springer.

(Microsoft Flight 2012) Microsoft Flight (2012) Available online at <http://www.microsoft.com/games/flight/>. Last Checked: 1 April 2012

(Moore, Burtan, et al. 1996) Moore, D., J. Burtan, R. Myres (1996) *"Multiple-channel communication: the theoretical and research foundations of multimedia."* Handbook of research on educational communications and technology. Edited by D. Jonassen. New York: Simon & Schuster.

(Moreno-Ger, Blesius et al. 2008) Moreno-Ger, P., C. Blesius, P. Currier, J. Sierra, B. Fernandez-Manjon (2008) *"Online Learning and Clinical Procedures: Rapid Development and Effective Deployment of Game-Like Interactive Simulations"* Lecture Notes in Computer Science, Vol. 5080. Springer.

(Moreno-Ger, Burgos, et al. 2008) Moreno-Ger, P., D. Burgos, I. Martínez-Ortiz, J. L. Sierra, B. Fernández-Manjón (2008) *"Educational Game Design for Online Education."* Computers in Human Behavior, Vol. 24, 2530-2540. Elsevier.

(Moreno-Ger Burgos, et al. 2007) Moreno-Ger, P., D. Burgos, et al. (2007) *"A Game-Based Adaptive Unit of Learning with IMS Learning Design and <e-Adventure>."* 2nd European Conference on Technology Enhanced Learning, 17-20 September, Crete, Greece. Available online at http://www.e-ucm.es/drafts/e-UCM_draft_72.pdf. Last Checked: 1 April 2012.

(MOT 2010) MOT (2010) *"MOT: My Online Teacher."* <http://www.dcs.warwick.ac.uk/~acristea/mot.html>. Available online at <http://www.dcs.warwick.ac.uk/~acristea/mot.html>. Last Checked: 1 April 2012.

(Musa and de Oliveira 2005) Musa, D. L., J. P. M. de Oliveira (2005) *"Integration of Distributed Learner Information through the Web Service."* Available online at <http://www.sbc.org.br/ifipacademy2005/11247.pdf>. Last Checked: 1 April 2012.

(Mustafa and Sharif 2010) Mustafa, Y., M. Sharif (2010) *"An Approach to Adaptive E-Learning Hypermedia System based on Learning Styles (AEHS-LS): Implementation and Evaluation."* International Journal of Library and Information Science, Vol. 3, Issue 1, 15-28. Academic Journals.

(Mac Isaac 1995) Mac Isaac, D. (1995) *"An Introduction to Action Research"*. Available online at <http://physicsed.buffalostate.edu/danowner/actionrsch.html>. Last Checked: 11 December 2011

(McNiff and Whitehead 2011) McNiff, J., Whitehead, J. (2006) *"All you need to know about action research."* Thousand Oaks, CA: Sage.

(NDLR 2012) NDLR (2012) *"National Digital Learning Repository."* Available online at www.ndlr.ie. Last Checked: 1 April 2012.

(Neelamkavi 1987) Neelamkavi, F. (1987) *"Computer simulation and modelling."* New York: Wiley.

(NexLearn 2010) NexLearn (2010) *"SimWriter Authoring Tool."* Available online at <http://www.nexlearn.com/?q=node/1>. Last Checked: 1 April 2012.

(Nielsen and Landauer 1993) Nielsen, J., T. K. Landauer (1993) *"A mathematical model of the finding of usability problems."* Conference on Human Factors in Computing Systems, Amsterdam, Netherlands. ACM.

(Nielsen and Molich 1990) Nielsen, J., R. Molich (1990) *"Heuristic Evaluation of User Interfaces."* Conference on Human Factors in Computing Systems, Amsterdam, Netherlands. ACM.

(Nkambou, Bourdeau et al. 2011) Nkambou, R., J. Bourdeau, V. Psyché (2011) *"Building Intelligent Tutoring Systems: An Overview."* Studies in Computational Intelligence, Vol. 308, 361-375. Springer.

(Noble 2002) Noble, C. (2002) *"The Relationship Between Fidelity and Learning in Aviation Training and Assessment."* Journal of Air Transportation World Wide, Vol. 7 Issue 2-3, 33-54. Aviation Institute.

(Northover and Wilson 2004) Northover, S., M. Wilson (2004) *"SWT : The Standard Widget Toolkit Volume 1."* Addison-Wesley Professional.

(Nurjanah and Davis 2012) Nurjanah, D., H. Davis (2012) "Improving the Workspce Awreness of Authors in Asynchronous Collaborative Authoring of Learning Designs." EdMedia - World Conference on Educational Media and Technology, Denver, USA, 26-29 June. AACE.

(Odhayani and Ratnapalan 2011) Odhayani, A., S. Ratnapalan (2011) *"Teaching communication skills."* Canadian Family Physician, Vol. 57, Issue 10, 1216-1218. CFP.

(Ohene-Djan, Gorle, et al. 2003) Ohene-Djan, J., M. Gorle, C. Bailey, G. Wills, H. Davis (2003) *"Understanding Adaptive Hypermedia: An Architecture for Personalisation and Adaptivity."* Workshop in Adaptive Hypermedia and Adaptive Web-Based Systems, 14th Conference on Hypertext and Hypermedia, Nottingham, UK, 26-30 August. ACM.

(Olsen, Sellers, et al. 1999) Olsen, D. E., W. A. Sellers, R. G. Phillips (1999) *"The simulation of a human subject for interpersonal skills training."* Interservice/Industry Training, Simulation and Education Conference (IITSEC), Orlando, FL. I/ITSEC.

(O'Shea and Reddy 2007) O'Shea, M., L. Reddy (2007) *"Action change in New Zealand mental health nursing: One team's perspective."* Practice Development in Health Care, Vol. 6, 137-142. Wiley.

(Ottmann and Tomek 1998) Ottmann, T., I. Tomek (1998) *"Proceedings of World conference on Educational Multimedia and Hypermedia & World Conference on Educational Telecommunications."* Conference on Educational Multimedia and Hypermedia & World Conference on Educational Telecommunications. AACE.

(Page and Ghiglione 2012) Page, J., E. Ghiglione (2012) *"What is LAMS?"* Available online at <http://wiki.lamsfoundation.org/display/lamsdocs/About+LAMS#AboutLAMS-whatIs>. Last Checked: 1 April 2012.

(Pagulayan, Keeker, et al. 2003) Pagulayan, R. J., K. Keeker, D. Wixon, R. Romero, T. Fuller (2003) *"User-centered design in games."* The human-computer interaction handbook. Edited by A. J. Julie and S. Andrew. Associates Inc.

(Park and Lee 2004) Park, O. C., J. Lee (2004) *"Adaptive Instructional Systems."* Handbook of research for education communications and technology. NJ: Erlbaum.

(Peirce, Conlan, et al. 2008) Peirce, N., O. Conlan, V. Wade (2008) *"Adaptive Educational Games: Providing Non-invasive Personalised Learning Experiences."* 2nd IEEE International Conference on Digital Games and Intelligent Toys Based Education, 17-19 Nov, Banff, Canada. IEEE.

(Piaget 1967) Piaget, J. (1967) *"Logique et Connaissance scientifique."* Encyclopédie de la Pléiade. Gallimard

(Pilkington and Parker-Jones 1996) Pilkington, R., C. Parker-Jones (1996) *"Interacting with computer-based simulation: The role of dialogue."* Computers & Education, Vol. 27 Issue 1. Science Direct.

(Pintrich 2000) Pintrich, P. (2000) *"The role of goal orientation in self-regulated learning."* Handbook of Self-Regulation. San Diego, CA: Academic.

(Pintrich and Schunk 1996) Pintrich, P., D. Schunk (1996) *"Motivation in Education: Theory, Research and Applications."* NJ: Prentice Hall Merrill.

(Power, Dagger et al. 2010) Power, G., D. Dagger, V. Wade (2010) *"Final Implementation of the Adaptive Simulation Authoring Tool."* GRAPPLE Deliverable D3.5c. Available online at <http://www.grapple-project.org/public-files/deliverables/D3.5c-WP3-AuthoringAdaptiveSimulations-v1.0.pdf>. Last Checked: 9 April 2013.

(PowerPoint 2011) PowerPoint (2011) *"Microsoft PowerPoint."* Available online at <http://office.microsoft.com/en-us/powerpoint/>. Last Checked: 1 April 2012.

(Preece 1994) Preece, J. (1994) *"Human-computer interaction."* Wokingham, Addison-Wesley.

(Prince and Davis 2008) Prince, R., H. Davis (2008) "The Scope of Adaptive Digital Games for Education." 2nd International Conference on Digital Games and Intelligent Toys Based Education, 17-19 November, Banff, Canada. IEEE.

(Psocka 1996) Psocka, J. (1996) *"Immersive training systems: Virtual reality and education and training."* Instructional Science, Vol. 23 Issue 5-6. Springer.

(Quinton 2006) Quinton, S. (2006) *"A brief critique on the future of learning (Assessing the potential for research)."* Contemporary Approaches to Research on Learning Environments. Worldviews, Singapore.

(Ramachandran, Sincoff et al. 2011) Ramachandran, S., E. Sincoff, M. Shuffler, C. Wiese, E. Salas (2011) *"An Examination of the Effectiveness of Simulations as Performance Assessment Instruments."* The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC), Florida, USA, December 2011. I/ITSEC

(Raptivity 2012) Raptivity (2012) Available online at <http://www.raptivity.com/>. Last Checked: 1 April 2012.

(Reams and Bashford 2011) Reams, S., C. Bashford (2011) *"Interdisciplinary Role Play: Nursing and Theater Students Advance Skills in Communication."* International Journal for Professional Educators: Competition and Competitiveness, Vol. 77, Issue 4. DKG.

(Reason 1990) Reason, J. (1990) *"Human error."* Cambridge University Press, New York.

(Reeves and Nass 1996) Reeves, B., C. Nass (1996) *"The Media Equation."* New York: Cambridge University Press.

(Reigeluth and Schwartz 1989) Reigeluth, C., E. Schwartz (1989) *"An instructional theory for the design of computer-based simulations."* Journal of Computer-Based Instruction, Vol. 16 Issue 1. ACM.

(Reiser 1987) Reiser, R. (1987) *"Instructional Technology: A History."* Instructional Technology: Foundations. Edited by R. Gagne. Hillsdale, NJ: Lawrence Erlbaum Associates.

(Rieber 1996) Rieber, L. P. (1996) *"Seriously considering play: Designing interactive learning environments based on the blending of microworlds, simulations, and games."* Educational Technology Research and Development, Vol. 44. Springer.

(Rieman and Franzke 1995) Rieman, J., M. Franzke (1995) *"Usability evaluation with the cognitive walkthrough."* Conference Companion on Human factors in Computing Systems, May 7-11, Denver, Colorado. ACM.

(Rocha, Araujo, et al. 2010) Rocha, R., R. Araujo, M. Campos, A. Boukerche (2010) *"Understanding and Building Interoperable, Integrable and Composable Distributed Training Simulations"* IEEE/ACM 14th International Symposium on Distributed Simulation and Real Time Applications, Oct 17-20, Virginia, USA, 2010. IEEE/ACM.

(Sampson and Karagiannidis 2002) Sampson, D., C. Karagiannidis (2002) *"Personalised learning: Educational, technical and standardisation perspective."* Interactive Educational Multimedia, Vol. 4. AACE.

(Schuman 1996) Schuman, L. (1996) *"Perspectives on Instruction: Behaviorism, Cognitivism, and Constructivism."* Sage.

(Shang, Shi, et al. 2001) Shang, Y., H. Shi, S. Chen (2001) *"An Intelligent Distributed Environment for Active Learning."* International World Wide Web Conference, May 1-5, Hong Kong. ACM.

(Sharples and Beale 2003) Sharples, M., R. Beale (2003) *"A technical review of mobile computational devices."* Journal of Computer Assisted Learning Vol. 19. JCAL.

(Sheinberg 2001) Sheinberg, M. (2001) *"Know Thy Learner: The Importance of Context in E-Learning Design."* Web-Page: Learning Circuits. Available online at <http://ipislam.edu.my/kplir/Bacaan/Artikel/Learning%20Circuits%20--%20ASTD%27s%20Online%20Magazine%20All%20About%20E-Lear.htm>. Last Checked: 1 April 2012.

(Shneiderman 1992) Shneiderman, B. (1992) *"Designing the user interface : strategies for effective human-computer interaction."* Addison-Wesley.

(Shute and Psotka 1995) Shute, V., J. Psotka (1995) *"Intelligent Tutoring Systems: Past, Present and Future."* Handbook of Research on Educational Communications and Technology. Edited by D. Jonassen. Scholastic Publications.

(SkillSoft 2012) SkillSoft (2012) Available online at <http://www.skillsoft.com/>. Last Checked: 1 April 2012.

(Sleeman and Brown 1982) Sleeman, D., J. Brown (1982) *"Intelligent Tutoring Systems."* New York Academic Press.

(Snow and Lohman 1984) Snow, E. R., D. F. Lohman (1984) *"Toward a theory of cognitive aptitude for learning from instruction."* Journal of Educational Psychology, Vol. 76. APA Psyc NET.

(Stewart 2001) Stewart, J. (2001) *"Radical Constructivism in Biology and Cognitive Science."* Foundations of Science, Vol. 6 Issue 1. Springer.

(Suchman 1987) Suchman, L. A. (1987) *"Plans and situated actions."* Cambridge: Cambridge University Press.

(SuperKids 2010) SuperKids (2010) *"Future Lab: Circuits for Physical Science"* Review. Available online at <http://www.superkids.com/aweb/pages/reviews/science/10/circuits/merge.shtml>. Last Checked: 1 April 2012.

(Surjono and Maltby 2003) Surjono, H., J. Maltby (2003) *"Adaptive educational hypermedia based on multiple student characteristics."* 2nd International Conference on Web-based Learning, 18-20 August, Melbourne, Australia, Australia: LNCS-Springer.

(Sutton, Cole et al 1998) Sutton, S., R. Cole, J. de Villiers, J. Schalkwyk, P. Vermeulen, M. Macon, Y. Yan, E. Kaiser, B. Rundle, K. Shobaki, J. Hosom, A. Kain, J. Wouters, D. Massaro, M. Cohen (1998) *"Universal Speech Tools: The CSLU Toolkit"*. 5th International Conference on Spoken Language Processing, Sydney, Australia, November 1998. ICSLP.

(Sweat 2006) Sweat, P. (2006) *"The Importance of Artificial Intelligence for Naval Intelligence Training Simulations."* Thesis, Naval Postgraduate School. Monterey, California. Available online at <http://www.dtic.mil/dtic/tr/fulltext/u2/a457150.pdf>. Last Checked: 1 April 2012.

(Tanner, Stewart et al. 2012) Tanner, J., G. Stewart, M. Totaro, M. Hargrave (2012) *"Business Simulation Games: Effective Teaching Tools Or Window Dressing?"* American Journal Of Business Education, Vol. 5, Issue 2. The Clute Institute.

(Taras 2005) Taras, M. (2005) *"Assessment-Summative and formative-Some theoretical reflections."* British Journal of Educational Studies, Vol. 53. Wiley.

(Tochtermann, Westbomke, et al. 1999) Tochtermann, K., J. Westbomke, U. Will, J. Leggett, J. Haak (1999) *Returning to Our Diverse Roots*. 10th ACM Conference on Hypertext and hypermedia Hypertext, 21-25 February, Darmstadt, Germany. New York: ACM.

(Torrente, Moreno-Ger, et al. 2009) Torrente, J., P. Moreno-Ger, B. Fernández-Manjón, Á. Blanco (2009) *"Game-like Simulations for Online Adaptive Learning: A Case Study."* 4th International Conference on E-learning, Edutainment, August 9-11, Banff, Canada. Springer LNCS. Available online at http://www.e-ucm.es/drafts/e-UCM_draft_151.pdf. Last Checked: 1 April 2012.

(Towell, Shavlik, et al. 1990) Towell, G. G., J. W. Shavlik, M. O. Noordewier (1990) *"Refinement of approximately correct domain theories by knowledge-based neural networks."* 8th National Conference on Artificial Intelligence, 29 July - 3 August, Boston, MA. MIT Press.

(Towne 1995) Towne, D. M. (1995) *"Learning and instruction in simulation environments."* Englewood Cliffs, NJ: Educational Technology Publications.

(Trollip 1985) Trollip, A. (1985) *"Computer based instruction: Methods and development."* New York: Prentice Hall.

(Tyldesley 1988) Tyldesley, D. A. (1988) *"Employing Usability Engineering in the Development of Office Products."* The Computer Journal. Oxford Journals.

(Uibu and Kikas 2012) Uibu, K., E. Kikas (2012) *"Authoritative and authoritarian-inconsistent teachers' preferences for teaching methods and instructional goals."* International Journal of Primary, Elementary and Early Years Education. Taylor and Finch. Available online at:

<http://www.tandfonline.com/doi/abs/10.1080/03004279.2011.618808>. Last Checked: 1 April 2012

(Van Joolingen and De Jong 2003) Van Joolingen, W., T. De Jong (2003) *"Authoring Tools for Advanced Technology Learning Environments - Chapter One: SIMQUEST Authoring Educational Simulations"* Kluwer Academic Publishers

(Varsidais 2000) Varsidais, B. (2000) *"Constructivism versus objectivism: implications for interaction, course design, and evaluation in distance education."* International Journal of Educational Telecommunication, Vol. 4, Issue 1, 339-362. AACE.

(Vázquez-Salceda, Ceccaroni et al. 2009) Vázquez-Salceda, J., L. Ceccaroni, F. Dignum, W. Vasconcelos, J. Padget, S. Clarke, P. Sergeant, K. Nieuwenhuis (2009) *"Combining Organisational and Coordination Theory with Model Driven Approaches to develop Dynamic, Flexible, Distributed Business Systems."* 1st International Conference on Digital Businesses, London, UK, June 2009. Published in Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, Vol. 21. Springer.

(Visual Basic 2011) Visual Basic (2011) "*Microsoft Visual Basic.*" Available online at <http://www.microsoft.com/visualstudio/en-us/products/2010-editions/visual-basic-express>. Last Checked: 1 April 2012.

(Vygotsky 1978) Vygotsky, L. S. (1978) "*Mind in society: the development of higher psychological processes.*" Cambridge Harvard University Press.

(Wade 2009) Wade, V. (2009) "*Challenges for Multi Dimensional Personalised Web.*" Invited Keynote. 1st International Conference of User Modeling, Adaptation, and Personalization, Trento, Italy, June 2009. Abstract available online at <http://umap09.fbk.eu/wade>. Last Checked: 22 September 2011.

(Wade and Power 1998) Wade, V. P., C. Power (1998) "Evaluating the Design and Delivery of WWW Based Educational Environments and Courseware." 6th Annual Conference on the Teaching of Computing, Dublin, Ireland, August 1998, 243-248. ACM.

(Wade and Power 1998a) Wade, V. P., C. Power (1998) "*Network Based Delivery and Automated Management of Virtual University Courses.*" 10th World Conference on Educational Multimedia and Hypermedia & World Conference on Educational Telecommunications, Freiburg, Germany, 1421-1427. AACE.

(Walkington and Maull 2011) Walkington, C., K. Maull (2011) "*Exploring the assistance dilemma: The case of context personalization*" 33rd annual conference of the Cognitive Science Society, Boston, Massachusetts, July 20 - 23, 2011. Cognitive Science Society.

(Walsham 1993) Walsham, G. (1993) "*Interpreting Information Systems in Organizations.*" John Wiley & Sons, Inc.

(Widdison, Aikenhead et al. 1998) Widdison, R., M. Aikenhead, T. Allen (1998) "*Computer Simulation in Legal Education.*" International Journal of Law Information and Technology, Vol. 5, Issue 3, 279. Oxford Journals

(Winn and Synder 1996) Winn, W., D. Synder (1996) "*Cognitive perspectives in psychology. Handbook of research on educational communications and technology.*" New York: Simon & Schuster.

(Woolfolk 2007) Woolfolk, A. (2007) "*Educational Psychology.*" 2007 Pearson International.

(Woolrych and Cockton 2001) Woolrych, A., G. Cockton (2001) "*Why and When Five Test Users aren't Enough.*" 1st International Conference IHM-HCI, Lille, France, Vol. 2, 105-108. September 2001. Cépaduès éd.

(Yakimovicz and Murphy 1995) Yakimovicz, A., K. Murphy (1995) "*Constructivism and collaboration on the Internet: Case study of a graduate class experience.*" Computers & Education, Vol. 24, Issue 3, 203-209. Science Direct.

(Zunzarren and Rodriguez-Sedano 2011) Gregorio, M., A. Rodriguez-Sedano (2011) "Creating Scenarios and Guiding Learning in a Medical Simulation Center." Social Sciences & Humanities, Vol. 2, Issue 3, 316-320. Scirp.

Appendices

Appendix I – Soft Skill Simulations

Soft skill simulations (also known as instructional social simulations or dialectic based educational simulations) are generally used for teaching skills based on interpersonal relationships, in particular communication skills where the learner is taught through the process of interacting with simulated people and scenarios through dialogue presented in a rich media format. The learner takes on a particular role and interacts with the simulation, from which they learn by constructing their own understanding by means of interaction and control the dialogue with the choices they make. Soft skill simulations are typically used to teach communication skills within the domain of business, such as customer care, interviewing skills and sale process simulations. The knowledge models that soft skill simulations depend on are dialogue based, the visualisation of which within authoring becomes very important as it needs to capture and display the features and attributes of communication between two or more people.

There are many advantages of using soft skill simulations. From an educational point of view soft skill simulations are very effective. It has been argued that serious learning only occurs if it takes place in the social and physical context in which it is to be applied. Soft skill simulations also have advantages over the real world alternatives of either employing actors to take on the roles or allowing the learner to participate in real scenarios. These include:

Cost: Soft skill simulations can be reused so are a much cheaper option compared to that of constantly employing actors.

Convenience: Compared to the use of actors, soft skill simulations are far more convenient, especially if they are available online. They can also be easily saved for replay, removing the necessity of recording equipment.

Time: Time within a soft skill simulation can be decelerated which is not possible if training with the use of actors. This could for example allow the learner to actively reflect during their simulation.

