

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

Developing An Authoring Environment For Procedural Task Tutoring Systems

A dissertation presented in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Computer Science at Massey University, Palmerston North, New Zealand.

Shamus Paul Smith

1997

Abstract

The use of computers in education is becoming more and more common as the price of technology drops and its general availability is increased. Unfortunately, building computer based tutoring systems is a difficult process which is fraught with many problems. A significant problem in this area is the lack of reuse of system components between computer tutor developments. This means that each new system must be started from scratch and mistakes from earlier projects can easily be repeated. A complementary difficulty is the variety of specialist skills that are required to build these systems. Typical developers do not usually possess the combination of domain, cognitive science and programming knowledge that is needed to build computer tutors. One solution to these problems is the use of an authoring environment for facilitating the building of computer based tutoring systems.

This thesis presents an authoring tool for the construction of computer based tutoring systems teaching procedural tasks in a discovery learning environment. TANDEM (Task ANd Domain Environment Model) provides tools for domain and task definition, sub-domain definition and a domain independent tutoring engine.

It is argued that such an environment can provide a non-expert user with access to advanced techniques from artificial intelligence research for knowledge acquisition and representation. Several tasks from the construction process have been automated, thus simplifying this activity. The use of sub-domain partitioning has been considered and techniques for the integration of custom built domain interfaces are described. Also, it is proposed that by providing a domain independent tutoring engine, reuse can be encouraged over numerous domains which can reduce the development time required to build these systems.

Acknowledgements

Firstly, I would like to thank my main supervisor, Ray Kemp, for his help, support and encouragement over the course of this research. He provided the motivation for this topic, co-authored several publications and proof-read multiple versions of this thesis.

Also, I have appreciated the support provided by my two second supervisors, Daniela Mehandjiska-Stavreva and Mark Apperley, especially Daniela's observations on the first draft of this thesis.

Thanks is also necessary to the FIMS Computer Support Staff for keeping all the equipment working and their flexibility and consideration, i.e. network disk space, during the write-up period of this thesis.

I am also grateful for the financial assistance provided by Massey University in the form of a MURF (Massey University Research Fund) grant, a Massey University Doctoral Scholarship and a graduate assistant position in the department of Computer Science.

Four years is a long time to keep focused and motivated on one topic and the emotional pressures of post-graduate research can be intense. I have been fortunate to have been able to share this time with four years of Honours, Masterate and Doctoral students and Computer Support Staff from the Computer Science, Information Systems and Mathematics departments. This has made life bearable with all its ups and downs.

Finally, I wish to thank my parents for their continuing encouragement and optimism.

Publications

The following publications are associated with the research presented in this thesis.

Journal Article

Kemp, R. H. and Smith, S. P. (1994). Domain and task representation for tutorial process models. *International Journal of Human Computer Studies*, 41(3), 363-383.

Conference Proceedings

Kemp, R. H. and Smith, S. P. (1994). Teaching procedural skills by computer simulation. In *Second Singapore International Conference on Intelligent Systems (SPICIS '94)*, (pp. B73-B78), Singapore.

Kemp, R. H. and Smith, S. P. (1994). Using Planning Techniques to Provide Feedback in Interactive Learning Environments. In P. T. Metaxas (Ed.), *Sixth International Conference on Tools with Artificial Intelligence*, (pp. 700-703). New Orleans, USA: IEEE Computer Society Press.

Smith, S. P. and Kemp, R. H. (1995). Development of a Discovery Learning Tutoring System Construction Environment. In J. Greer (Ed.), *AI-ED 95 : World Conference on Artificial Intelligence in Education*, (pp. 595). Washington DC, USA: AACE : Association for the Advancement of Computing in Education.

Smith, S. P. and Kemp, R. H. (1995). Efficient Modelling of Domains for Computer Tutoring Systems. In R. Kotagiri (Ed.), *ACSC 95 : Proceedings of the 18th Australasian Computer Science Conference*, Vol. 17, Number 1 (pp. 491-498). Adelaide, South Australia: Australian Computer Science Communications.

Kemp, R. H. and Smith, S. P. (1996). A Visual Approach to Procedural Tutor Specification. In J. Grundy and M. Apperley (Eds.), *Sixth Australian Conference on Computer-Human Interaction*, (pp. 190-196). Hamilton, New Zealand: IEEE Computer Society Press.

Internal Report

Kemp, R. H. and Smith, S. P. (1994). Facilitating feedback in discovery learning systems. *Information and Mathematical Sciences Report No. 94/19*. Massey University.