Safety: Soft skill simulations are often used to replicate situations where if mistakes are made by the learner in the real world there would have an adverse affect for example, a trainee sales representative making a mistake with a customer which results in the loss of a valuable client.

The soft skill simulations are based on the dialogue that takes place within the simulation. This is the model that details all the possibilities that can occur in the simulation While there have been many different approaches to modelling human-to-human dialogue; such as Chat Circles and Comic Chat, a complex task in itself; ensuring pedagogic appropriateness adds further to the development complexities. Soft skill simulations not only have to be authentic simulations but they must support serious learning. The idea that learning in a highly interactive environment is not something appended to the side of the experience but something that flows implicitly within the experience.

To fully understand the experience of the learner within the simulation and hence the transfer of learning, an illustration of the interactions in such a simulation is now described. This is accomplished using the example of an online simulation provided by Experience Builder. It follows a format that is utilised by most simulations that are used to teach soft skills. This particular simulation is a coaching simulation, the goal of which is to teach managers how to communicate and relate better with their staff. The scenario is as follows with a screen shot of the simulation in Figure I.1.

The learner is to take on the role of a manager of a technical support centre. The manager's company has recently introduced a new training initiative which the manager has implemented in their department. One of the employees that works under the manager has had difficulty in adopting to the new initiative. This simulation reproduces a meeting with the employee to discover what the issues are.

The simulation starts by showing a video of the employee entering an office, sitting opposite the manager's desk and asking:

"Hi, what did you want to see me about?"

The learner has three options to choose from:

"Ask him if he has concerns about the new training initiative"

"Ask him why his enthusiasm for work has dropped"

"Explain you've noticed his dropping enthusiasm lately and ask him how it's going"

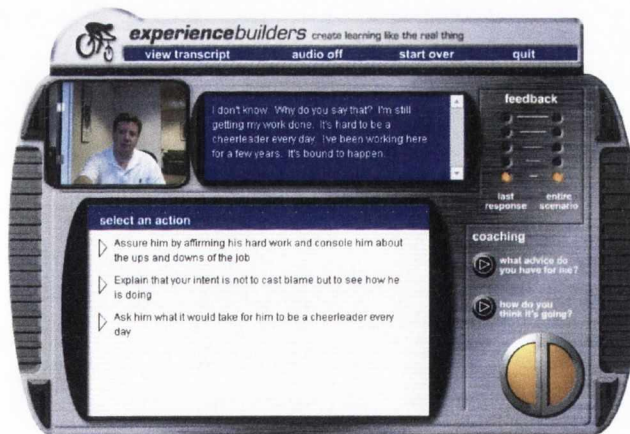


Figure I.1: Experience Builder soft skill simulation

By choosing the second option the simulation progresses another stage with another video clip being played. The employee's response to the learners question is:

"I don't know. Why do you say that? I'm still getting my work done It's hard to be a cheerleader every day. I've been working here for a few years. It's bound to happen"

The learner is then presented with another set of options to chose from:

“Assure him by affirming his hard work and console him about the ups and downs of the job”

“Explain that your intent is not to cast blame but to see how he is doing “

“Ask him what it would take for him to be a cheerleader every day”

The simulation continues in this format, with each step preceded by another video and set of options that the learner can choose from, the accumulation of which, creates a realistic and engaging simulation. The true value in using these types of simulations is that the learner wouldn't just use this simulation once but would work with it several times being able to try out different routes gaining more experience each time.

As mentioned soft skill online simulations typically take on this format of a combination of video followed by multiple choice options that the learner can choose from. There are however online rich media educational simulations that adopt a different approach. The simulation provided by SIVOX, for example, allows the learner to interact with the simulation using voice recognition as an alternative to multiple choices. While the soft skill simulations of Extempo Systems offer a free text input.

As previously mentioned, the key problem with incorporating any type of simulation is their composition and allowing non-technical domain experts to compose simulations. Generally, simulations are expensive and very time consuming to produce. This is particularly true where the simulation is designed to be an educational tool and is based on interpersonal relationships and role-play.

Appendix II – Simulation Learning Verbs

analyze
appraise
breakdown
calculate
categorize
classify
compare
contrast
differentiate
discover
discriminate
distinguish
examine
experiment
formulate
identify
illustrate
infer
interpret
locate
memorize
practice
recall
recognize
relate
review
solve

Appendix III – Annotated Java

Dialogue Element

```

package actsim;
import org.eclipse.emf.common.util.EList;
import org.eclipse.emf.ecore.EObject;

/**
 * @model
 */
public interface DialogueElement extends EObject {
    /**
     * @model
     */
    String getId();
    /**
     * @model
     */
    String getName();
    /**
     * @model
     */
    EList<String> getStatementURI();
    /**
     * @model
     */
    EList<String> getResponse();
    /**
     * @model
     */
    EList<String> getResponseURI();
    /**
     * @model
     */
    EList<String> getCategory();
    /**
     * @model
     */
    EList<String> getSubjects();
    /**
     * @model
     */
    EList<String> getRole();
    /**
     * @model
     */
    EList<String> getLearningOutcomes();
    /**
     * @model
     */
    EList<String> getAssessmentQandA();
    /**
     * @model
     */
    String getNotes();
    /**
     * @model
     */
    EList<String> getPreRequisites();
    /**
     * @model type="DialogueElement"
     */
    EList<DialogueElement> getDialogueElement();
    /**
     * @model type="Assessment" containment="true"
     */
    EList<Assessment> getAssessmentTrigger();
    /**
     * @model type="Feedback" containment="true"
     */
    EList<Feedback> getFeedbackTrigger();
    /**
     * @model type="Reflection" containment="true"
     */
    EList<Reflection> getReflectionTrigger();
}

```

Assessment Trigger

```

package actsim;
import org.eclipse.emf.common.util.EList;
import org.eclipse.emf.ecore.EObject;
/**
 * @model
 */
public interface Assessment extends EObject {
    /**
     * @model
     */
    String getName();
    /**
     * @model
     */
    String getAssessmentType();
    /**
     * @model
     */
    String getAssessmentTiming();
    /**
     * @model
     */
    String getAssessmentExecution();
    /**
     * @model
     */
    String getURI();
    /**
     * @model
     */
    EList<String> getLearningOutcomes();
    /**
     * @model
     */
    EList<String> getRole();
    /**
     * @model
     */
    EList<String> getSubjects();
    /**
     * @model
     */
    EList<String> getCategory();
}

```


Feedback Trigger

```

package actsim;
import org.eclipse.emf.common.util.EList;
import org.eclipse.emf.ecore.EObject;
/**
 * @model
 */
public interface Feedback extends EObject {
    /**
     * @model
     */
    String getName();
    /**
     * @model
     */
    EList<String> getRoles();
    /**
     * @model
     */
    EList<String> getSubjects();
    /**
     * @model
     */
    EList<String> getCategory();
    /**
     * @model
     */
    EList<String> getLearningOutcomes();
    /**
     * @model
     */
    String getDuration();
    /**
     * @model
     */
    String getProgress();
    /**
     * @model
     */
    String getAssessmentResults();
    /**
     * @model
     */
    String getPercentageComplete();
    /**
     * @model
     */
    String getNumberOfAttempts();
    /**
     * @model
     */
    String getRelatedMaterial();
}

```

Reflection Trigger

```

package actsim;
import org.eclipse.emf.common.util.EList;
import org.eclipse.emf.ecore.EObject;
/**
 * @model
 */
public interface Reflection extends EObject {
    /**
     * @model
     */
    String getName();
    /**
     * @model
     */
    EList<String> getLearningOutcomes();
    /**
     * @model
     */
    EList<String> getRole();
    /**
     * @model
     */
    EList<String> getSubjects();
    /**
     * @model
     */
    EList<String> getCategory();
}

```

Appendix IV – Component Interfaces

Learning Outcomes Nouns

Learning Outcome Nouns

the problem a patient presents with using appropriate interview techniques

the patients views on their diagnosis and possibility of having to take the patient about their diagnosis and check understanding.

the patients views on medication.

the patient about the medication including the pros and cons of taking the patients understanding of the medication and ascertain their preferences final decisions allowing the patient time to reflect over time.

alternative options with the patient.

about any practical problems the patient maybe experiencing.

adherence issues with the patient who has decided to take the medication patients understanding and educate re side effects of medication.

how ready the patient is to take medication.

how bad interview skills can alienate a patient.

the relevant medical history with the patient.

Name:

the problem a patient presents with using appropriate interview techniques

Description:

The subject describes the problem that a patient presents...

Update

Add

Delete

Clear

Save

Cancel

Roles

Roles

Doctor

Nurse

Care Giver

Name:

Nurse

Description:

A person formally educated and trained in the care of the sick or infirm. Compare nurse-midwife, nurse-practitioner, physician's assistant, practical nurse, registered nurse.

Update

Add

Delete

Clear

Save

Cancel

Subjects

Subjects

Greeting

No Category

Psychotic Symptoms - Delusions

Psychotic Symptoms - Hallucinations

Appetite/ Weight

Treatment/ Options

Mood Symptoms (Depression/Elation)

Energy/ Concentration

Interest/Enjoyment

Social history

Medical/ Medication History

Hopelessness and Suicide

Sleep

Mood Symptoms (Anxiety)

Mood Symptoms (Psychotic features)

Family History

Name:

Hopelessness and Suicide

Description:

The patient shows signs of hopelessness and suicide

Update

Add

Delete

Clear

Save

Cancel

Categories

Categories

Initial Statement

Orientation to/Permission for consultation

Open question

closed question

Clarifying / Checking Question

Invitation to discuss other options

Summarising Statement

Advice

Validation / Legitimation

Treatment plan

Advice and request to continue with interview

Name:

Orientation to/Permission for consultation

Description:

A meeting for deliberation, discussion, or decision

Update

Add

Delete

Clear

Save

Cancel

Appendix V – Usability Definition

There are many classifications of ‘usability’ within the HCI community. The International Organization of Standardization (ISO 2012) definition is however generally accepted as the most authoritative:

Usability is the effectiveness, efficiency and satisfaction with which specified users can achieve specified goals in particular environments.

- *Effectiveness is the accuracy and completeness with which users achieve specific goals.*
- *Efficiency is the accuracy and completeness of goals achieved in relation to resources expended.*
- *Satisfaction is the comfort and acceptability of using the system*

This interpretation of usability is incorporated throughout the evaluation sequences completed within this research. It is clear, concise and allows evaluations to be effectively divided into categories which closely examine the important and relevant aspects of a user interface. The incorporation of the ISO (ISO 2012) usability definition in this research is particularly evident in the usability evaluations that were completed.

Appendix VI – Ethical Approval

Overview

Ethical approval was required for a number of the evaluations completed as part of this research. To accommodate such approval and ensure that a high standard of ethics was maintained several principles were adhered. These include:

- All users (authors) were over 18.
- Questionnaires and tests were completed anonymously.
- All data associated with evaluations was stored and published anonymously.
- No video recording of users was conducted; any video screen capture completed during an evaluation was insured to be recorded anonymously.

Some simulations developed during this research were targeted at adolescent (under the age of 18). These simulations were not tested with such learner groups due to the ethical implications and were instead evaluated with the aid of expert opinion. Simulations developed for adults (over the age of 18) were evaluated with appropriate real world learners.

Where necessary the appropriate ethical approval was applied for from the School of Computer Science and Statistics in Trinity College Dublin and granted during the course of this research. A sample of the ethical approval application can be found in the following subsection.

Ethical Approval Application

Purpose of Project

Adaptive soft skill simulations are a very effective form of education but the key impediment with their wide usage is the cost and complexity of authoring them. While there has been a great deal of research in adaptive hypermedia composition it has primarily been focused on courseware and content presentation with little attention given to simulations. The adaptivity that does exist in soft skill simulations tends to be rather limited in the areas in which it is applied and is generally only used for adapting to the learning experience of the user. Existing adaptivity also tends to be inextensible, embedded and hard coded into the design of the simulation. Unlike other approaches this research incorporates adaptivity as a separate concern to the simulation models and content. This allows adaptivity to be expressed and manipulated by authors while the simulation is being composed. By separating adaptivity from the simulation models the adaptivity can be easily extended to areas other than just the learning experience of the user. This approach also allows non-technical domain experts to develop the simulations making educational simulation composition accessible to all. The objective of this research is to address these issues by developing authoring processes to compose adaptive soft skill simulations and develop a tool which supports these processes.

Project Methods and Measurements

Overview

- The participant will initially complete a questionnaire using their assigned anonymous username.
- The participant will then complete a short training session.

- To insure your proficiency using the composition tool the participant will be required to complete several short tasks.
- The participant will then be presented with a series of evaluation tasks to complete using the ACTSim composition tool.
- While the evaluation tasks are being completed a video screen capture will be running to record the participants actions (the image or voice of the participant will not be recorded allowing them to stay anonymous).
- While the evaluation tasks are being completed the participant will be observed and particular events will be noted.
- Once all the evaluation tasks have been completed the participant will be required to complete a short test using their assigned anonymous username.
- Once the test is completed the participant will be required to complete a second questionnaire using their assigned username.
- Finally, the participant will be debriefed.
- During the study the participant is free to ask any questions, these question will be noted under the participants anonymous name and will then be answered.

Evaluation Methods

Question Classification and Analysis

One of the goals of this research is that the software and procedures involved in authoring should be accessible to domain experts who do not possess any technical skills. Authors do not need to write code or create rules in their composition. All principals involved in authoring adaptive soft skill simulations are designed to be intuitive and easily comprehended by the authors. By analysing the frequency and type of questions that are being asked by the participant during the study, including the training period, it will help determine if the participants are finding the software and procedures intuitive and easy to comprehend.

Participant Testing

Once the authors have completed the evaluation tasks they will be asked to complete a short test without the aid of any documentation. This method of evaluation allows the measurement of memory retention which is used as another indicator of participant comprehension of the principals and involved.

Time Analysis

Composing adaptive hypermedia and educational simulations typically takes considerable time. A goal of this research is to allow to authors to compose their simulation models quickly and efficiently. In order to analyse the amount of time participants spend on different aspects of task completion (e.g. reading task, asking questions, completing actions, etc) it is necessary to note the times that participants begin and end individual subtasks. This is accomplished by observing the participants while they are completing the evaluation tasks and making note of the times.

Error Classification and Analysis

Video screen capture of the participants also allows the classification and analysis of errors that they have made while completing the evaluation tasks. A high frequency of errors indicates that the tasks are not being completed in an efficient manner. Categorising the errors will indicate ineffectiveness in the design of the composition tool. To insure the participants remain anonymous software is run during the study which only captures the actions on the screen. No video or audio recorder are used during the study.

Questionnaire

The study requires the participants to complete two questionnaires. The first of these questionnaires is completed at the very beginning of the evaluation and is used to gather information regarding the participants' background, past experience with eLearning composition tools, etc. The second questionnaire is completed at the end of the study and is used to gather information about the participants experience using the composition tool. This questionnaire is used to measure the effectiveness of the tool and the user satisfaction. Neither questionnaire will require personal details such name or academic institution/company.

Participants

The participants of this study will be male and female soft skill domain experts primarily based in academia.

- **Number of participants:** 5-10
- **Age:** 25+
- **Gender:** male and female
- **Exclusion/Inclusion criteria:** participants are required to be experts in a domain that involves teaching aspects of soft skills. They are also requires to have previous experience with the ACTSim composition tool and to have previously developed at least one non-adaptive title.
- **Recruitment methods:** participants will be/have been primarily recruited within the college. Recruitment has been based on mutual benefit where the participant will use their experience with ACTSim in their own research or to develop titles that will be beneficial to Trinity College.

Debriefing Arrangements

The participants are not misled in any way during the study. It will be verbally explained to them during the debriefing that their actions on the screen were recorded in order to analyse the errors that they may have made during the study. It will also be explained to them that the questions that asked will also be analysed.

Ethical Considerations

Physiological effects of the study (RSI, Photosensitive Epilepsy, fatigue, injury, etc.):

There should be no danger of RSI or fatigue as the participants will only be using the software for a relatively short time, roughly 30 minutes. Should a participant exhibit symptoms of RSI they will be asked to cease their participation and will be excluded from future experiments. Appropriate medical advice will be sought through the appropriate channels.

The training does include some video screen capture of the composition tool being used but this does not include strobe or flashing lights. Participants are informed and warned of this.

Personal Data Retention and Privacy (confidentiality, anonymity, data security, publication of results):

All questionnaires and tests are to be completed with an assigned anonymous name in order to protect the anonymity of the participants.

The retention of anonymised personal data within this study is necessary in order to analyse the results of the study based on age, gender, experience, and other personal traits. The personal data will be collected through user provided questionnaires, tests and through observed behaviour within the study.

In order to protect the privacy of the participants' personal data, each participant will be allocated a unique anonymous username for the duration of the study. Throughout the study each participant will only be identified by their anonymous username.

Further to this the following participant information will **not** be gathered at any point during the study:

- Name
- Address
- Telephone number, email address, IM/internet usernames, etc.

In order to facilitate follow-up interviews with the participants, a list of participant names and corresponding anonymous user names will be retained by the supervising teacher. At no point will this data be held by those conducting this study and its usage will be restricted to solely facilitating follow-up questionnaires.

During the course of this study, should a participant inappropriately divulge personal details within a questionnaire, this information will be anonymised and or discarded as appropriate.

Through the use of anonymous usernames the privacy of participants' personal data is protected. Further to this, suitable security measures will be undertaken to ensure the personal data remains secure, and is only used for the purpose of this study.

Relevant Legislation

Data Protection Acts 1988 and 2003:

In accordance with the Data Protection Acts 1998 and 2003 the personal data recorded and stored during this project shall:

- Be obtained solely for the project outlined above.
- Be protected against unauthorised access, alteration, or destruction through the use of appropriate security measures.
- Be only processed provided that written consent is provided by a participant or in the case of a minor, the consent of an authorised parent or guardian

The 'Right of Access' (Section 4) to the stored data does not apply in the case of this project due to the exclusion in Section 5(1)(h), which pertains to data stored for research purposes whereby any publication of the data is done in a manner that does not identify any data subjects.

Freedom of Information Act 1997:

Under this legislation participants have the following statutory rights:

- a legal right for each person to access information held by public bodies
- a legal right for each person to have official information relating to him/herself amended where it is incomplete, incorrect or misleading
- a legal right to obtain reasons for decisions affecting oneself.

The above statutory rights will be upheld within this project in as far as is practically and financially viable.

Appendix VII – Evaluation User Groups

Psychiatry: ACTSim was developed in cooperation with the Department of Psychiatry in Trinity College Dublin as part of the ADAPT project. This association provided users for evaluations throughout ACTSims development. The simulations composed were video based and were designed to simulate psychiatric interviews. The objective of these simulations was to allow medical students to practice clinical interviews. Simulations were very large, complex and highly interactive. Several titles were developed which focused on areas which included depression, mania and schizophrenia. During the course of the research two users from the Department of Psychiatry completed multiple evaluations. An additional four users from the department completed one of the more extensive evaluations. These users possessed a great deal of clinical experience and were involved in instructing third-level students and developing Trinity College Dublin's Department of Psychiatry curriculum.

Customer Care: Soft skills are an integral part of customer care within business and services industries. A very high percentage of soft skill simulations that are authored are employed within these domains. As part of this research customer care simulations were developed in cooperation with a very large well known software company. The simulations composed were audio based and were designed to simulate interactive telephone dialogue. The objective of these simulations was to allow customer care agents to converse with European sales customers by offering advice and answering queries. Three titles were developed in this domain during the course of the evaluations. They each examined a distinct area of sales and presented the learners with a different customer and set of challenges. Two users participated in several of the evaluations completed. The users employed in evaluations within this domain were experienced team leaders responsible for sales agents training and development.

Exam Stress: Formal examinations can be a very stressful time for students effecting their exam performance and mental health. Several initiatives have been developed within Trinity College Dublin to assist students in dealing with this stress. As part of one of these initiatives an informative soft skill simulation was developed using the ACTSim authoring tool. The objective of this simulation was to allow third-level students to query and converse with a simulated expert in the area of exam stress. This allowed students to learn about different aspects of exam stress such as how it can be dealt with and why it occurs. The development of the exam stress simulation allowed several evaluations to be completed. One user participated from this domain and completed several evaluations. The user involved in developing the simulation is an expert in the area of student counselling with extensive experience particularly in the area of exam stress.

Healthcare (Social Interaction for Adolescent with Autistic Spectrum Disorders): Intellectually challenged adolescents often find it difficult to interact with others in everyday social situations. Several soft skill simulations were developed using ACTSim that were designed to assist such adolescents with their social skills. The video based simulations were developed as part of a research project within Trinity College Dublin. The objective of these simulations was to teach mentally challenged children socially acceptable behaviour. This was to be achieved by introducing them into social situations that would typical encounter and supportively correct socially incorrect behaviour. The simulations would also act informatively as they introduce the adolescents to situations that they may not have

previously experienced. One user participated from this domain and completed several evaluations; the user had previously completed a Masters in this area of research.

Healthcare (Informative Adolescent Instruction): Invalid adolescents diagnosed with a serious ailment often find it difficult to communicate with their Doctor or healthcare professional. Young patients may have queries and questions that they would like to ask but feel overwhelmed or embarrassed. Informative soft skill simulations were composed to offer an alternative to real world interaction. The objective of the simulations was to allow adolescent patients to converse with a simulated virtual healthcare professional. The hypothesis is that a young patient would feel more comfortable discussing certain topics related to their illness in a simulated environment. Furthermore, health professionals, particularly specialists, may only have a very short time with their patient. In such a brief period questions may not occur to a young patient. Online soft skill simulations would offer such patients the opportunity to ask questions at their own convenience. The nature of interaction implemented by ACTSim soft skill simulations (multiple choice options) would also suggest areas of discussion that the children may not have thought of independently. One user participated from this domain and completed several evaluations. The user developed the simulations as part of a Masters in the area of Health Informatics.

Education (Renaissance Tutorial): The objective of the Renaissance Tutorial simulation was to allow third level learners to interact with an expert as if it were a one-on-one learning experience. The simulation composed, which focused on Renaissance art, was informative in nature but incorporated a fine level of granularity which created a highly interactive experience. The simulation developed was based on a previously developed eLearning course. The content of the course was reworked so as to suit an enquiry based teaching style. A single user participated from the education domain and they composed one simulation. The user who created the simulation is greatly experienced in the area of eLearning. She has worked in the Centre for Learning Technology (CAPSL 2011) in Trinity College Dublin and were also associated with the Irish Learning Technology Association and National Digital Learning Repository (NDLR 2012). The simulation was developed as part of the user's research into the viability of incorporating soft skill simulations into existing eLearning courses.

Appendix VIII – Storyboards

The purpose of this document is to illustrate the steps involved in building a dialogue within the ADAPT composition tool.

Learning Outcomes are predefined in a file for version 0.1. They are described using a limited vocabulary, reference JISC.

The application is divided into three sections, Learning Outcomes Section, Dialogue Space Section and Tool Box Section. See Figure VI.1.

The functions available in the Tool Box include:
Add Node, Add Edge, Zoom-In, Zoom-Out, Search, Sort, Encapsulate, Associate, Highlight, Build. See Figure VI.2.

The user starts with a blank document within a tab, where each tab holds an entire dialogue for a specific concept/scenario. Exemplified here are MANIA, SCHIZOPHERNIA and MEDICATION CONCORDANCE. See Figure VI.2.

The Learning Outcomes Section hold all learning outcomes to be achieved, learning outcomes for Mania can be seen in Figure VI.3.

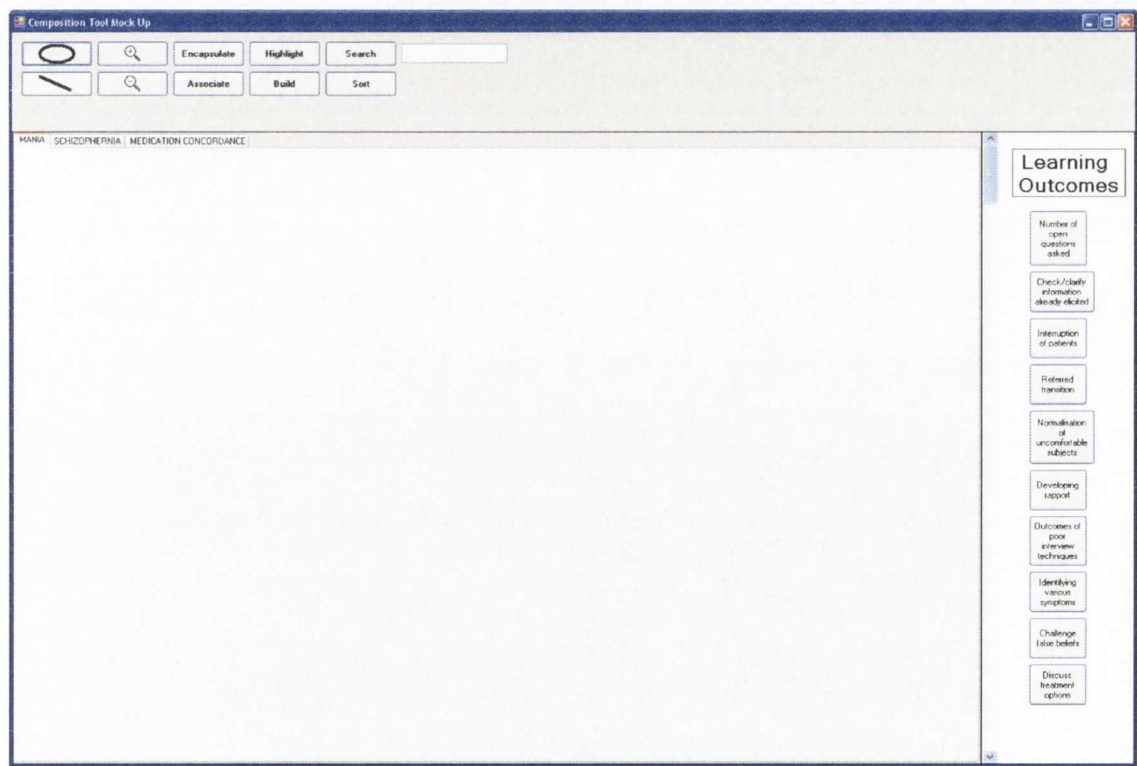


Figure VI.1 Composition Tool with Tab options

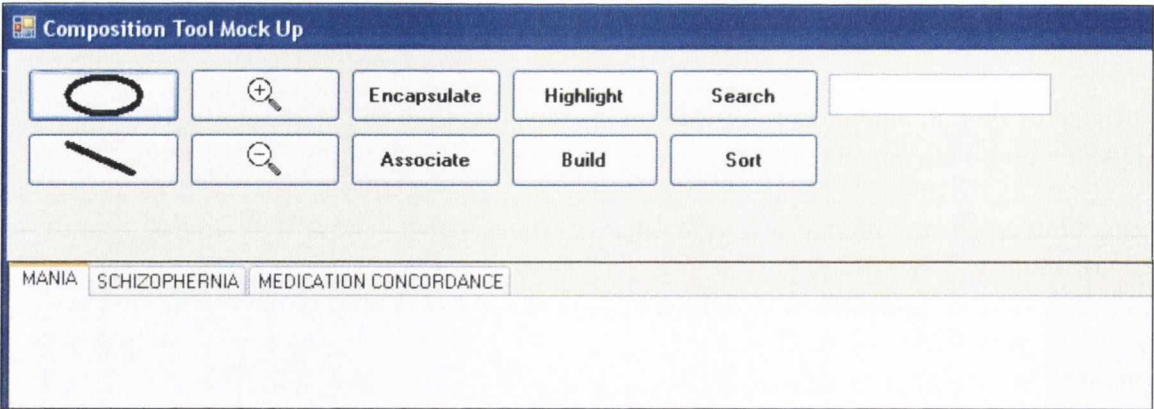


Figure VI.2 Tool Box Section

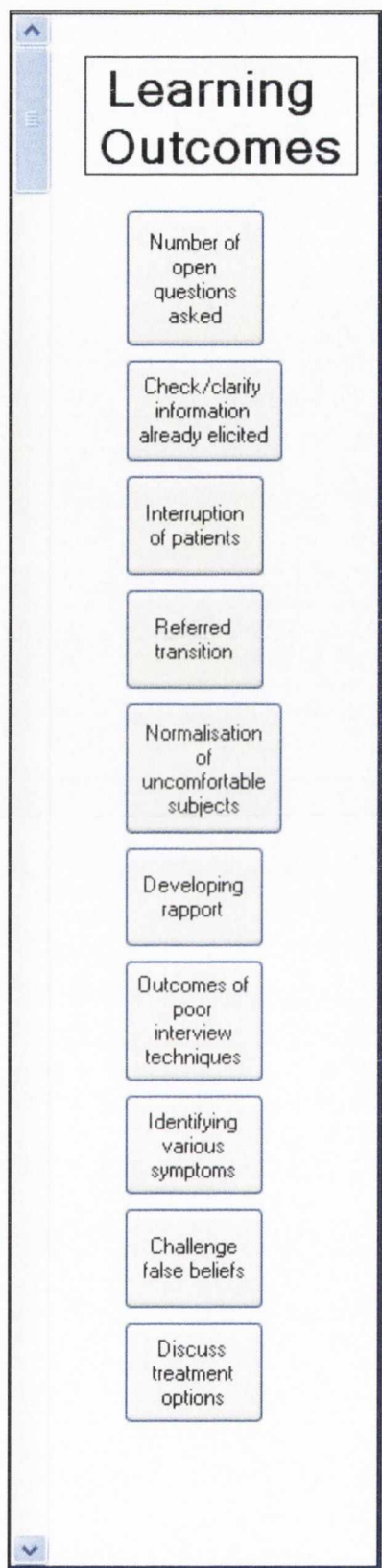


Figure VI.3 Learning Outcomes Section

The user can now begin to construct the dialogue; they do this by beginning with a ‘main stem’, which can be considered the best route through the dialogue. The user drags-and-drops nodes and edges in the appropriate places to build their dialogue graph. See Figure VI.4.

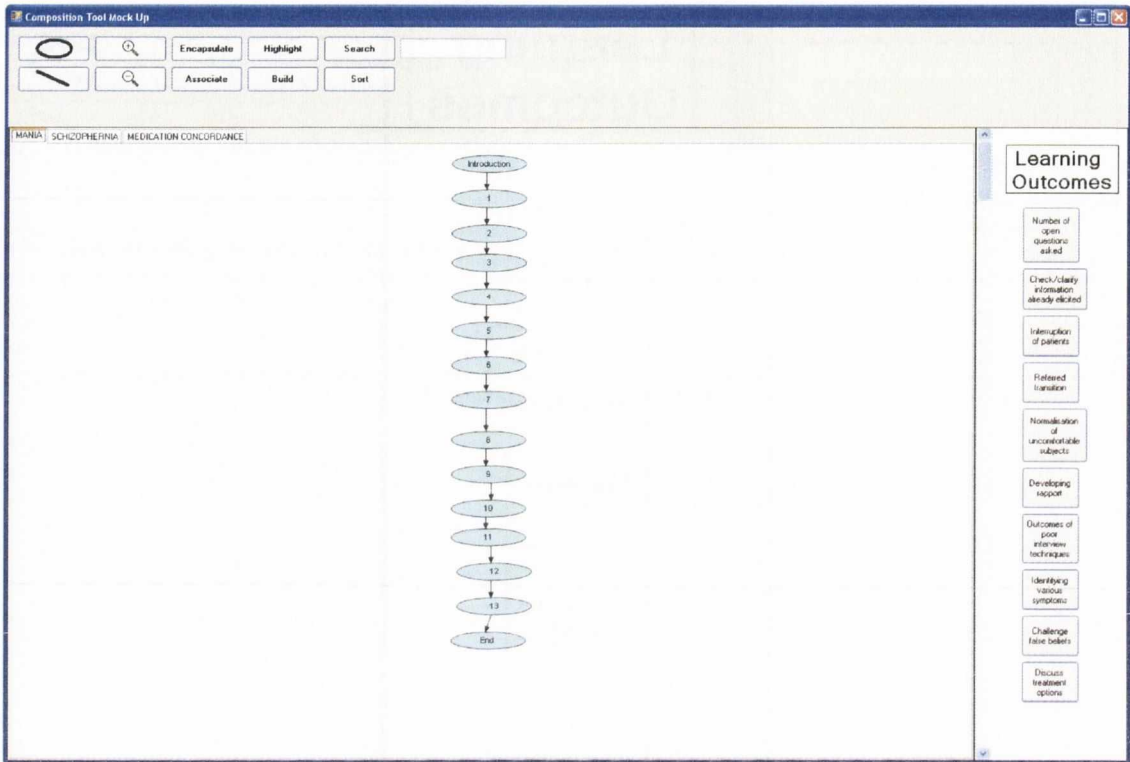


Figure VI.4 Dialogue Space with Main-Stem

Once the main-stem is completed the user can add sections of nodes that intuitively work towards certain learning outcomes. See Figure VI.5.

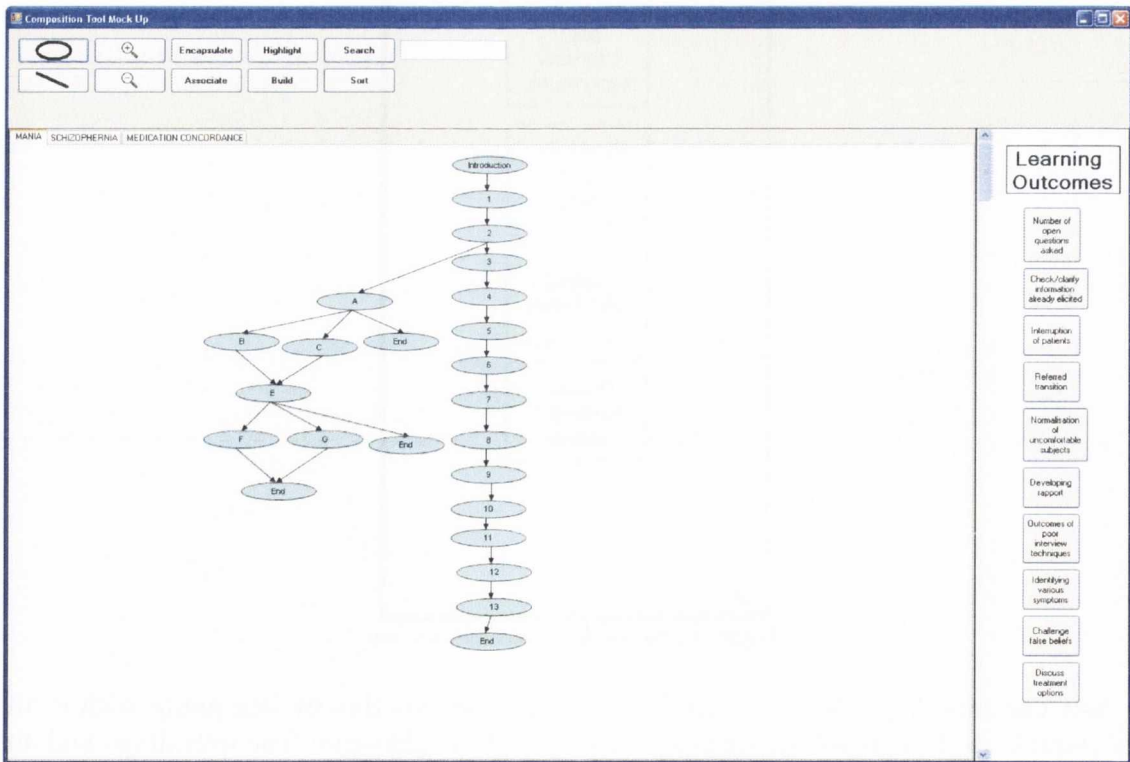


Figure VI.5 Main -Stem with added nodes

The dialogue space allows for navigation through the graph which inevitably will become very large and complex. The user can Zoom-in and Zoom-out of the space to allow them see the overall picture and specific parts, see Figure VI.6.

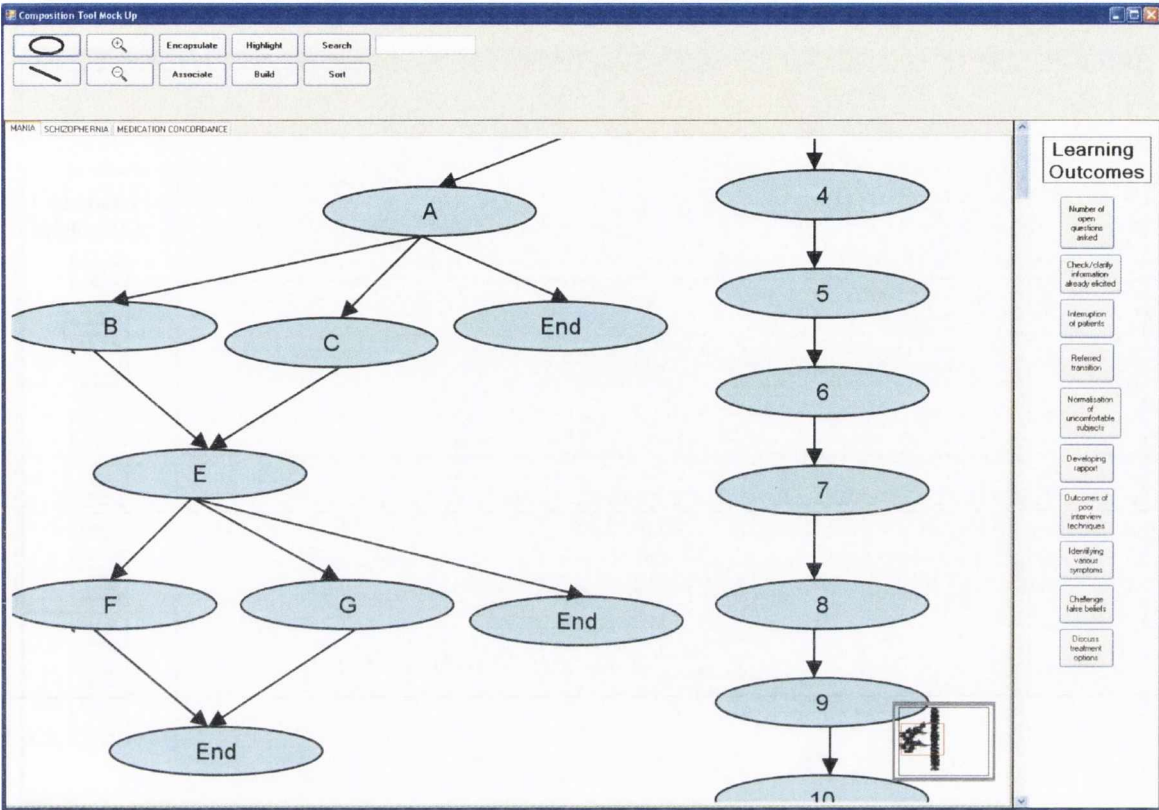


Figure VI.6 Use of the Zoom-In function with a map of the Dialogue Space in the bottom right corner
The composition tool also allows the user to highlight nodes, see Figure VI.7.

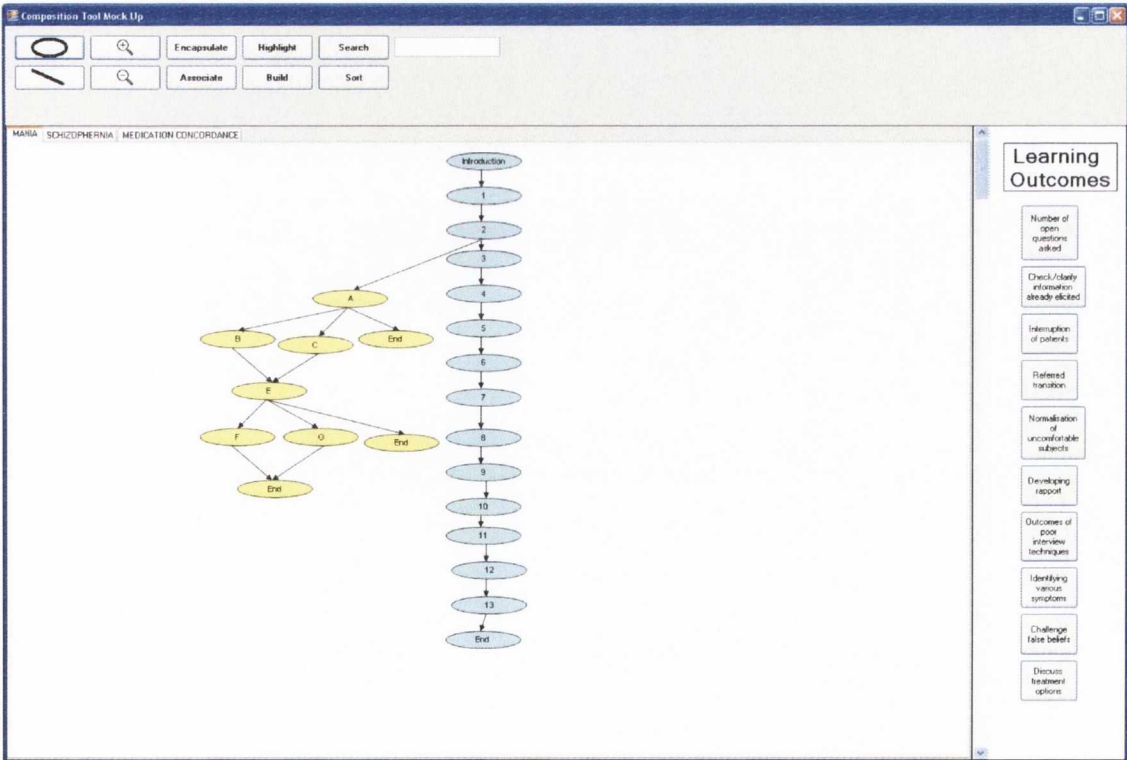


Figure VI.7 Highlighted Nodes

These nodes can then be encapsulated into a sub-dialogue. See Figure VI.8.

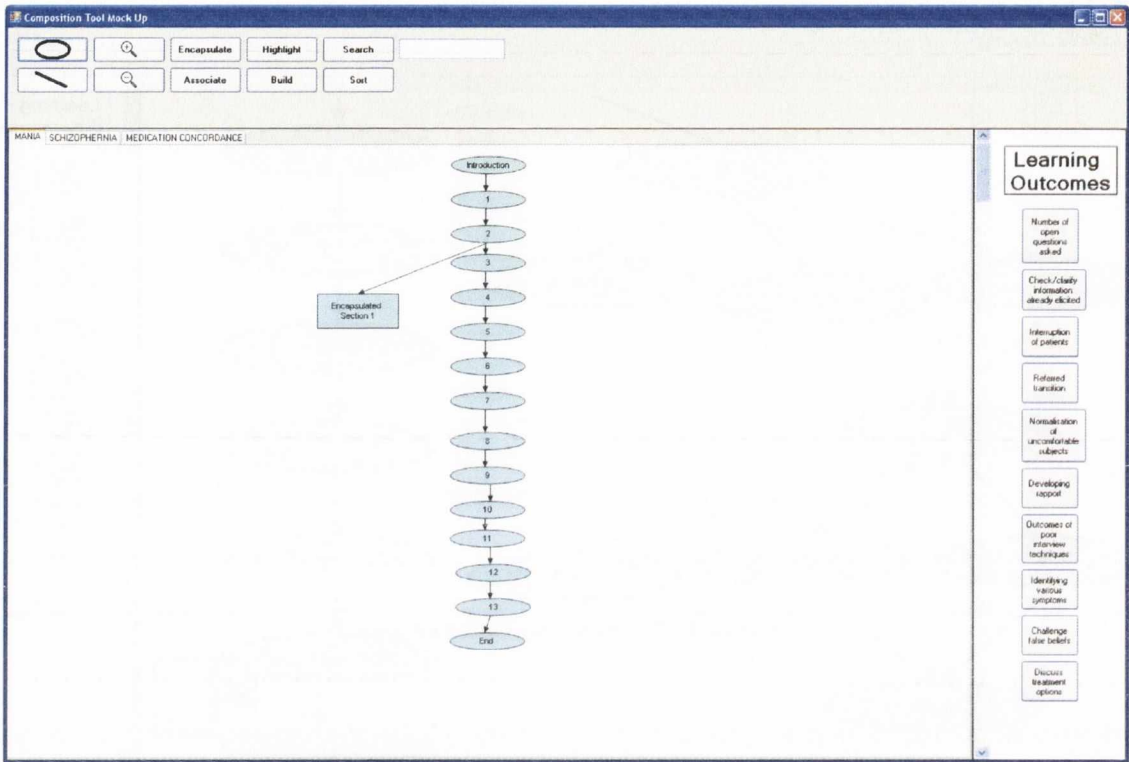


Figure VI.8 Highlighted Section that has been encapsulated

These steps can then be repeated so as to build a complete dialogue with multiple encapsulated sub-dialogues. See Figure VI.9.

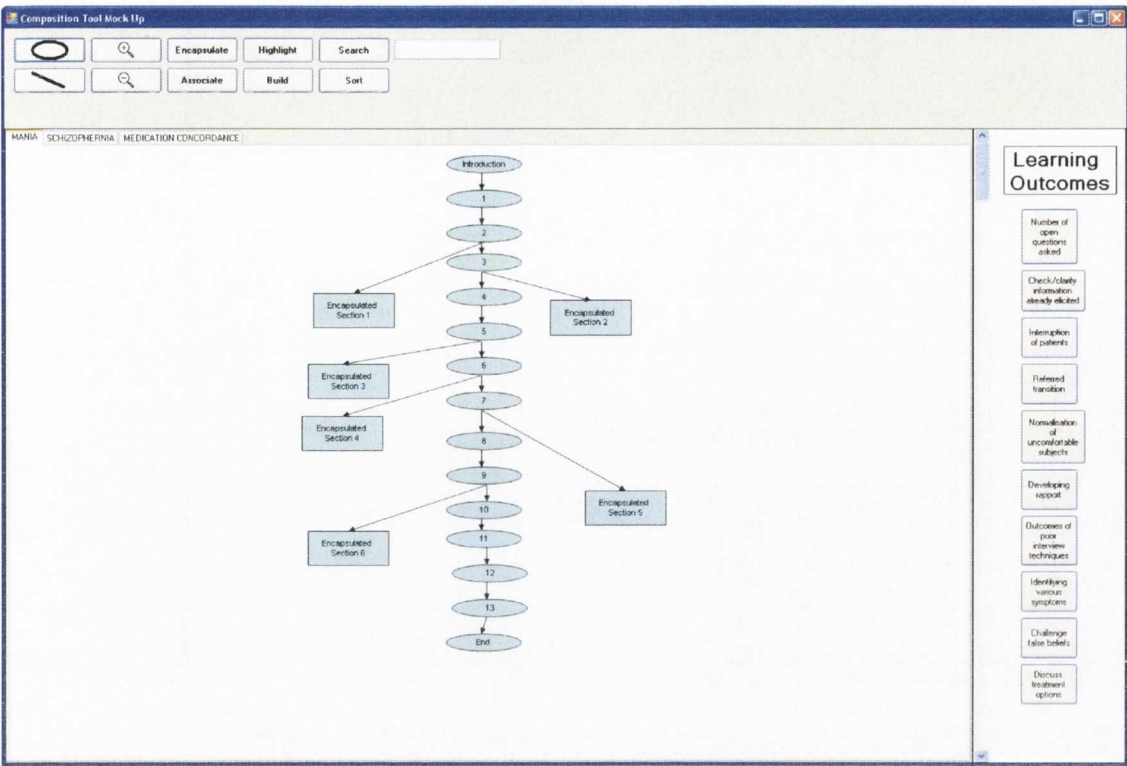


Figure VI.9 Dialogue Space with multiple sub-dialogues

For the user to access the sub-dialogues they simply double-click the encapsulation node to open a window which allows them to edit the sub-dialogue. See Figure VI.10 and VI.11.

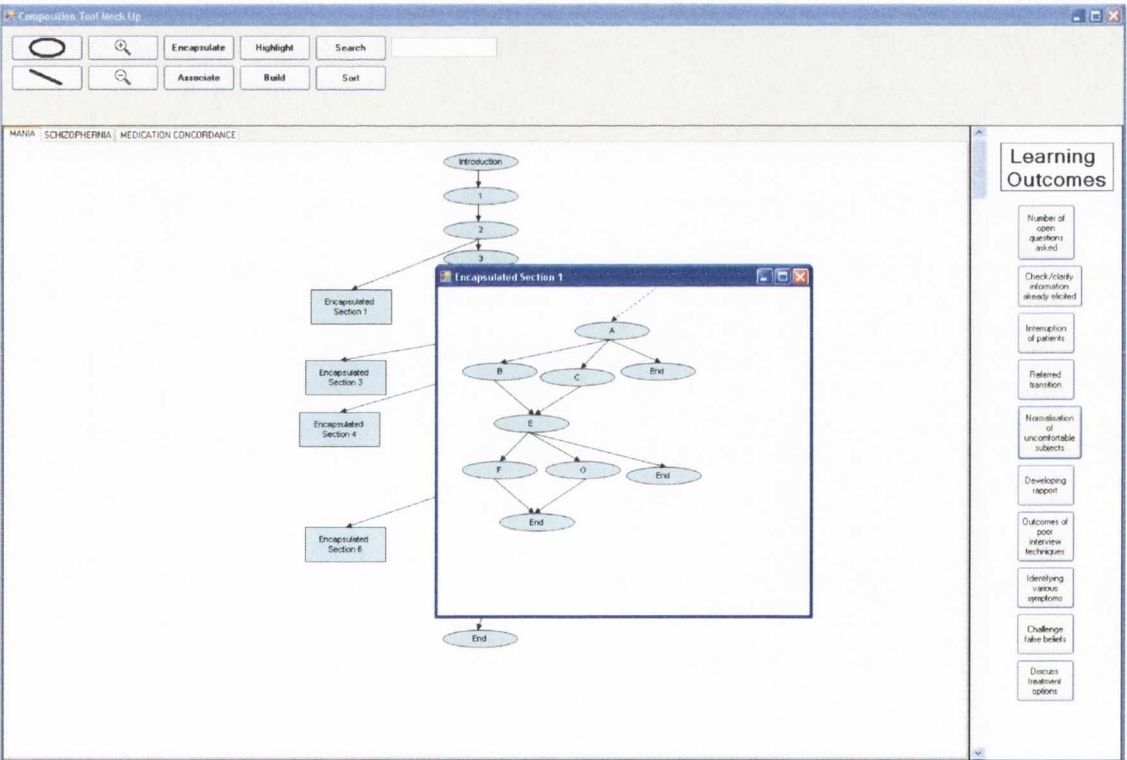


Figure VI.10 Accessing a Sub-Dialogue

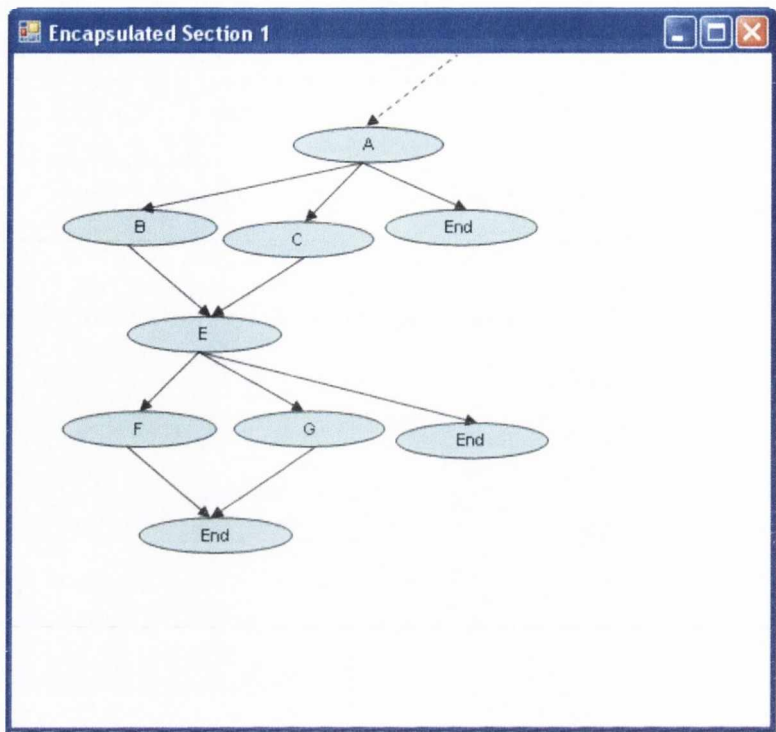


Figure VI.11 Encapsulated Sub-Dialogue

To insure pedagogically sound development the composition tool allows the user to associate the encapsulated sub-dialogues with the defined learning outcomes. This is done by first highlighting sub-dialogues to be associated, see Figure VI.12. The user then applies the associate function by pressing on the associate button and then the learning outcome they want the sub-dialogues associated with. In a similar way the user can associate a learning outcome with sub-dialogues.

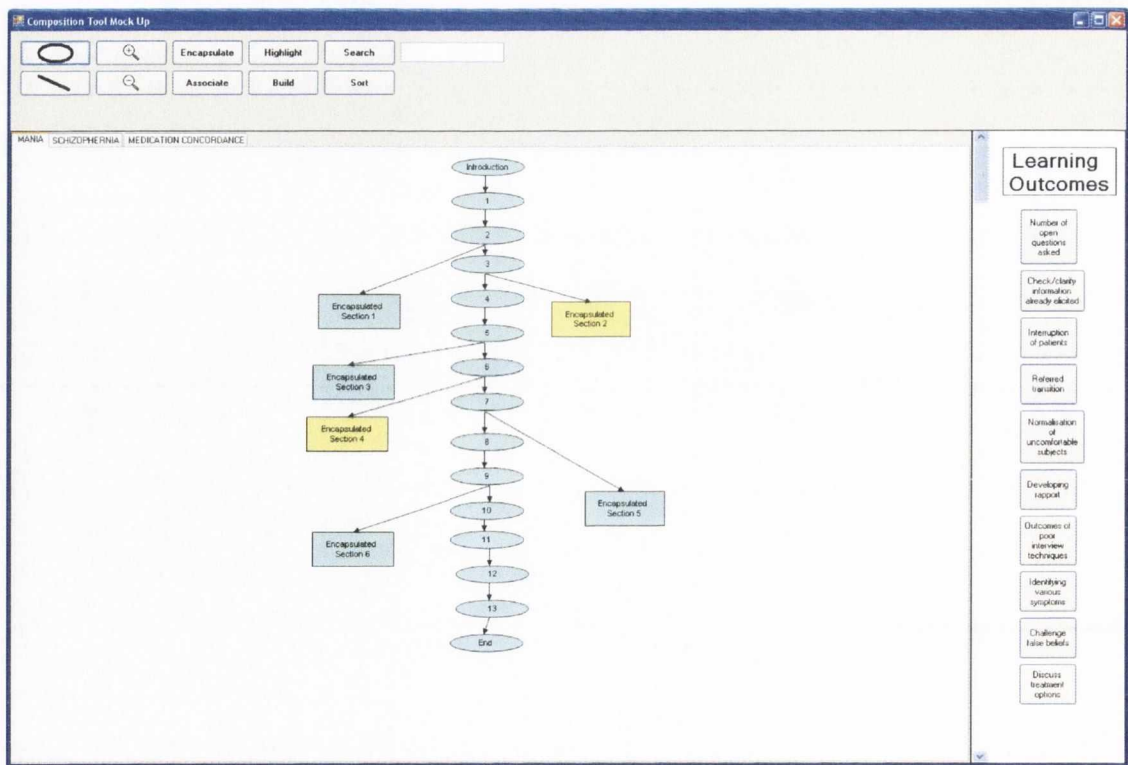


Figure VI.12 Highlighted Sub-Dialogues

The user can easily see which sub-dialogues have been associated with a learning outcome by clicking on the learning outcome icon which highlights the associated sub-dialogues in the dialogue space. See Figure VI.13.

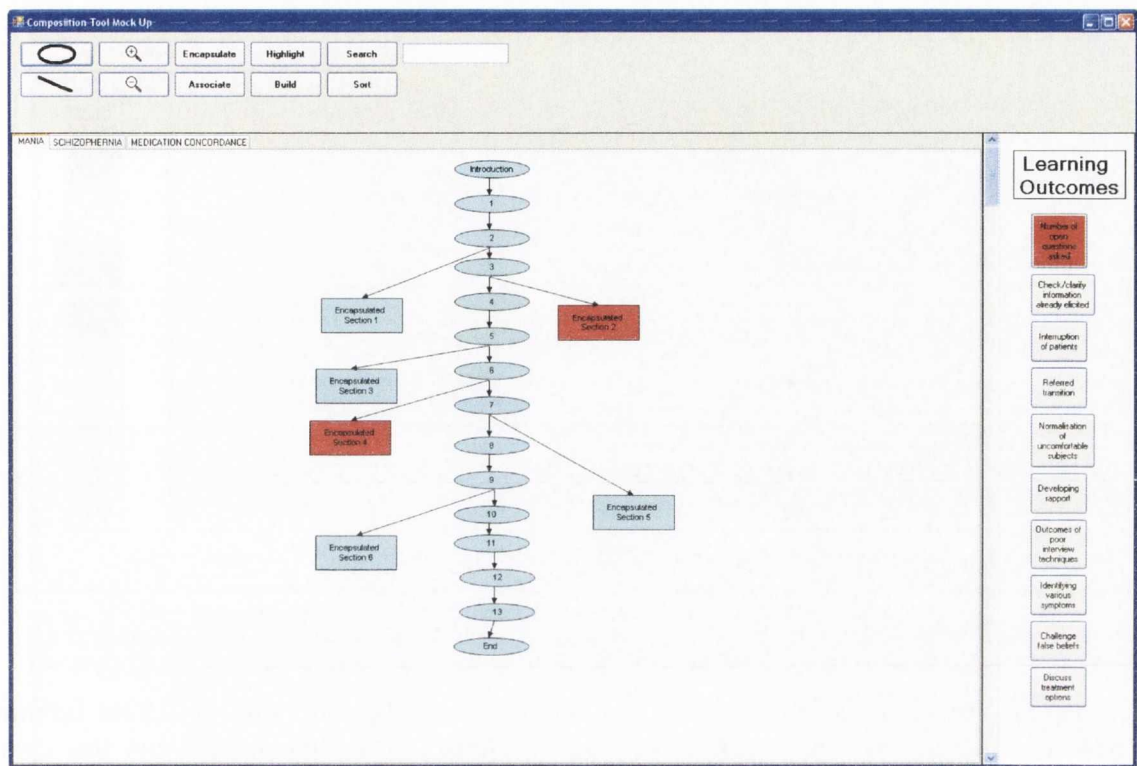


Figure VI.13: Sub-Dialogues associated with highlighted Learning Outcome

Similarly, if the user clicks on a sub-dialogue in the dialogue space it becomes highlighted and all associated learning outcomes in the Learning Outcomes Section also become highlighted. See Figure VI.14.

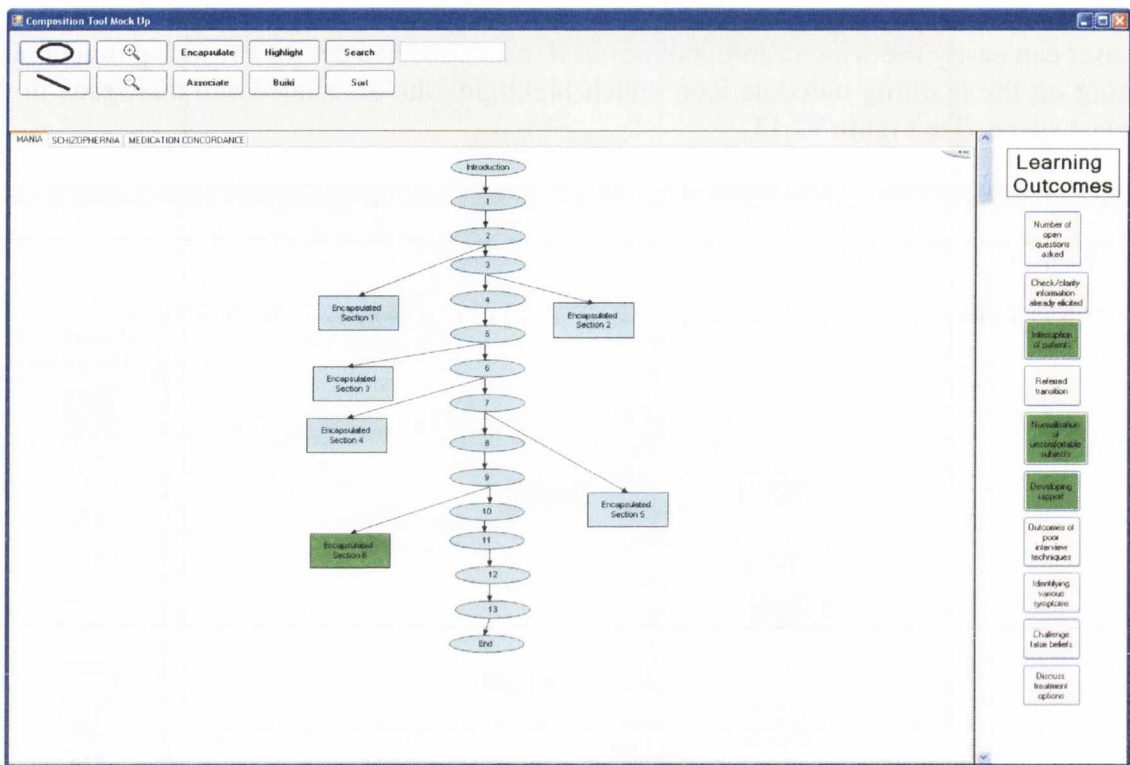


Figure VI.14: Learning Outcomes associated with highlighted Sub-Dialogue

For the user to view and edit the details of the learning outcomes the double click the learning outcome icon to access the Learning Outcomes Editing Box which contains the name, description and a list of associated sub-dialogues. See Figure VI.15 and VI.16.

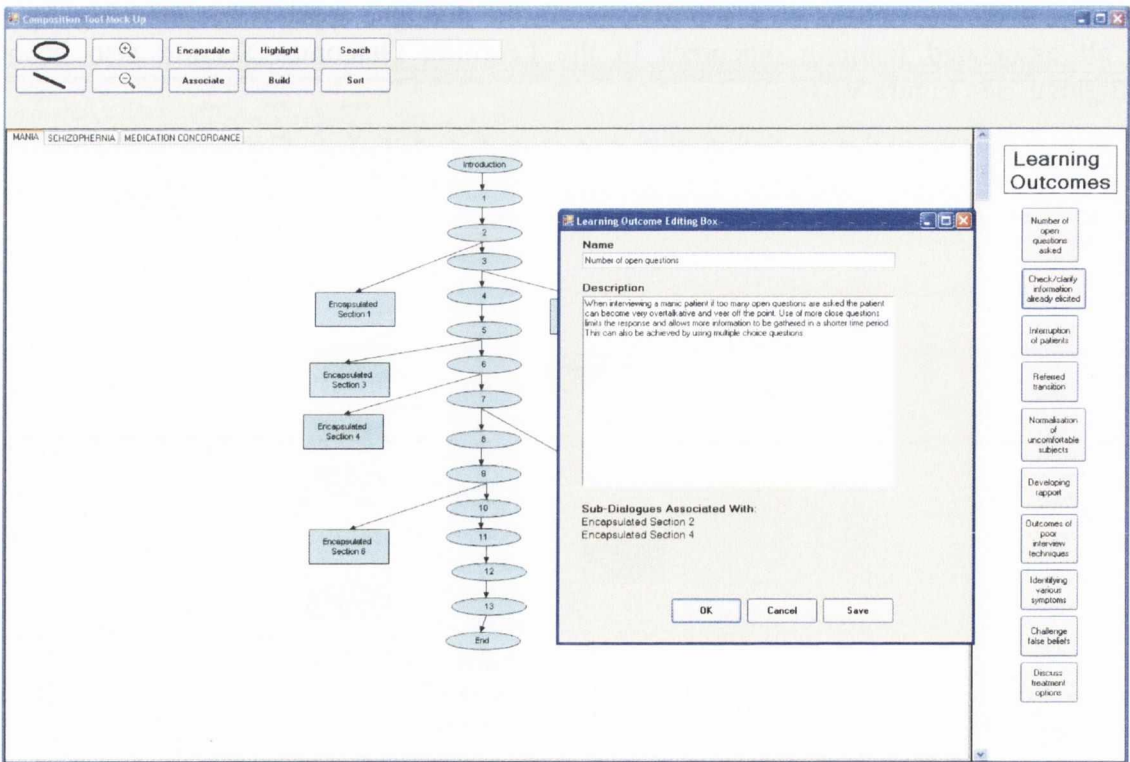


Figure VI.15 Learning Outcome Editing Box opened in the Composition Tool

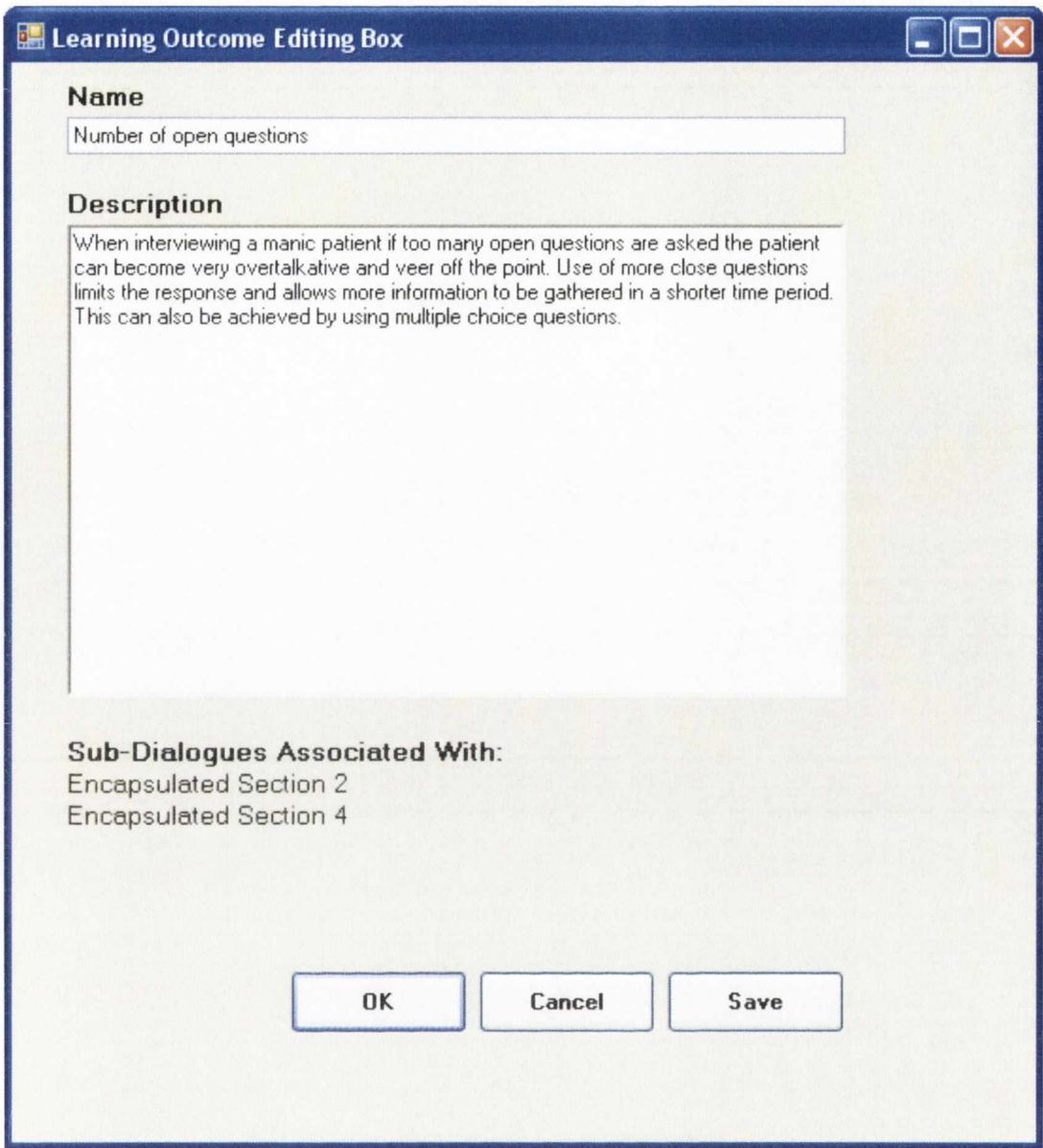
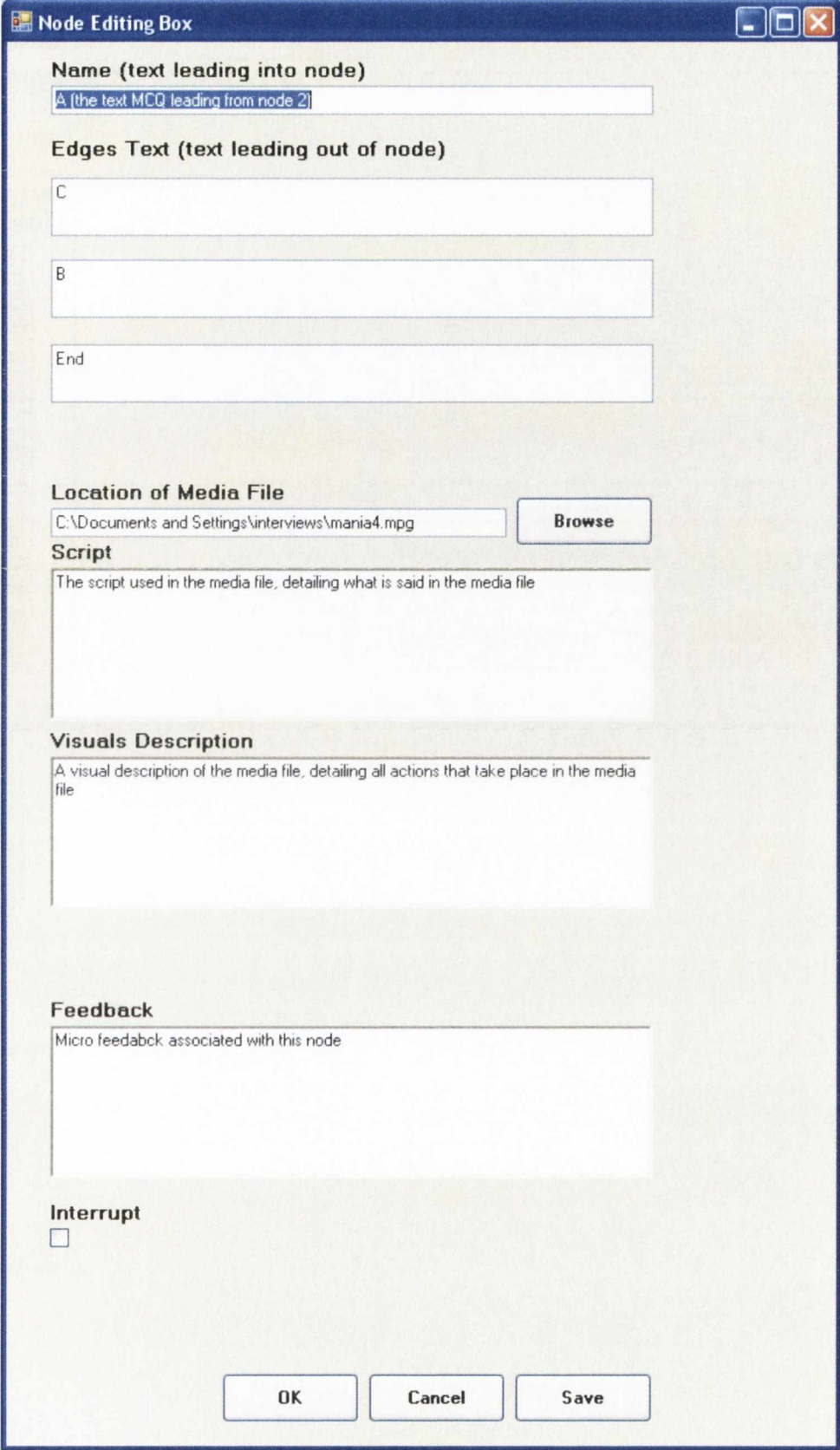


Figure VI.16 Learning Outcomes Editing Box

For the user to view and edit the details of an individual node they double click the node in the dialogue space to access the Node Editing Box which contains the name, edges text (text leading out of node), location of media file, script, visual description, feedback and Boolean interrupt. See Figure VI.17 and VI.18.



The image shows a Windows-style dialog box titled "Node Editing Box". It contains several sections for editing node details:

- Name (text leading into node)**: A text box containing "A (the text MCQ leading from node 2)".
- Edges Text (text leading out of node)**: Three stacked text boxes containing "C", "B", and "End".
- Location of Media File**: A text box containing "C:\Documents and Settings\interviews\mania4.mpg" and a "Browse" button.
- Script**: A text box with the placeholder text "The script used in the media file, detailing what is said in the media file".
- Visuals Description**: A text box with the placeholder text "A visual description of the media file, detailing all actions that take place in the media file".
- Feedback**: A text box with the placeholder text "Micro feedback associated with this node".
- Interrupt**: A checkbox that is currently unchecked.

At the bottom of the dialog box are three buttons: "OK", "Cancel", and "Save".

Figure VI.18 Node Editing Box

For the user to view and edit the details of a sub-dialogue they double click the encapsulation node in the dialogue space to access the Encapsulated Node Editing Box which contains the name, description, feedback and associated learning outcomes. See Figure VI.19 and VI.20.

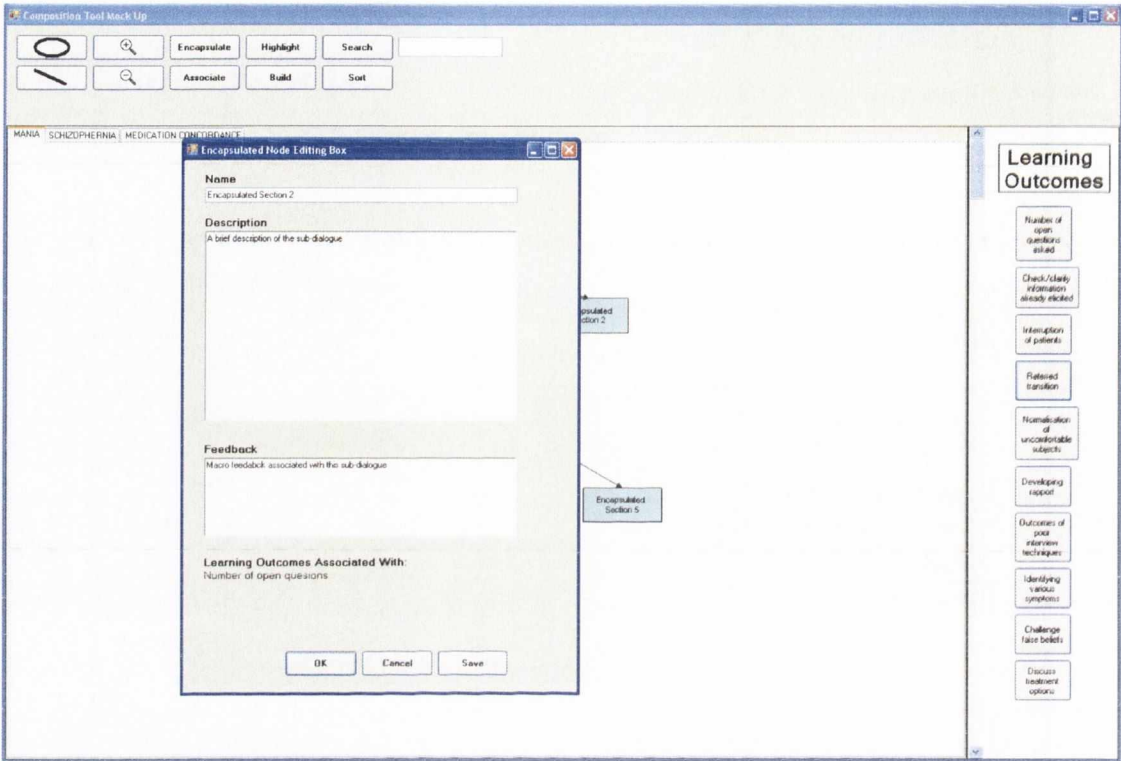


Figure VI.19 Encapsulated Node Editing Box opened in the Composition Tool

 Encapsulated Node Editing Box

Name

Encapsulated Section 2

Description

A brief description of the sub-dialogue

Feedback

Macro feedabck associated with this sub-dialogue

Learning Outcomes Associated With:

Number of open quesions

OK

Cancel

Save

Figure VI.20 Encapsulated Node Editing Box

Appendix IX – Composition Tool Sub-Dialogue Feature Comparison

Tabbed Sub-Dialogue v Compartment Sub-Dialogue

After being presented with two different approaches to graphically visualising sub-dialogues within a dialogue space, please answer the following questions by ticking the appropriate box for Tabbed or Compartment Sub-Dialogue:

1. In your opinion which of the two approaches is more complex to use?

Tabbed	
Compartment	

2. In your opinion which of the two is more intuitive in expressing sub-dialogues?

Tabbed	
Compartment	

3. In your opinion which of the two approaches do you think would have a better ability to display large dialogue spaces?

Tabbed	
Compartment	

4. In your opinion which of the two approaches is better at representing the ‘flow’ of the dialogue?

Tabbed	
Compartment	

5. In your opinion which of the two approaches is more user friendly?

Tabbed	
Compartment	

6. In your opinion which of the two approaches might cause information overload for the user?

Tabbed	
Compartment	

7. Additional comments on sub-dialogue feature:

Appendix X – First Usability Evaluation

Task

Evaluation Task – Psychological Interview

Initial Interview

Use the ADAPT Composition Tool to compose an initial interview which would take place between a Doctor and a patient.

The purpose of the interview is to teach a Doctor how to perform an initial interview with a patient.

Topics

Topics to be included in this interview should include but are not limited to questions about the patients:

- mother/father/family
- age
- employment
- relationship
- medical problems
- psychiatric problems
- medical history
- psychiatric history

The patient should have some psychiatric history.

The patient's family should include 4 siblings with questions about each ones:

- order in the family (eldest, youngest, etc)
- married or not
- employment

All other details are at your discretion.

The final interview should consist of about **40** Dialogue Elements

To design the interview:

- for each Dialogue Element the initial statement (Question) would be that of the Doctors with response that of the patient
- compose the main stem by connecting multiple Dialogue Elements
- add branches to this stem, these would normally be different topics you wish to include
- use the different features described in the users manual at least once (map, zoom, arrange, etc)

Please note roughly the amount of time you spend composing this interview (also the percentage of time spent thinking about the interview and percentage of time using the composition tool)

Questionnaire

Introduction

Description of package to be examined.

The ADAPT composition tool is used for creating dialectic based simulations. The present package allows the user to compose the dialogue models on which these simulations are to run. The package includes all standard operations for file functionality, a graphical display for modelling dialogue and tools to aid in the navigation of the dialogue space.

Sections

There are 3 different sections of questions:

- Creating and Manipulating Dialogue Elements
- File Operations and Navigating the Dialogue Space
- Representation and Overall Usability

Questions

There are three types of questions:

- **Yes/No:** Please indicate your answer yes or no by ticking the one of the boxes
- **Disagree/Somewhat Disagree/Somewhat Agree/Agree:** Please indicate a how strongly you disagree or agree with the associated statement by ticking one of the boxes
- **Comment Boxes:** Please enter any further comments you might have about the question

Section 1: Creating and Manipulating Dialogue Elements

1. Did you succeed in creating a dialogue element?

Yes No

Creating a dialogue element was an efficient process. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

2. Did you succeed accessing the dialogue element properties?

Yes No

Accessing the dialogue properties was an efficient process. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

3. Did you succeed updating the dialogue element properties?

Yes No

Updating the dialogue element properties was an efficient process. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

4. The window for the dialogue elements properties was easy to use. (Please tick one box)
 Disagree Somewhat Disagree Somewhat Agree Agree

--

5. Did you succeed moving the dialogue elements within the dialogue space?

Yes No

Was moving the dialogue elements an efficient process. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

6. Did you succeed in deleting dialogue elements from the dialogue space?

Yes No

Deleting dialogue elements was efficient. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

7. Did you succeed in connecting different dialogue elements?

Yes No

Connecting different dialogue elements was efficient. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

8. Did you succeed in creating the main stem?

Yes No

This was an efficient process. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

9. Did you succeed in augmenting the main stem by adding more branches?

Yes No

Adding more branches was an efficient process. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

10. Did you succeed moving multiple dialogue elements within the dialogue space?

Yes No

Moving multiple dialogue elements was efficient. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

11. Did you succeed deleting multiple dialogue elements within the dialogue space?

Yes No

Deleting multiple dialogue elements was efficient. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

12. Any further comments regarding the creation or manipulation of dialogue elements.

Section 2: File Operations and Navigating the Dialogue Space

13. Did you succeed opening new or existing dialogue space files

Yes No

Opening new or existing dialogue space files was efficient. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

14. Did you succeed saving existing dialogue space files?

Yes No

Saving files was an efficient process. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

15. Did you succeed in closing dialogue space files?

Yes No

Closing dialogue space files was an efficient process. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

16. Did you succeed in using the undo/redo feature?

Yes No

Using the undo/redo feature was an efficient process. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

17. Did you succeed changing focus between different windows (views) within the application.

Yes No

Changing focus between different windows was efficient. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

18. Did you succeed minimising/maximising windows (views) within the application?

Yes No

Minimising/maximising windows was efficient. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

19. Did you succeed in moving windows (views) within the application?

Yes No

Moving windows was efficient. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

20. Did you succeed in zooming in and out of the dialogue space?

Yes No

Zooming in and out was efficient. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

21. Did you succeed using the map to navigate through the dialogue space?

Yes No

Using the map was efficient. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

22. Did you like the map feature?

Yes No

--

23. Did you succeed in using the list of dialogue elements provided to aid navigation in the dialogue space?

Yes No

Using the list of dialogue elements was efficient. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

24. Did you use the arrange feature?

Yes No

Using the arrange feature efficient. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

25. Did you succeed using the note functionality provided?

Yes No

Using the note functionality provided was an efficient process. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

26. Did you succeed in using the question category/topics interface?

Yes No

Using the question category/topics interface was an efficient process. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

Section 3: Representation and Overall Usability

27. How long did it take to create the dialogue? _____

28. The use of nodes (dialogue elements) and edges (connections) is a good way of representing dialogue.

Disagree Somewhat Disagree Somewhat Agree Agree

29. You feel you could model the questions you wanted to.

Disagree Somewhat Disagree Somewhat Agree Agree

30. You feel you could connect the questions you wanted to.

Disagree Somewhat Disagree Somewhat Agree Agree

31. You spent more time thinking about using the tool than thinking about creating the dialogue.

Disagree Somewhat Disagree Somewhat Agree Agree

32. Using the dialogue space and palette is a good way to compose dialogue models.

Disagree Somewhat Disagree Somewhat Agree Agree

33. Overall you found the tool efficient for creating dialogue models.

Disagree Somewhat Disagree Somewhat Agree Agree

34. The ability to hide sections of the dialogue model would aid in the navigation of the dialogue space.

Disagree Somewhat Disagree Somewhat Agree Agree

35. The use of colours to group certain branches would aid in the navigation of the dialogue space. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

36. Overall you liked using the tool. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

37. Other Comments:

Appendix XI – Customer Care Directed Discussion

Telecommunication Simulation

Evaluation – Directed Discussion

Dialogue Representation

1. Did you think the use of nodes (dialogue elements) and edges (connections) was a good way of representing dialogue?
2. Did you feel you could model the questions you wanted to?
3. Did you spend more time thinking about creating the dialogue or more time thinking about using the tool?
4. Was using the dialogue space and palette a good way of composing dialogue models?
5. Overall did you find the tool efficient for creating dialogue models?
6. Any further comments regarding the creation or manipulation of dialogue elements.

Navigation

7. Did you like using the map feature?
8. Did you like using the list feature?
9. Did you like using the arrange feature?
10. Do you think the ability to hide sections of the dialogue model would aid in the navigation of the dialogue space?
11. Do you think the use of colours to group certain branches would aid in the navigation of the dialogue space?

Methodology

12. Do you think the main stem/branching approach to authoring the dialogue was useful?
13. How do you think the methodology should be incorporated into the composition tool?
14. Do you feel there is a need for validation or verification functionality in the composition tool?

Adaptivity

15. Was the adaptivity supplied appropriate?

16. Do you think the design of the Triggers as separate nodes and the use of Dialogue Element Compartments intuitive?
17. Did you think it was easy to understand their concept?
18. Was there any adaptivity that you thought of which was not included in the composition tool?

Overall Usability

19. Was there any functionality that you thought was particularly good?
20. Was there any functionality that you thought was particularly bad?
21. Was there any functionality that you thought was particularly missing?
22. Overall you liked using the tool.
23. Any further comments?

Appendix XII – Customer Care Short Questionnaire

Telecommunication Simulation

Evaluation – Questionnaire

Section 1: Standard Functionality

38. Did you succeed in opening new or existing dialogue space files?

Yes No

Opening new or existing dialogue space files was efficient. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

39. Did you succeed saving existing dialogue space files?

Yes No

Saving files was an efficient process. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

40. Did you succeed in closing dialogue space files?

Yes No

Closing dialogue space files was an efficient process. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

41. Did you succeed in using the undo/redo feature?

Yes No

Using the undo/redo feature was an efficient process. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

42. The window for the dialogue elements properties was easy to use. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

Section 2: Authoring Methodology

43. The use of nodes (dialogue elements) and edges (connections) is a good way of representing dialogue.

Disagree Somewhat Disagree Somewhat Agree Agree

44. Did you succeed in creating the main stem?

Yes No

This was an efficient process. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

45. Did you succeed in augmenting the main stem by adding more branches?

Yes No

Adding more branches was an efficient process. (Please tick one box)

Disagree Somewhat Disagree Somewhat Agree Agree

46. Did you succeed moving multiple dialogue elements within the dialogue space?
Yes No

Moving multiple dialogue elements was efficient. (Please tick one box)
Disagree Somewhat Disagree Somewhat Agree Agree

47. Did you succeed deleting multiple dialogue elements within the dialogue space?
Yes No

Deleting multiple dialogue elements was efficient. (Please tick one box)
Disagree Somewhat Disagree Somewhat Agree Agree

Section 3: Navigation

48. Did you succeed in zooming in and out of the dialogue space?
Yes No

Zooming in and out was efficient. (Please tick one box)
Disagree Somewhat Disagree Somewhat Agree Agree

49. Did you succeed using the map to navigate through the dialogue space?
Yes No

Using the map was efficient. (Please tick one box)
Disagree Somewhat Disagree Somewhat Agree Agree

50. Did you succeed in using the list of dialogue elements provided to aid navigation in the dialogue space?
Yes No

Using the list of dialogue elements was efficient. (Please tick one box)
Disagree Somewhat Disagree Somewhat Agree Agree

51. Did you use the arrange feature?
Yes No

Using the arrange feature efficient. (Please tick one box)
Disagree Somewhat Disagree Somewhat Agree Agree

Section 4: Triggers

52. Did you succeed in placing the triggers on the Dialogue Elements?
Yes No

Placing the triggers was efficient. (Please tick one box)
Disagree Somewhat Disagree Somewhat Agree Agree

53. Could you access the properties of the Triggers?
Yes No

Did you think accessing the Triggers properties was efficient. (Please tick one box)
Disagree Somewhat Disagree Somewhat Agree Agree

54. Did you succeed in moving the Triggers from one Dialogue Element to another?
Yes No

Moving the Triggers between different Dialogue Elements was efficient. (Please tick one box)

Disagree	Somewhat Disagree	Somewhat Agree	Agree
----------	-------------------	----------------	-------

Appendix XIII – Second Usability Evaluation

Design Document

2nd ACTSim Usability Evaluation

Introduction

This document presents a detailed account of the processes involved in developing a usability evaluation of the ACTSim composition tool. This is the second of two usability evaluations and is to be completed in the final stages of development as part of a user based experiment. The goal of this evaluation is to build on the results of the earlier evaluations and to gather comprehensive feedback regarding the usability of the ACTSim composition tool. Usability is defined as the effectiveness, efficiency and ease of use (user satisfaction) of the system. The system is examined with regard to the requirements that were identified during development.

The process for developing this evaluation closely follows standard practices incorporated in usability evaluations. Each requirement is examined separately under the following headings:

- **Goal:** an abstract view of the purpose of the requirement (what is to be achieved)
- **Evaluation Objectives:** concrete points describing the different components of the requirement
- **Methodology:** the methods used to examine the usability of the requirement. These are selected with respect to their suitability to the evaluation objectives described above
- **Task:** the evaluation task to be completed by the user which examines the evaluation objectives and is derived from the methodology
- **Plan:** an outline of the procedure to follow which examines the requirement
- **Training:** necessary training associated with the requirement that the user should complete
- **Evaluation:** the final details of the evaluation(s) to be used in examining the requirement, these are derived from the evaluation objectives, methodology and task

The following section briefly describes an overview of the users, software and typical environment in which the systems would be used. This is followed by a detailed account of the each of the requirements being examined in this usability evaluation.

Overview

Users: Users are typically educators or trainers who do not have a technical background but are experts in the area of soft skills. Users will be required to complete the associated evaluation tasks. Users will have completed a short training session and have access to relevant documentation.

System: ACTSim is composition tool that allows non-technical subject matter experts to author and publish adaptive soft skill simulations.

Environment The system will be used in a private secluded room which approximates a usability lab as best as possible; no outside noise or distractions. It can be assumed ACTSim will be installed on a computer (PC/laptop) with standard specs. No additional equipment is required although a mouse should be available.

Requirements

Dialogue representation; adaptivity representation and application; have been identified as the two most critical requirements⁷ within the development of the ACTSim composition tool. While other requirements are also important it should be noted dialogue representation and adaptivity are central to ACTSim and are examined more thoroughly than other requirements.

1. Dialogue Representation

1.1. Description

The models upon which soft skill simulations operate are based on the dialogue that occurs within the simulation. A soft skill authoring tool must represent this dialogue in a user friendly and intuitive manner. The dialogue is decomposed into components which represent small parts of the dialogue. These components are linked together to create the overall model of the dialogue. An author needs to be able to add the dialogue components, link them, access their properties and easily understand all the underlying principals.

1.2. Goal

The dialogue representation requirement is critical to the success of the ACTSim composition tool and. Several methodologies are to be employed to examine the usability of the dialogue representation. Aspects of dialogue representation have been successfully evaluated in previous evaluations, specifically user satisfaction and efficiency. The effectiveness of the dialogue representation does however require further evaluation. The goal of this evaluation is to examine how effective the authoring tool is at representing dialogue. Does the authoring tool decompose the dialogue into components that are easily managed? Is the granularity of the dialogue components suitable to the needs of the author and an effective way of representing a structured dialogue? Does the author understand the principals of linking the dialogue elements to create the dialogue model?

1.3. Evaluation Objectives

1. The author should understand how the dialogue relates to the final simulation.
2. The author should understand how the graph represents the dialogue.
3. Gauge how effective the author found the graph representation of the dialogue.
4. The author should understand what each node represents.
5. Gauge how effective the authors found the use of statement and response.
6. The author should understand that the directed graph edges represent the possible paths available to the learner.

⁷ A critical requirement can be defined by asking question 'could adaptive soft skill simulations still be created without this requirement?'

7. Gauge how effective the authors found the use of edges to represent the paths available to the learner.

1.4. Methodology

Due to the importance of dialogue representation within the composition tool there are two techniques employed within the evaluation of its effectiveness:

- **Questionnaire:** there are several objectives that require the opinion of the author to be analyzed. These opinions should be gathered with a questionnaire once the associated task has been completed. The evaluation metric is based on a Likert scale – a high percentage of authors should return favorable opinions on the effectiveness of the dialogue representation. The use of an open text sections as well as the Likert scale allows qualitative data to be collected as well as quantitative. If the authors return non-favorable results the qualitative will indicate why⁸.
- **Short Test:** the principals involved in the dialogue representation have been designed to be simple and straightforward. The dialogue representation should be easily understood by non technical people and should therefore be easy to learn. By testing the authors understanding and retention of the principals involved it will assists in indicating that they found this to be the case. The evaluation metric is based on the percentage of correct answers the authors give within the test.

Success criteria:

- **Questionnaire:** Authors should return favorable opinions (between 4 and 5 on the Likert scale) of the effectiveness of the approach to model representation. This success criteria is assigned a high percentage due to the importance of the effectiveness of the dialogue representation.
- **Short Test:** All authors should score 80% of the test successfully. This is a high percentage as the author's understanding of the model representation is critical.

1.5. Task

After appropriate training the authors are required to complete a short task consisting of adding, editing and manipulating Dialogue Elements to show a required level of proficiency. All authors will have previously created a Dialogue Model (minimum of 30 Dialogue Elements) but it may be some time since they engaged with their models so are required to spend 10-15 minutes familiarising themselves with their Dialogue Model.

1.6. Plan

⁸ The questionnaire requests authors to give reasons why they gave a negative response to a question - i.e. it is important to know why a user was not satisfied with an aspect of the tool. Collecting qualitative data for every question is not efficient as there will be a number of questions and participants will loose interest in the evaluation if they need to fill out every text box. Users can still add any comments anyway if they so desire

Appropriate training should be completed followed by 15 minutes to allow the authors to complete the task. Once all of the tasks have been completed authors should be tested and then asked to complete the questionnaire. The author should familiarise them selves with the model (iteratively) in the first iteration of the authoring process (this task is component of the authoring task). The model should then be checked in the second iteration of the authoring process.

1.7. Training

Recorded power point presentation should include:

- diagram explaining that the model represents dialogue
- diagram explaining what each node represents
- diagram explaining what each edge represents
- diagram of how the model relates to the final simulation

1.8. Evaluation

Test

The test should explain to authors that they are not being tested but the effectiveness of the system is being examined. It should be completed anonymously by the authors with no access to documentation. Test questions for dialogue representation are numerically associated with evaluation objectives above (this convention is used throughout this document):

4. The author should understand what each node represents.
What are the two main properties that each dialogue element contains?
Answer: Statement and Response
2. The author should understand that the directed graph edges represent the possible paths available to the learner.
What do the arrows that connect the dialogue elements represent?
Answer: Paths the learner can take within the simulation
6. The author should understand how the graph represents the dialogue.
What does the model within ACTSim represent?
Answer: The dialogue that occurs with in the soft skill simulation.
1. The author should understand how the dialogue relates to the final simulation.
How does this model relate to the simulations that are authored?
Answer: The simulations that are being authored are based on the dialogue that occurs within the simulation. The model represents this dialogue.

Questionnaire:

The questionnaire should be anonymously completed by the authors after the test previously described. Questionnaire questions for dialogue representation are numerically associated under the headings of the evaluation objectives above:

1, 2, 4, 6: These Evaluation Objectives examine how easy it was to learn the principals behind dialogue representation.

1, 2, 4, 6 Please select the most appropriate answer to the following statement:

"I felt it was easy to understand how the dialogue model represents the dialogue that occurs in the soft skill simulation"

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

3. Gauge how effective the author found the graph representation of the dialogue.

Please select the most appropriate answer to the following statement:

"The approach of breaking down the dialogue into smaller components and using arrows to connect them was easy to understand"

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

Please select the most appropriate answer to the following statement:

"The use of dialogue elements and arrows allowed me to create the dialogue that I wanted"

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

5. Gauge how effective the authors found the use of statement and response.

Please select the most appropriate answer to the following statement:
“Breaking the dialogue into components of ‘statement’ and ‘response’ was good way of encapsulating the components of the dialogue”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

Please select the most appropriate answer to the following statement:
“Authoring a dialogue with components that were comprised of ‘statement’ and ‘response’ was easy”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

7. Gauge how effective the authors found the use of edges to represent the paths available to the learner.

Please select the most appropriate answer to the following statement:
“The arrows that connect the dialogue elements easily allowed me to create all the paths that I wanted available to the learner”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

2. Adaptivity Application and Representation

2.1. Description

There are two types of adaptivity that can be specified in the ACTSim composition tool. The first of these is adapting the dialogue model (or content). This is accomplished by tagging the dialogue model with properties that are relevant to the learner. The model can then be adapted so only relevant parts of the model are accessible in the simulation. The second approach is to personalize triggers firing in the simulation. This is accomplished again by tagging the triggers so they only fire if they are relevant to the learner. In order for adaptivity to become accessible to non-technical subject matter experts authoring tools need to incorporate a graphical approach to expressing personalization.

2.2. Goal

The adaptivity requirement is critical to the success of the ACTSim composition tool and must be evaluated thoroughly. This is the first usability evaluation to thoroughly examine adaptivity application and representation. The effectiveness, efficiency and user satisfaction of how adaptivity is applied and represented in the composition tool are all examined. Can an author easily understand the principals involved in adapting a dialogue model for a learner? Can an author understand how these principals are implanted in the composition tool? Does the authoring tool represent adaptivity in a manner that is easy for the author to visualise, apply and manage? Can an author easily and efficiently tag the dialogue model for adaptivity? Can an author easily and efficiently add adaptive triggers to the dialogue model?

2.3. Evaluation Objectives

As there are two types of adaptivity their evaluation objectives are described under subheadings.

Effectiveness:

Tagging

1. The authors should understand how the dialogue (content) can be adapted (how different routes/branches would be available some learners but not to others)
2. The authors should understand how tagging allows adaptivity to occur in the dialogue.
3. The authors should understand how the tagging will effect the final simulation
4. Gauge how effective the author found the use of tagging to describe the adaptivity
5. Gauge how effective the author found the representation of the tagging in the dialogue model
6. Gauge how relevant the authors found the adaptivity properties (i.e. role, learning outcome, subject, etc.)

Triggers

7. The authors should understand what a trigger is
8. The authors should understand how the triggers in the dialogue model will effect the final simulation
9. The authors should understand how the triggers are adaptive
10. Gauge how effective the author thinks the use of triggers is in adapting the final simulation
11. Gauge how effective the author found the representation of the triggers in the Dialogue Model
12. Gauge how relevant the author found the triggers properties that are related adaptivity (i.e. role, learning outcome, subject, etc.)

Efficiency:

Tagging

13. The authors could efficiently create the tagging properties
14. The author could efficiently tag individual Dialogue Elements
15. The author could efficiently tag groups of Dialogue Elements
16. The author could efficiently locate Dialogue Elements with particular tags
17. The author could efficiently remove tags from the Dialogue Elements

Triggers

18. The authors could efficiently add the triggers to the Dialogue Elements
19. The authors could efficiently move the triggers in the Dialogue Space
20. The authors could efficiently delete the triggers from the Dialogue Elements
21. The authors could efficiently access the triggers properties

User Satisfaction:

Tagging

22. Gauge how easily and comfortable an experience the author found creating the tagging properties
23. Gauge how easily and comfortable an experience the author found tagging individual Dialogue Elements
24. Gauge how easily and comfortable an experience the author found tagging groups of Dialogue Elements
25. Gauge how easily and comfortable an experience the author found highlighting Dialogue Elements
26. Gauge how easily and comfortable an experience the author found deleting Dialogue Elements

Triggers

27. Gauge how easily and comfortable an experience the author found adding triggers
28. Gauge how easily and comfortable an experience the author found editing trigger properties
29. Gauge how easily and comfortable an experience the author found moving triggers
30. Gauge how easily and comfortable an experience the author found deleting triggers

2.4. Methodology

Due to the importance of adaptivity within the composition tool and the number of aspects that are being examined there are three techniques employed within the evaluation of its effectiveness, efficiency and user satisfaction:

- **Questionnaire:** there are several objectives that require the opinion of the author to be analyzed. These opinions should be gathered with a questionnaire after the associated task has been completed. The evaluation metric is based on a Likert scale – a high percentage of authors should return favorable opinions on the effectiveness and user satisfaction of the adaptivity. The questionnaire allows qualitative data to be collected as well as the quantitative. If the authors return non-favorable results the qualitative will indicate why.
- **Short Test:** the principals involved in adaptivity have been designed to be simple and straightforward. The adaptivity should be easily understood by non technical people and should therefore be easy to learn. By testing the authors understanding and retention of the principals involved it will indicate that they found this to be the case. The evaluation metric is based on the percentage of correct answers the authors give within the test.
- **Video Screen Capture Analysis:** an effective method of measuring the efficiency of the adaptivity features is to record the actions of the author and analyze the video. The evaluation metrics are based on the number of errors made by the authors and the amount of time the author takes in completing the individual parts of the task. This approach requires significant time for analysis and software for video capturing, e.g. Camtasia Studio.

Success criteria:

- **Questionnaire:** Authors should return favorable opinions (between 4 and 5 on the Likert scale) of the effectiveness of the approach to adaptivity. This success criteria is assigned a high percentage due to the importance of the effectiveness of the adaptivity.
- **Short Test:** Authors should achieve at last 80% in the associated part of this test. This is a very high percentage as the authors understanding of the model representation are critical.
- **Video Screen Capture Analysis:** All authors should master critical functionality but be allowed to make some errors when first attempting use; users should not consistently commit errors. All critical functionality should be completed in a good time; this time is relative to each function and should be compared to an average. Critical functionality includes tagging, highlighting Dialogue Elements and adding, editing Triggers.

2.5. Task

After appropriate training the authors should use the dialogue model that they have previously created and apply adaptivity, tagging and triggers. The author should complete each part of this task before moving onto the next⁹.

- a) Create values for one tagging property. Select from role, subject or category.
- b) Tag Dialogue Elements appropriately with tagging property values that you have created. Tag at least one third of your Dialogue Model. Use both the Dialogue Element popup window (accessed through double clicking) and the multiple selection tagging function at least once.
- c) Remove tags from a single Dialogue Element and a group of Dialogue Elements.
- d) Add three triggers to the Dialogue Model, an assessment trigger, reflection trigger and feedback trigger. Their properties should be updated appropriately.
- e) Delete one of the triggers you have added to the Dialogue Model.
- f) Move a trigger you have created to another Dialogue Element.

NOTE: the evaluation of the 'highlight' function, which is associated with adaptivity, has been placed with the second sub task as its evaluation requires a large fully tagged complex model.

NOTE: this task might be completed on second iteration of the authoring process.

NOTE: users may have already applied adaptivity to their models, if so all adaptivity should be removed from model.

2.6. Plan

Appropriate training should be completed followed by 10 minutes to allow the authors to complete the task. Once all tasks have been completed the authors should be tested and then asked to complete the associated questionnaire questions.

2.7. Training

Recorded power point presentation should include:

- diagram explaining how the dialogue will become adaptive
- diagram explaining how tagging works
- diagram explaining how tagging allows for adaptivity
- diagram explaining triggers
- diagram explaining how triggers effect the final simulation
- diagram explaining how triggers are adaptive
- diagram explaining highlighting

Training for the highlight function should also include the use of the logic functions available to the author, i.e. AND OR operations.

2.8. Evaluation

⁹ During the evaluation users are observed and time stamps of tasks segments are recorded for adaptivity task

Test

The test should explain to authors that they are not being tested but the effectiveness of the system is being examined. It should be completed by the authors with no access to documentation. Test questions for adaptivity are numerically associated with evaluation objectives above:

1. The authors should understand how the dialogue (content) can be adapted (how different routes/branches would be available some learners but not to others)
Briefly (one or two sentences) explain how the dialogue model can be made adaptive.
Answer: Only certain areas of the dialogue will be relevant to each individual learner. Adapting the dialogue means only the relevant areas of the dialogue will be used in the simulation.
2. The authors should understand how tagging allows adaptivity to occur in the dialogue.
Briefly (one or two sentences) explain how tagging allows the dialogue to become adaptive.
Answer: Tagging allows the author to contextualize the dialogue model, to place dialogue elements in categories that the author has created. These descriptors are then matched to the user so only the relevant parts are made available in the simulation.
3. The authors should understand how the tagging will effect the final simulation
Briefly (one or two sentences) explain how tagging the dialogue will effect the final soft skill simulation.
Answer: The simulations operate on the dialogue model that the author composes.
7. The authors should understand what a trigger is
What is a trigger?
Answer: A trigger is an event that occurs when a learner reaches a certain point within a simulation. The triggers are based on educational principals of assessment, feedback and reflection.
8. The authors should understand how the triggers in the dialogue model will effect the final simulation
Briefly (one or two sentences) explain how a trigger in the dialogue model will effect the final simulation.
Answer: If a learner reaches the part of the model in their simulation that has a trigger placed there the trigger may fire.
9. The authors should understand how the triggers are adaptive
Briefly (one or two sentences) explain how the triggers are adaptive.
Answer: A trigger will only fire if it is relevant to the learner.

Video Analysis:

Performance-based analysis (Preece, pg 620) of video: "Video should be captured of associated task being completed. Analysis can then be conducted of recordings. Clearly defined measures need to be stated. Measures could include frequency of correct task completion, task timing, use of commands, frequency of user errors and time taken up by

various cognitive activities, such as pausing within and between commands and reading or inspecting various areas of the screen display.”

NOTE: Due to ethical issues a screen capture video will be used

NOTE: The tagging task requires the author to tag their Dialogue Model based on one property.

The following describe the evaluation objectives which are to be examined in the video analysis:

1, 2, 3, 7, 8, 9

The effectiveness of the design may also be analysed by examining the video of the composition tool in use. By employing an error classification scheme it is possible to indicate an aspect of design that may need improvement. If an error is consistently being made users it may be that the design is ineffective.

13. The authors could efficiently create the tagging properties.

Measure the time taken to create each Tagging property value.

Calculate the average time to create a Tagging property value.

Measure the number and type of errors.

Time Success Criteria: 30-45 seconds

14. The author could efficiently tag individual Dialogue Elements

Measure time taken for Tagging each individual Dialogue Element.

Calculate the average time to Tag each individual Dialogue Element

Measure the number and type of errors.

Time Success Criteria: 10 seconds

15. The author could efficiently tag groups of Dialogue Elements

Measure time taken for Tagging each group of Dialogue Elements.

Calculate the average time to Tag each group of Dialogue Elements.

Measure the number and type of errors.

Time Success Criteria: 30-45 seconds

16. The author could efficiently locate Dialogue Elements with particular tags

Note: task for this evaluation objective is described in Navigational Aids

Measure time taken to locate a particular Dialogue Element with certain tagging.

Calculate the average time to locate a particular Dialogue Element.

Measure the number and type of errors.

Time Success Criteria: 30 seconds

17. The author could efficiently remove tags from the Dialogue Elements

Measure time taken for removing Tags from Dialogue Elements (single and multiple).

Calculate the average time to remove Tags from Dialogue Elements.

Measure the number and type of errors.

Time Success Criteria (single): 20 seconds

Time Success Criteria (multiple): 20 seconds

18. The authors could efficiently add the triggers to the Dialogue Elements
 Measure time taken to add each Trigger to a Dialogue Elements.
 Calculate the average time to add Triggers to a Dialogue Elements
 Measure the number and type of errors.
 Time Success Criteria: 60 seconds (all Trigger types)

19. The authors could efficiently move the triggers in the Dialogue Space
 Measure time taken to move each Trigger in the Dialogue Space.
 Calculate the average time to move Triggers in the Dialogue Space
 Measure the number and type of errors.
 Time Success Criteria: <10 seconds

20. The authors could efficiently delete the triggers from the Dialogue Elements
 Measure time taken to delete each Trigger from a Dialogue Element.
 Calculate the average time to delete Triggers from Dialogue Elements.
 Measure the number and type of errors.
 Time Success Criteria: <10 seconds

21. The authors could efficiently access the triggers properties
 Measure time taken to access each Trigger property.
 Calculate the average time to access Trigger properties.
 Measure the number and type of errors.
 Time Success Criteria: <1 second

Questionnaire:

The questionnaire should be completed by the authors after the test previously described. Questionnaire questions for adaptivity are numerically associated under the headings of the evaluation objectives above:

4. Gauge how effective the author found the use of tagging to describe the adaptivity

Please select the most appropriate answer to the following statement:

"I thought tagging the dialogue model was a good way for me to indicate where I wanted the adaptivity to be in the dialogue."

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

Please select the most appropriate answer to the following statement:
“It was easy to see how tagging would effect the final simulation.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

5. Gauge how effective the author found the representation of the tagging in the dialogue model

Please select the most appropriate answer to the following statement:
“I liked the fact that tagging could in the dialogue model could only be viewed with the highlight function.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

6. Gauge how relevant the author found the adaptivity properties (i.e. role, learning outcome, subject, etc.)

Please select the most appropriate answer to the following statement:
“I thought the properties available for tagging were suitable to the types of adaptivity that I wanted to apply.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

22. Gauge how easily and comfortable an experience the author found creating the tagging properties

Please select the most appropriate answer to the following statement:

“I found it easy to create the tagging properties used for adaptivity.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

Please select the most appropriate answer to the following statement:

“The popup window which displayed the adaptivity properties was intuitive and easy to follow.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

23. Gauge how easily and comfortable an experience the author found tagging individual Dialogue Elements

Please select the most appropriate answer to the following statement:

“I found it easy to tag a single Dialogue Element for adaptivity.”

- Strongly agree

- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

24. Gauge how easily and comfortable an experience the author found tagging groups of Dialogue Elements

Please select the most appropriate answer to the following statement:
“I found it easy to tag a group of Dialogue Elements for adaptivity.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

25. Gauge how easily and comfortable an experience the author found highlighting Dialogue Elements

Please select the most appropriate answer to the following statement:
“I found it easy to highlight Dialogue Elements that were tagged for adaptivity.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

Please select the most appropriate answer to the following statement:

"I found it easy to find the highlighted Dialogue Elements in the Dialogue Model."

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

26. Gauge how easily and comfortable an experience the author found deleting tags from the Dialogue Elements

Please select the most appropriate answer to the following statement:

"I found it easy to delete tags from Dialogue Elements."

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

10. Gauge how effective the author thinks the use of triggers is in adapting the final simulation

Please select the most appropriate answer to the following statement:

"I thought the use of adaptive triggers was a good way for me to indicate where I wanted personalised assessment, feedback and reflection to occur in the dialogue."

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

11. Gauge how effective the author found the representation of the triggers in the Dialogue Model

Please select the most appropriate answer to the following statement:
“I liked the way the triggers were represented in the Dialogue Model.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

12. Gauge how relevant the author found the triggers properties that are related adaptivity (i.e. role, learning outcome, subject, etc.)

Please select the most appropriate answer to the following statement:
“I thought the properties available to tag the triggers were suitable for how I wanted to personalize them.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

27. Gauge how easily and comfortable an experience the author found adding triggers

Please select the most appropriate answer to the following statement:

“I found it easy to add triggers to the Dialogue Elements.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

28. Gauge how easily and comfortable an experience the author editing trigger properties

Please select the most appropriate answer to the following statement:

“I found it easy to access the properties of the triggers.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

Please select the most appropriate answer to the following statement:

“The popup window which displayed the trigger properties was intuitive and easy to follow.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

29. Gauge how easily and comfortable an experience the author found moving triggers

Please select the most appropriate answer to the following statement:
“I found it easy to move the triggers to other Dialogue Element compartments.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

30. Gauge how easily and comfortable an experience the author found deleting triggers

Please select the most appropriate answer to the following statement:
“I found it easy to delete the triggers from the Dialogue Model.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

3. Authoring Process

3.1. Description

The authoring process is an outline or blueprint that an author follows while developing eLearning solutions. This is an aspect of composition that runs through out the development of the dialogue model and encompasses all the different requirements that ACTSim should include.

3.2. Goal

Aspects of authoring process have been successfully evaluated in previous evaluations, specifically the process involved in authoring the dialogue model; dialogue model authoring process (creating the main stem, adding branches, etc). Other components however have not been evaluated in a usability evaluation. They are examined with respect to effectiveness, efficiency and user satisfaction. The goal of this evaluation is to insure that the authoring process helps the author to compose a dialogue model in an efficient, efficient and user friendly manner. Can the author follow the principal of iteration? Does the author understand

each of the steps involved in the authoring process? Do they know how each step is helping them in creating a more effective adaptive soft skill simulation? Can the authoring process be easily followed?

3.3. Evaluation Objectives

Effectiveness:

1. Gauge how effective the author found the authoring process in aiding them to compose the dialogue model
2. Gauge how easy it was for the author to understand the structure of the authoring process

Efficiency:

3. Gauge how efficient the author found the authoring process was at allowing them to compose the dialogue model
4. Gauge how efficient the author found the iterative approach incorporated in the authoring process

User Satisfaction:

5. Gauge how comfortable an experience the author found using the authoring interface

3.4. Methodology

- **Questionnaire:** there are several objectives that require the opinion of the author to be analyzed. These opinions should be gathered with a questionnaire after the associated task has been completed. The evaluation metric is based on a Likert scale – a high percentage of authors should return favorable opinions on the effectiveness, efficiency and user satisfaction of the adaptivity. The questionnaire allows qualitative data to be collected as well as the quantitative. If the authors return non-favorable results the qualitative will indicate why.

Success criteria:

- **Questionnaire:** Authors should return favorable opinions (between 4 and 5 on the Likert scale) of the effectiveness of the approach to the authoring process. This success criteria is assigned a moderately high percentage as, although important, the authoring process is not critical in creating adaptive soft skill simulations.

3.5. Task

The authoring process is not related to one specific task, it encompasses aspects from all other requirements that the composition tool includes. It should however be insured that the author goes through at least two iterations of the authoring process. On the second iteration the author should insure they have entered data for every component of the authoring process (except video association). The authoring interface should also be used by the learner twice, once for each iteration of the authoring process. By following each of the evaluation tasks, one after another, the author will in fact be completing one full iteration of the authoring process. Once all other tasks are completed the author should be asked to go through a second iteration of the authoring process without any assistance.

NOTE: all non-model components must be removed before the task is started

3.6. Plan

Appropriate training should be completed by the author before the dialogue representation and the adaptivity training. This will allow the author to have a complete and full overview of how the adaptive soft skill simulations will be completed.

3.7. Training

- diagram explaining how the authoring process works
- diagram explaining what is the purpose of each component
- diagram explaining iterations
- diagram explaining why following the authoring process will allow the authors to compose more effective simulations

3.8. Evaluation

Questionnaire:

The questionnaire should be completed by the authors once the task has been completed. Questionnaire questions for the authoring process are numerically associated under the headings of the evaluation objectives above:

Note: must include note at the beginning that authoring process only includes the high level process and not the dialogue model authoring process.

1. Gauge how effective the author found the authoring process in aiding them to compose the dialogue model

Please select the most appropriate answer to the following statement:
“I found the authoring process described for creating the simulations to be very helpful.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

Please select the most appropriate answer to the following statement:
“By following the authoring process I created the dialogue model that I wanted.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree

- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

Please select the most appropriate answer to the following statement:

"I felt the authoring process supported the composition of a detailed and complete dialogue model."

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

2. Gauge how easy it was for the author to understand the structure of the authoring process

Please select the most appropriate answer to the following statement:

"I found it easy to understand the structure of the authoring process."

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

Please select the most appropriate answer to the following statement:

"I found it easy to follow the iterative approach incorporated in the authoring process."

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

3. Gauge how efficient the author found the authoring process was at allowing them to compose the dialogue model

Please select the most appropriate answer to the following statement:
“The authoring process helped me to quickly compose a detailed well supported dialogue model.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

4. Gauge how efficient the author found the iterative approach incorporated in the authoring process

Please select the most appropriate answer to the following statement:
“I found I could quickly go through each iteration of the authoring process .”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

5. Gauge how comfortable an experience the author found using the authoring interface

Please select the most appropriate answer to the following statement:

"I found it easy to access the authoring process popup window."

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

Please select the most appropriate answer to the following statement:

"I found it easy to understand the authoring process popup window and it presented some useful information."

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

4. Pedagogical Framework

4.1. Description

The simulations that are developed in ACTSim must be educationally effective. To assist the author in creating simulations that meet this criteria the authoring process includes a pedagogical framework. The pedagogical framework allows the author to create learning outcomes that they want the learner to achieve. These learning outcomes can then be associated with parts of the Dialogue Model to insure learning outcomes are being addressed within the simulation.

4.2. Goal

This is the first usability evaluation to examine the pedagogical framework. While this requirement is important, it is not critical to the success of the ACTSim composition tool. The effectiveness, efficiency and user satisfaction of developing a pedagogical framework are all examined in this evaluation. The goal of this evaluation is to insure that the pedagogical framework created by an author will help the author in composing an adaptive soft skill simulation that is educationally effective. The evaluation will examine if the author

found the design of the framework to be helpful. That they understood the principals involved in including the framework in the authoring process. Did the authors find it easy to create the learning outcomes? Could the authors associate (tag) the Dialogue Elements easily?

4.3. Evaluation Objectives

Effectiveness:

1. Gauge how effective the author found the use of learning outcomes to describe the educational goals of the simulation.
2. Gauge how easy it was for the author to learn how to use the learning outcomes.
3. Gauge how effective the authors found tagging the Dialogue Model in order to insure it was educationally effective.
4. Gauge how effective authors found using the Pedagogical Approach

Efficiency:

5. Gauge how efficiently the author could create the learning outcomes (this includes verbs and nouns)
6. Gauge how efficiently the author could associate (tag) the learning outcomes with the Dialogue Model.
7. Gauge how efficient authors found using the Pedagogical Approach

User Satisfaction:

8. Gauge how comfortable an experience the author found using the learning outcomes interface to create the learning outcomes
9. Gauge how comfortable an experience the author found tagging the Dialogue Model with the learning outcomes they created.
10. Gauge how satisfied users were using the Pedagogical Approach.

4.4. Methodology

- **Questionnaire:** there are several objectives that require the opinion of the author to be analyzed. These opinions should be gathered with a questionnaire after the associated task has been completed. The evaluation metric is based on a Likert scale – a high percentage of authors should return favorable opinions on the effectiveness, efficiency and user satisfaction of the pedagogical framework. The questionnaire allows qualitative data to be collected as well as the quantitative. If the authors return non-favorable results the qualitative will indicate why.

Success criteria:

- **Questionnaire:** Authors should return favorable opinions (between 4 and 5 on the Likert scale) of the effectiveness, efficiency and user satisfaction with the implementation and design of the pedagogical framework. These success criteria are assigned a moderately high percentage as, although important, they are not critical in creating adaptive soft skill simulations.

4.5. Task

The authors should each create three learning outcomes which are suitable to the Dialogue Model that they have composed. Appropriate learning verbs should be available to the author but they will be required to create the learning nouns and completed learning outcomes. The author should then associate two of the learning outcomes with appropriate sections of the Dialogue Model

4.6. Plan

Appropriate training should be completed by the author as at a time dictated by the authoring process training. The task should be completed once the Dialogue Model and adaptivity has been completed. It may be necessary to inform the author that they will need to introduce learning outcomes in the composition prior to the evaluation. This will allow them time consider what the learning outcomes will be which will influence the composition of the Dialogue Model. It may in fact be necessary for the authors to create the learning outcomes BEFORE they create the Dialogue Model.

4.7. Training

Training should be completed by the author in a relatively short time (compared to dialogue representation and adaptivity training). The training should include:

- a general definition of learning outcomes
- how learning outcomes are structured
- how learning outcomes are created
- diagram illustrating the Dialogue Model being tagged

4.8. Evaluation

Questionnaire:

The questionnaire should completed by the authors once the task has been completed. Questionnaire questions for the pedagogical framework are numerically associated under the headings of the evaluation objectives above:

1. Gauge how effective the author found the use of learning outcomes to describe the educational goals of the simulation.

Please select the most appropriate answer to the following statement:

"I found using learning outcomes a useful way to describe what I wanted the learner to achieve."

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

2. Gauge how easy it was for the author to learn how to use the learning outcomes.

Please select the most appropriate answer to the following statement:
“I thought that it was easy to learn how to use the learning outcomes while creating my Dialogue Model.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

3. Gauge how effective they authors found tagging the Dialogue Model in order to insure it was educationally effective.

Please select the most appropriate answer to the following statement:
“Tagging sections of the Dialogue Model with the learning outcomes was a good way to associate different parts of the simulation with what I wanted the learner to achieve.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

4. Gauge how effective authors found using the Pedagogical Approach

Please select the most appropriate answer to the following statement:
“I found the Pedagogical Approach was easy to understand”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

5. Gauge how efficiently the author could create the learning outcomes (this includes verbs and nouns)

Please select the most appropriate answer to the following statement:

"I could quickly create my learning outcomes using the verbs and nouns."

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

6. Gauge how efficiently the author could associate (tag) the learning outcomes with the Dialogue Model.

Please select the most appropriate answer to the following statement:

"I could quickly tag appropriate sections of the Dialogue Model with the learning outcomes I created."

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

7. Gauge how efficient authors found using the Pedagogical Approach

Please select the most appropriate answer to the following statement:
“I could quickly access and update a Simulations Pedagogical Appraoch”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

8. Gauge how comfortable an experience the author found using the learning outcomes interface to create the learning outcomes

Please select the most appropriate answer to the following statement:
“I found it easy to use the learning outcomes popup window.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

9. Gauge how comfortable an experience the author found tagging the Dialogue Model with the learning outcomes they created.

Please select the most appropriate answer to the following statement:
“I found it easy to tag the Dialogue Model with learning outcomes that I created.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

10. Gauge how satisfied users were using the Pedagogical Approach.

Please select the most appropriate answer to the following statement:
“I found it easy to access, edit and update the Pedagogical Approach.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

5. Navigational Aids

5.1. Description

The Dialogue Models that are created by the author can become very large and complex. Models can have between 100 and 200 connected Dialogue Elements. Navigation Aids are required to assist the author in navigating the Dialogue Model they have created. Several different approaches have been incorporated in the design of the ACTSim composition tool. These include a map feature, a list feature, an arrange feature, zoom functionality and search. Different approaches have been implemented as authors may have a personal preference to which one to use, different navigational aids may also be more suitable to use in different situations. While an author may not like/use all navigational aids provided they should find a subset which satisfies their requirements.

5.2. Goal

This is the first usability evaluation to examine the most of the navigational aids, the map navigational aid has been successfully evaluated in a previous user evaluation. While this requirement is important, it is not critical to the success of the ACTSim composition tool. The effectiveness, efficiency and user satisfaction of using the navigational aid are all examined in this evaluation. The goal of this evaluation is to insure that an author understands how to use each of the navigational aids. The authors should be able to navigate the dialogue model they have created quickly and easily using the different aids. The authors should also know the correct situations to use the navigational aids. The authors should find the navigational aids easy to use and find them user friendly and intuitive.

5.3. Evaluation Objectives

Effectiveness:

1. Gauge how effective the author found the zoom functionality was for navigating about the dialogue space
2. Gauge how effective the author found the search functionality was for locating dialogue elements in the dialogue space
3. Gauge how effective the author found the list feature was for locating dialogue elements in the dialogue space
4. Gauge how effective the author found the arrange feature was for reducing the complexity of the dialogue
5. Gauge how effective the author found the highlight feature was for locating tagged Dialogue Elements

Efficiency:

6. Gauge how efficient the author found the zoom functionality was for navigating about the dialogue space
7. Gauge how efficient the author found the search functionality was for locating dialogue elements in the dialogue space
8. Gauge how efficient the author found the list feature was for locating dialogue elements in the dialogue space
9. Gauge how efficient the author found the arrange feature was at reducing the complexity of the dialogue
10. Gauge how efficient the author found the highlight feature was for locating tagged Dialogue Elements

User Satisfaction:

11. Gauge how comfortable an experience the author found the zoom functionality was for navigating about the dialogue space
12. Gauge how comfortable an experience the author found the search functionality for locating dialogue elements in the dialogue space
13. Gauge how comfortable an experience the author found the list feature for locating dialogue elements in the dialogue space
14. Gauge how comfortable an experience the author found the arrange feature was for reducing the complexity of the dialogue
15. Gauge how comfortable an experience the author found using the highlight feature was for locating tagged Dialogue Elements

5.4. Methodology

- **Questionnaire:** all of the stated objectives require that the opinion of the author is analyzed. These opinions should be gathered with a questionnaire after the associated task has been completed. The evaluation metric is based on a Likert scale – a moderately high percentage of authors should return favorable opinions on the effectiveness, efficiency and user satisfaction of the navigational aids. While all of the navigational aids are always available to the author they may find one particular suits their personal preferences. The questionnaire allows qualitative data to be collected as well as the quantitative. If the authors return non-favorable results the qualitative will indicate why.

NOTE: navigational aids are self explanatory and can be considered standard operations and concepts that a user would be familiar with – as apposed to dialogue representation/adaptivity/authoring process that would all be very concepts to an author. Training should simply include video screen grabs of the navigational aids in use.

Success criteria:

- **Questionnaire:** the navigational aids are designed to assist the authors in different ways in different situations. One navigational aid might be more suitable to use in a particular situation than another. However, for the majority of the time more than one navigational aid (or combination of navigational aids) can be used by an author to reach their goal. How the authors use the navigational aids to achieve their goal will often come down to personal preference. The success criteria reflects the personal choice of preferring one navigational aid over another. The author should return favorable results (between 4 and 5 on the Likert scale) for at least half of the navigational aids.

5.5. Task

This task is separated from all other tasks in this evaluation. The authors are each provided with a large complex Dialogue Model in which they are instructed to use the different navigational aids to locate particular Dialogue Elements or navigate to particular areas of the Dialogue Model (the model does not need to be related to author's area of expertise – possibly smoking cessation with added adaptivity). The task requires the author to use each of the navigational aids in situations that are best suited to that navigational aid.

Zoom

- begin the task with the Dialogue Model zoomed out at 25%
- author required to zoom into the top rightmost Dialogue Element, using zoom from the palette, to read the Statement
- author required to set the size of the Dialogue Model at 75%

Search

- author required to find all Dialogue Elements that Response contain a particular string
- the author should complete a search and navigate to their location which will be highlighted

List

- the author will be asked to find a particular Dialogue Element that has a Statement that they are provided
- the author should then access the Dialogue Elements Response

Highlight

- the Dialogue Model will contain tagged Dialogue Models that the author will be required to highlight and navigate to
- the author will be required to complete three separate operations:

- highlight all Dialogue Elements with one particular tag
- highlight all Dialogue Elements with two particular tags (OR)
- highlight all Dialogue Elements with two particular tags (AND)

Arrange

- the author will be asked to use the arrange feature

5.6. Plan

Each author is provided adequate training at the beginning of the evaluation about each of the navigational. Once all other tasks are complete the author should then be provided with the Dialogue Model associated with this task. In order to insure continuity the associated questions should appear at the end of the questionnaire.

5.7. Training

A distinction between each of the navigational aids should be made clear during the training. Most of the concepts encapsulated in the navigational aids should be familiar to the authors so do not require a detailed explanation, i.e. search, highlight, zoom, list. Training should be based on video screen capturing of the functions being used and do not require diagrams. Training for the highlight function should however be included with the adaptivity training as part of the adaptivity requirement.

5.8. Evaluation

Questionnaire:

The questionnaire should be completed by the authors once the task has been completed. Questionnaire questions for the navigational are numerically associated under the headings of the evaluation objectives above:

Zoom Function

- I. Gauge how effective the author found the zoom functionality was for navigating about the dialogue space
6. Gauge how efficient the author found the zoom functionality was for navigating about the dialogue space
11. Gauge how comfortable an experience the author found the zoom functionality was for navigating about the dialogue space

Please select the most appropriate answers to the following statements:

"I found the zoom function useful for navigating the about the Dialogue Model."

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

"The zoom functionality allowed me to find what I was looking for very quickly."

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

“I found the zoom icon in the palette and drop down menu that allowed me to select a particular magnification were both very easy to use.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree with any of the questions above.

Search Function

2. Gauge how effective the author found the search functionality was for locating dialogue elements in the dialogue space
7. Gauge how efficient the author found the search functionality was for locating dialogue elements in the dialogue space
11. Gauge how comfortable an experience the author found the search functionality for locating dialogue elements in the dialogue space

Please select the most appropriate answers to the following statements:

“I found the search function very useful for locating Dialogue Elements in the Dialogue Model.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

“The search function allowed me to find particular Dialogue Elements very quickly.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

“I found the search function popup window to be easy and intuitive to use.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree with any of the questions above.

List Function

- 3. Gauge how effective the author found the list feature was for locating dialogue elements in the dialogue space
- 8. Gauge how efficient the author found the list feature was for locating dialogue elements in the dialogue space
- 13. Gauge how comfortable an experience the author found the list feature for locating dialogue elements in the dialogue space

Please select the most appropriate answers to the following statements:
“I found the list feature to be useful in allowing me to locate particular Dialogue Elements in the Dialogue Model.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

“The list function allowed me to find a particular Dialogue Element in the Dialogue Model very quickly.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

“I found the list function window easy and intuitive to use.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree

- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree with any of the questions above.

Arrange Function

4. Gauge how effective the author found the arrange feature was for reducing the complexity of the dialogue
9. Gauge how efficient the author found the arrange feature was at reducing the complexity of the dialogue
14. Gauge how comfortable an experience the author found the arrange feature was for reducing the complexity of the dialogue

Please select the most appropriate answers to the following statements:

"I found the arrange function to be very useful for reducing the complexity of the Dialogue Model."

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

"I found the arrange function executed very quickly and efficiently."

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

"I found the button for using the arrange was easy to locate and use."

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree with any of the questions above.

Highlight Function

- 5. Gauge how effective the author found the highlight feature was for locating tagged Dialogue Elements
- 10. Gauge how efficient the author found the highlight feature was for locating tagged Dialogue Elements
- 15. Gauge how comfortable an experience the author found using the highlight feature was for locating tagged Dialogue Elements

Please select the most appropriate answers to the following statements:
“I found the highlight function to be very useful for locating Dialogue Elements that were tagged for adaptivity.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

“The highlight function allowed me to locate tagged Dialogue Elements very quickly.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

“I found the highlight function popup window intuitive and easy to use.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree with any of the questions above.

6. Validation
6.1. Description

The ACTSim composition tool also includes additional functionality which allows authors to validate their Dialogue Models. This functionality includes the features model verification, script and highlights. Verification allows authors to automatically inspect the whole of the Dialogue Model to insure that essential information has been entered into the Dialogue Model; such as Dialogue Elements Statement and Response values. Script allows authors to view Statement and Responses of Dialogue Elements sequentially to display how the dialogue in a simulation would play out. An extension of this feature allows authors to view, and print, the full script of the entire Dialogue Model which is used when recording the associated video clips. The final feature, Highlight, allows authors to visualize adaptivity Tags within a Dialogue Model so they may view of personalization of the model may take place.

6.2. Goal

Model Verification: the verification function has not been evaluated in any previous usability studies so must be examined with respect to effectiveness, efficiency and user satisfaction.

Script: the script function has not been evaluated in any previous usability studies so must be examined with respect to effectiveness, efficiency and user satisfaction.

Highlight: the highlight function has not been evaluated in any previous usability studies but its evaluation has been fully described in the Adaptivity section of this document. The remainder of this section will only describe the verification and script features.

6.3. Evaluation Objectives

Model Verification

Effectiveness

1. Gauge how effective the author found the verification functionality

Efficiency

2. Gauge how effective the author found the verification functionality

User Satisfaction

3. Gauge how comfortable an experience the author found using the verification functionality

Script

Effectiveness

4. Gauge how effective the author found the script functions

Efficiency

5. Gauge how effective the author found the script functions

User Satisfaction

6. Gauge how comfortable an experience the author found using the script functions

6.4. Methodology

- **Questionnaire:** all of the stated objectives require that the opinion of the author is analyzed. These opinions should be gathered with a questionnaire after the associated tasks have been completed. The evaluation metric is based on a Likert scale – a relatively high percentage of authors should return favorable opinions on the effectiveness, efficiency and user satisfaction for each of the features that are being examined.

Success criteria:

- The author should return favorable results (between 4 and 5 on the Likert scale) for each area of the validation functions that are being examined.

6.5. Task

Model Verification: the author should use the verification functionality to validate the Dialogue Model that they have been presented.

Script: the author should use both types of script functionality to examine section of the Dialogue Model that they have been presented. Once complete the authors should generate an entire full script of the Dialogue Model.

6.6. Plan

Model Verification/Script: the author should be provided with associated training before beginning the task(s). The author should complete the task with the demonstration Dialogue Model that they has been provided

6.7. Training

Model Verification: this function is unlikely to be familiar to the authors so may require a more detailed explanation. Training should include both diagrams and video capturing.

Script: both of these functions should be intuitive to most authors and require only a short amount of instruction. Training should include both diagrams and video capturing.

6.8. Evaluation

Questionnaire:

The questionnaire should completed by the authors once the task has been completed. Questionnaire questions for the pedagogical framework are numerically associated under the headings of the evaluation objectives above:

1. Gauge how effective the author found the verification functionality
Please select the most appropriate answer to the following statement:

“I found the verification function was very helpful for finding mistakes in the structure of the model.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

2. Gauge how efficient the author found the verification functionality
Please select the most appropriate answer to the following statement:

“I thought the verification function found mistakes in my model quickly.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

3. Gauge how comfortable an experience the author found using the verification functionality

Please select the most appropriate answer to the following statement:

“I thought the information presented by the verification function was intuitive and easy to understand.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

4. Gauge how effective the author found the script functions
Please select the most appropriate answer to the following statement:
“I found that scripting small segments of the dialogue model that I created was very useful in allowing me to see the flow of my simulation.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

Please select the most appropriate answer to the following statement:
“I thought that generating a full script of the dialogue would be useful when recording the video scripts.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

5. Gauge how effective the author found the script functions
Please select the most appropriate answer to the following statement:
“I thought that scripting segments of the dialogue model and generating the flow of the dialogue was a quick process.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

Please select the most appropriate answer to the following statement:

“I thought that scripting the whole dialogue model was quick.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

6. Gauge how comfortable an experience the author found using the script functions

Please select the most appropriate answer to the following statement:

“I thought that using the script functions was easy and the scripts that were generated were easy to follow.”

- Strongly agree
- Agree
- Neither agree/nor disagree
- Disagree
- Strongly disagree

Please give reasons for your answer if you did not Agree or Strongly Agree.

Usability Evaluation Plan

Completed Task

All tasks should be compiled into single document which is delivered to the authors at appropriate stages of the evaluation. The first task should follow the steps in the authoring process without adaptivity i.e. authoring the DM if necessary. The second iteration of the authoring process should including adaptivity. The third iteration of the authoring process should be completed by the author without prompting. The final task, involving an addition model, should then be completed by the author.

Training Documentation

All training documentation should follow their descriptions and be developed separately with associated voice recording. Each segment should then be delivered to the authors with time between them to allow for questions. A slide should be included to insure the authors know they can ask questions.

Training Analysis

A question classification scheme is required to analyse the questions that the authors asks during training. Each requirement that involves training analysis should have a separate question classification schemas.

Video Analysis

An error classification scheme is required to analyse the video captured of the authors completing the tasks. Each requirement that involves video analysis should have a separate error classification schema.

Test (Examination) Document

All test data should be compiled into a single document. This document should be comprised of the test the authors should complete, a separate document which includes the answers used to correct the tests and a document for compiling the authors scores.

Questionnaire Document

All evaluation questions should be compiled into a single document. The questions should be suitably presented and be numbered from start to finish. Subheadings may be required to break up the questionnaire. A separate questionnaire should also be developed to collect a profile of each author.

Evaluators Document

A document that details the steps involved in the evaluation that the evaluator should follow so each user is delivered the same evaluation and to insure that no steps are overlooked. The document should be printed for each separate evaluation and include the authors name and boxes that can be ticked by the evaluator as each step in completed.

Users

Authors should be experts in an aspect of soft skills. They should initially be given a brief description of the project and composition tool. If the authors have previous experience with the composition tool they qualify immediately. At least five users are required to complete the usability evaluation. More than five would be preferable. A questionnaire should be used to gather each author's personal data.

Pre-Evaluation Document

The authors should be given some initial instructions leading up to the usability evaluation. This will allow the authors to think about a scenario that they want to develop – if necessary. They should be told that the scenario should possibly include learning outcomes, roles, subjects, etc. This will allow the authors to more efficiently spend their time in the usability evaluation (i.e. spending more time using the composition tool rather than sitting reflecting how they might use the composition tool). The authors should also be asked at this time to complete the pre-evaluation questionnaire.

Pos-Evaluation Document

This document should describe how data should be analyzed and compared to success criteria.

Ethics Approval

All relevant documentation required by the ethics approval committee.

Equipment

- ACTSim installation (including a dialogue model for final task)

- AV Room
- Camtasia installation
- Laptop
- Mouse

Pre-Questionnaire

ACTSim Composition Tool

Pre-Questionnaire

The purpose of this questionnaire is to gather details regarding your previous experience and background.

PLEASE COMPLETE WITH BLOCK CAPITALS AND TICK APPROPRIATE BOXES

Assigned User Name: _____

1. Age:

2. Sex:

Male	☐
Female	☐

3. Professional/Academic Position:

4. Which of the following would best describe your prior experience with using a computer?

Very Experienced (e.g. technical background/formal qualification)	☐
Some what experienced (e.g. Microsoft Office, research, etc)	☐
Occasionally used a computer (e.g. internet, email, etc)	☐
Never used a computer before	☐

5. Which of the following would best describe your prior experience with eLearning in general?

Experienced (have developed multiple courses)	☐
Some what experienced (have used as a instructor)	☐
Novice (have used as a learner)	☐
No experience	☐

6. Which of the following would best describe your prior experience with adaptivity or personalization?

Experienced (have developed multiple courses)	☐
Some what experienced (have used as a instructor)	☐
Novice (have used as a learner)	☐
No experience	☐

7. Which of the following would best describe your prior experience with eLearning composition tools (not ACTSim)?

Experienced (used more than one other eLearning composition tool)	☐
Some what experienced (used one other eLearning composition tool)	☐
Novice (only used ACTSim previously)	☐
No experience	☐

8. Please name and describe the eLearning composition tools (if any) with which you have prior experience.

9. How many titles have you previously developed using ACTSim?

10. Please describe briefly the purpose of the ACTSim title(s) that you previously developed.

Test

ACTSim Evaluation Test

Please answer these questions to the best of you ability.

PLEASE COMPLETE WITH BLOCK CAPITALS

Assigned User Name: _____

1. What are the two main properties that each dialogue element contains?
2. What do the arrows that connect the dialogue elements represent?
3. What does the model within ACTSim represent?
4. How does this model relate to the simulations that are authored?
5. Briefly (one or two sentences) explain how the dialogue model can be made adaptive.
6. Briefly (one or two sentences) explain how tagging allows the dialogue to become adaptive.

7. Briefly (one or two sentences) explain how tagging the dialogue will effect the final soft skill simulation.

8. What is a trigger?

9. Briefly (one or two sentences) explain how a trigger in the dialogue model will effect the final simulation.

10. Briefly (one or two sentences) explain how the triggers are adaptive.

Post-Questionnaire

ACTSim Composition Tool

Post-Questionnaire

The purpose of this questionnaire is to determine your opinion on several aspects of the ACTSim composition tool.

Assigned User Name: _____

PLEASE COMPLETE WITH BLOCK CAPITALS AND TICK APPROPRIATE BOXES

ONLY ADD COMMENTS IF YOU FEEL APPROPRIATE (PARTICULARILY IF YOU DISAGREE OR STRONG DISAGREE WITH THE STATEMENT)

Dialogue Model

1. Please select the most appropriate answer to the following statement:

“I felt it was easy to understand how the dialogue model represents the dialogue that occurs in the soft skill simulation”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

2. Please select the most appropriate answer to the following statement:

“The approach of breaking down the dialogue into smaller components and using arrows to connect them was easy to understand”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

3. Please select the most appropriate answer to the following statement:

“The use of dialogue elements and arrows allowed me to create the dialogue that I wanted”

- Strongly agree ☐
 - Agree ☐
 - Neither agree/nor disagree ☐
 - Disagree ☐
 - Strongly disagree ☐
-

4. Please select the most appropriate answer to the following statement:

“Breaking the dialogue into components of ‘statement’ and ‘response’ was good way of encapsulating the components of the dialogue”

- Strongly agree ☐
 - Agree ☐
 - Neither agree/nor disagree ☐
 - Disagree ☐
 - Strongly disagree ☐
-

5. Please select the most appropriate answer to the following statement:

“Authoring a dialogue with components that were comprised of ‘statement’ and ‘response’ was easy”

- Strongly agree ☐
- Agree ☐

- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

6. Please select the most appropriate answer to the following statement:

“The arrows that connect the dialogue elements easily allowed me to create all the paths that I wanted available to the learner”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

Adaptivity

7. Please select the most appropriate answer to the following statement:

“I thought tagging the dialogue model was a good way for me to indicate where I wanted the adaptivity to be in the dialogue.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

8. Please select the most appropriate answer to the following statement:

“It was easy to see how tagging would effect the final simulation.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

9. Please select the most appropriate answer to the following statement:

“I liked the fact that tagging in the dialogue model could only be seen when the model was highlighted.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

10. Please select the most appropriate answer to the following statement:

“I thought properties available for tagging were suitable to the types of adaptivity that I wanted to apply.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

11. Please select the most appropriate answer to the following statement:

“I thought the use of adaptive triggers was a good way for me to indicate where I wanted personalised assessment, feedback and reflection to occur in the dialogue.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

12. Please select the most appropriate answer to the following statement:

“I liked the way the triggers were represented in the Dialogue Model.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

13. Please select the most appropriate answer to the following statement:

“I thought the properties available to tag the triggers were suitable for how I wanted to personalize the dialogue.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐

- Disagree ☐
- Strongly disagree ☐

14. Please select the most appropriate answer to the following statement:

“I found it easy to create the tagging properties I used for adaptivity.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

15. Please select the most appropriate answer to the following statement:

“The popup window which displayed the adaptivity properties was intuitive and easy to follow.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

16. Please select the most appropriate answer to the following statement:

“I found it easy to tag a single Dialogue Element for adaptivity.”

- Strongly agree ☐

- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

17. Please select the most appropriate answer to the following statement:

“I found it easy to tag a group of Dialogue Elements for adaptivity.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

18. Please select the most appropriate answer to the following statement:

“I found it easy to highlight Dialogue Elements that were tagged for adaptivity.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

19. Please select the most appropriate answer to the following statement:

“I found it easy to find the highlighted Dialogue Elements in the Dialogue Model.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

20. Please select the most appropriate answer to the following statement:

“I found it easy to delete tags from Dialogue Elements.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

21. Please select the most appropriate answer to the following statement:

“I found it easy to add triggers to the Dialogue Elements.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

22. Please select the most appropriate answer to the following statement:

“I found it easy to access the properties of the triggers.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

23. Please select the most appropriate answer to the following statement:

“The popup window which displayed the trigger properties was intuitive and easy to follow.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

24. Please select the most appropriate answer to the following statement:

“I found it easy to move the triggers to other Dialogue Element compartments.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

25. Please select the most appropriate answer to the following statement:

“I found it easy to delete the triggers from the Dialogue Model.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

Authoring Process

26. Please select the most appropriate answer to the following statement:

“I found the authoring process described for creating the simulations to be very helpful.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

Please select the most appropriate answer to the following statement:

“By following the authoring process I created the dialogue model that I wanted.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

27. Please select the most appropriate answer to the following statement:

“I felt the authoring process supported the composition of a detailed and complete dialogue model.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

28. Please select the most appropriate answer to the following statement:

“I found it easy to understand the structure of the authoring process.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

29. Please select the most appropriate answer to the following statement:

“I found it easy to follow the iterative approach incorporated in the authoring process.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

30. Please select the most appropriate answer to the following statement:

“The authoring process allowed me to quickly compose a well supported dialogue model.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

31. Please select the most appropriate answer to the following statement:

“I found I could quickly go through each iteration of the authoring process.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

32. Please select the most appropriate answer to the following statement:

“I found it easy to use the authoring process popup window.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

33. Please select the most appropriate answer to the following statement:

“I found it easy to understand the authoring process popup window and it presented some useful information.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

Pedagogical Framework

34. Please select the most appropriate answer to the following statement:

“I found using learning outcomes a useful way to describe what I wanted the learner to achieve.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

35. Please select the most appropriate answer to the following statement:

"I thought that it was easy to learn how to use the learning outcomes (that they werent too complex)."

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

36. Please select the most appropriate answer to the following statement:

"Tagging sections of the Dialogue Model with the learning outcomes was a good way to associate different parts of the simulation with what I wanted the learner to achieve."

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

37. Please select the most appropriate answer to the following statement:

"I could quickly create my learning outcomes using the verbs and nouns."

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

38. Please select the most appropriate answer to the following statement:

“I could quickly tag appropriate sections of the Dialogue Model with the learning outcomes I created.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

39. Please select the most appropriate answer to the following statement:

“I found it easy to use the learning outcomes popup window.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

40. Please select the most appropriate answer to the following statement:

“I found it easy to tag the Dialogue Model with learning outcomes that I created.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐

- Strongly disagree ☐

Zoom Function

41. Please select the most appropriate answers to the following statements:

“I found the zoom function useful for navigating about the Dialogue Model.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

“The zoom functionality allowed me to find what I was looking for quickly.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

“I found the zoom icon in the palette and drop down menu that allowed me to select a particular magnification were both very easy to use.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

Search Function

42. Please select the most appropriate answers to the following statements:

“I found the search function very useful for locating Dialogue Elements in the Dialogue Model.”

- Strongly agree ☐
- Agree ☐

- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

“The search function allowed me to find particular Dialogue Elements very quickly.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

“I found the search function popup window to be easy and intuitive to use.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

List Function

43. Please select the most appropriate answers to the following statements:

“I found the list feature to be useful in allowing me located particular Dialogue Elements in the Dialogue Model.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

“The list function allowed me to find a particular Dialogue Element in the Dialogue Model very quickly.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

“I found the list function window easy and intuitive to use.”

- Strongly agree ☐
- Agree ☐

- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

Arrange Function

44. Please select the most appropriate answers to the following statements:

“I found the arrange function to be very useful for reducing the complexity of the Dialogue Model.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

“I found the arrange function executed very quickly and efficiently.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

“I found the button for using the arrange was easy to locate and easy to use.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

Highlight Function

45. Please select the most appropriate answers to the following statements:

“I found the highlight function to be very useful for locating Dialogue Elements that I tagged for adaptivity.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

“The highlight function allowed me to locate tagged Dialogue Elements very quickly.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

“I found the highlight function popup window intuitive and easy to use.”

- Strongly agree ☐
- Agree ☐
- Neither agree/nor disagree ☐
- Disagree ☐
- Strongly disagree ☐

Error Classification

Error Classification Scheme ACTSim Usability Evaluation

Overview

This document describes the error classification scheme associated with the second ACTSim usability evaluation. The error classification is employed is based on Reason’s (Human Error, 1990) approach. This approach allows errors to be classified into three categories; slip; lapse; and mistake. Both a slip and a lapse are unintended actions, while a mistake is associated with an intended action. Table 1 presents examples for each error category.

Error Category	Example
Slip	Action Intrusion Omission of Action Reversal of Action Miss-ordering of Action Mistiming of Action
Lapse	Omitting of Planned Actions Losing Place in Action Sequence Forgetting Intended Actions
Mistake	Misapplication of Good Procedure Application of Bad Procedure Misperception Poor Decision Making Failure to Consider Alternatives Overconfidence

Table 1: Reason’s Error Classification with Examples

Errors made by the user may have different implications depending on the error type and their frequency. For example, a low number of slips or lapse’s may be acceptable, especially if the user is using software for the first time or with a new feature that they have little experience. However, if there are a great deal of these types of errors with users that have a great deal of experience with the software it might mean that the software is difficult to use wghich would slow efficiency and highlight usability issues. A large number of mistakes may be the result of poor training or an action that is difficult to complete.

Requirement: Adaptivity Application and Representation

Associated Evaluation Objectives from Adaptivity Requirement:
(Objective numbers taken from Usability Evaluation Document)

- 13. The authors could efficiently create the tagging properties.
- 14. The author could efficiently tag individual Dialogue Elements
- 15. The author could efficiently tag groups of Dialogue Elements
- 16. The author could efficiently locate Dialogue Elements with certain tags
- 17. The author could efficiently remove tags from the Dialogue Elements
- 18. The authors could efficiently add the triggers to the Dialogue Elements
- 19. The authors could efficiently move the triggers in the Dialogue Space
- 20. The authors could efficiently delete the triggers from the Dialogue Elements
- 21. The authors could efficiently access the triggers properties

Examples of errors for each category under the above Evaluation Objectives

- 13. The authors could efficiently create the tagging properties.

Error Category	Example
Slip	• Author accidentally selects the wrong button to open tagging properties popup box but corrects • Author accidentally selects updates/add/delete/clear so must repeat action
Lapse	• Momentarily forgets location of popup properties (attempts to find location in different parts of application) • Forgets to input description of the property • Takes time considering how to updates/add/delete/clear
Mistake	• Enters name in description text box and description in name text box • Updates existing property instead of adding new property and thinks they added a new property successfully

14. The author could efficiently tag individual Dialogue Elements

Error Category	Example
Slip	• Author selects the wrong tagging property within in the Dialogue Element property display • Author attempts to exit the DE property box without saving • Author accidentally selects the wrong tagging property (from drop down menu) so must exit and repeat the action
Lapse	• The author can't remember where to find the DE tagging properties (tries several avenues before locating the correct one procedure) • The author attempt to select several instances under one tagging property without holding ctrl button but corrects themselves
Mistake	• Author thinks unselected items in DE property box are selected items and selected items are unselected • Author thinks tab (broken line) box represents the selection

15. The author could efficiently tag groups of Dialogue Elements

Error Category	Example
Slip	• Author selects items that are not DEs such as Triggers and Arrows • Author accidentally selects the wrong tagging property (from drop down menu) so must exit and repeat the action • Author accidentally selects apply/remove/cancel so must repeat action
Lapse	• Author tries to select multiple DE without holding ctrl button but corrects themselves • Author is at first not sure how to apply the tagging but eventually remembers where the function is located
Mistake	• Author does not select anything in the tagging popup window and pushes the apply button thinking they have applied all tags that are visible

16. The author could efficiently locate Dialogue Elements with certain tags

Note: task for this evaluation objective is described in Navigational Aids

Error Category	Example
Slip	• Author accidentally select the wrong tagging property that they want to highlight so they repeat the action • Author accidentally selects highlight/clear/cancel button
Lapse	• Author cannot, at first, remember location of the highlight function • Author forget to hold ctrl button for multiple selections
Mistake	• Author thinks the unselected are selected and selected are unselected in the highlight popup window

17. The author could efficiently remove tags from the Dialogue Elements

Error Category	Example
Slip	• Author selects the wrong tagging property within in the Dialogue Element property display • Author attempts to exit the DE property box without saving • Author accidentally selects the wrong tagging property (from drop down menu) so must exit and repeat the action
Lapse	• Author tries to select multiple DE without holding ctrl button but corrects themselves • Author is at first not sure how to remove tagging but eventually remembers where the function is located
Mistake	• Author thinks the unselected are selected and selected are unselected in the highlight popup window

18. The authors could efficiently add the triggers to the Dialogue Elements

Error Category	Example
Slip	• Author accidentally misses the compartment of the DE and places it outside but then moves it into the compartment • Author selects the wrong trigger or other palette function
Lapse	• Author tries to put trigger in the main body of the DE instead of the compartment, realises there mistake and then puts it in the compartment • Author has difficult remember location of the triggers or how to add them
Mistake	• Author places trigger in main body of the and thinks they have successfully added the trigger • Author places the trigger beside the DE and thinks they have successfully added the trigger

19. The authors could efficiently move the triggers in the Dialogue Space

Error Category	Example
Slip	• Author accidentally misses the compartment of the DE and places it outside but then moves it into the compartment
Lapse	• Author tries to put trigger in the main body of the DE instead of the compartment, realises there mistake and then puts it in the compartment
Mistake	• Author places the trigger beside the DE and thinks they have successfully added the trigger

20. The authors could efficiently delete the triggers from the Dialogue Elements

Error Category	Example
Slip	• Author does not select the trigger correctly so when they try to delete it nothing happens
Lapse	• Author cannot remember how to delete the trigger, tries several avenues, e.g. tries to delete from it popup property box, but does eventually remember
Mistake	• Author move the trigger outside of the compartment thinking that the trigger is now deleted

21. The authors could efficiently access the triggers properties

Error Category	Example
Slip	• Author accidently double clicks the DM
Lapse	• Author has difficulty remembering how to access the trigger properties but finds the correct action
Mistake	• Author updates something in the DM property box thinking that is related to the triggers properties

Error Correlation

Creating Tagging Properties

13. The authors could efficiently create the tagging properties.

Error Category	Description	Total
Slip		
Lapse		
Mistake		
Total		

Tagging Individual Dialogue Elements

14. The author could efficiently tag individual Dialogue Elements

Error Category	Description	Total
Slip		
Lapse		
Mistake		
Total		

Tagging Groups of Dialogue Elements

15. The author could efficiently tag groups of Dialogue Elements

Error Category	Description	Total
Slip		
Lapse		
Mistake		
Total		

Remove Tags from Dialogue Elements

17. The author could efficiently remove tags from the Dialogue Elements

Error Category	Description	Total
Slip		
Lapse		
Mistake		
Total		

Adding the triggers to Dialogue Elements

18. The authors could efficiently add the triggers to the Dialogue Elements

Error Category	Description	Total
Slip		
Lapse		
Mistake		
Total		

Moving Triggers in Dialogue Space

19. The authors could efficiently move the triggers in the Dialogue Space

Error Category	Description	Total
Slip		
Lapse		
Mistake		
Total		

Delete Triggers from Dialogue Elements

20. The authors could efficiently delete the triggers from the Dialogue Elements

Error Category	Description	Total
Slip		
Lapse		
Mistake		
Total		

Delete Triggers from Dialogue Elements

21. The authors could efficiently delete the triggers from the Dialogue Elements

Error Category	Description	Total
Slip		
Lapse		
Mistake		
Total		

Highlighting Dialogue Elements

16. The author could efficiently locate Dialogue Elements with certain tags

Error Category	Description	Total
Slip		
Lapse		
Mistake		
Total		

Appendix XIV – Evaluation Results - Model Validation