Table of Contents

Chapter 1 Introduction	1
1.1 Background.....	1
1.2 Context of the Research.....	6
1.2.1 Motivation.....	6
1.2.2 Thesis.....	7
1.3 Thesis Outline.....	7
Chapter 2 Intelligent Tutoring Systems	9
2.1 Introduction.....	9
2.2 CAI and Education.....	10
2.3 CAI and ITS.....	11
2.3.1 Domain Knowledge.....	14
2.3.2 Student Model.....	15
2.3.3 Tutoring Engine.....	18
2.3.4 System Interface.....	20
2.4 Knowledge Representation.....	21
2.5 Discovery Learning Systems.....	24
2.6 Environments for Authoring ITSs.....	26
2.7 Summary.....	31
Chapter 3 ITS Authoring Tools	33
3.1 Introduction.....	33
3.2 ITS Authoring Tools.....	34
3.3 Expert System Shells and ITS Authoring Tools.....	37
3.4 ITSAT Requirements.....	40
3.4.1 Reuse.....	40
3.4.2 System Control.....	43
3.4.3 Visual Notations for Knowledge Acquisition.....	44
3.4.4 Developing Domain Interfaces.....	45
3.4.5 General vs. Special Purpose Tools.....	47
3.5 Teaching Procedural Skills.....	49
3.5.1 Static World Assumption.....	51
3.6 Summary.....	52
Chapter 4 Knowledge Representation	55
4.1 Introduction.....	55
4.2 Domain Model.....	55
4.2.1 Defining the Domain Model.....	57
4.2.2 POP Tables.....	58
4.2.3 Graphical POP.....	62
4.3 Task Model.....	67
4.3.1 Defining the Task Model.....	69
4.4 Sub-Domain Generation.....	75
4.5 Summary.....	81

Chapter 5 Teaching Strategies	83
5.1 Introduction.....	83
5.2 Feedback.....	83
5.3 Feedback Approaches.....	86
5.3.1 Optimal Path Approach.....	86
5.3.2 Authoritarian Approach.....	87
5.3.3 Issue Based Approach.....	88
5.3.4 Device Based Approach.....	88
5.3.5 Primitive Operations Approach.....	89
5.3.6 High Order Operations Approach.....	89
5.4 Domain Feedback.....	91
5.5 Task Feedback.....	94
5.6 Teaching Strategies.....	98
5.7 Teaching Strategy Selection.....	106
5.8 Summary.....	108
Chapter 6 TANDEM	109
6.1 Introduction.....	109
6.2 TANDEM (Task ANd Domain Environment Model).....	110
6.3 DCE (Domain Construction Environment).....	116
6.3.1 The DCE Interface.....	117
6.3.2 Defining the Domain Model.....	121
6.3.3 Testing the Domain Model.....	124
6.3.4 Defining Task Information.....	125
6.3.5 POP Code Generation.....	128
6.3.6 POP Code Validation.....	130
6.3.7 Interface Template Generation.....	131
6.4 GDI (General Domain Interpreter).....	133
6.4.1 Domain Independence and TANDEM.....	136
6.4.2 Platform Independence and TANDEM.....	137
6.5 Summary.....	140
Chapter 7 TANDEM Examples	143
7.1 Introduction.....	143
7.2 GDI Sessions.....	143
7.3 A Sample GDI Session.....	144
7.4 Example 1 : Facsimile Machine Troubleshooting.....	147
7.4.1 The Domain.....	147
7.4.2 The Domain Definition.....	148
7.4.3 The Interface.....	149
7.4.4 A Sample Session.....	150
7.5 Example 2 : Car Maintenance.....	151
7.5.1 The Domain.....	151
7.5.2 The Domain Definition.....	151
7.5.3 The Interface.....	152
7.5.4 A Sample Session.....	155
7.6 Example 3 : Radio Studio Usage.....	156
7.6.1 The Domain.....	156
7.6.2 The Domain Definition.....	157
7.6.3 The Interface.....	158
7.6.4 A Sample Session.....	160
7.7 Summary.....	162

Chapter 8	Conclusions and Future Work	163
8.1	Introduction.....	163
8.2	Summary of the Research.....	163
8.3	Contributions of Research.....	165
	8.3.1 Theory Work.....	165
	8.3.2 Practical Work.....	166
8.4	Future Work.....	167
References		169
Glossary		185
Appendix A	Projection Graphs In TANDEM	197
A.1	Introduction.....	197
A.2	Node Placement.....	197
A.3	Iterative Deepening.....	198
A.4	Transition Selection.....	200
A.5	Generating Projection Graphs.....	202
A.6	Summary.....	204
Appendix B	DCE Generated Files	205
B.1	Introduction.....	205
B.2	The Layout File.....	206
B.3	The POP File.....	208
B.4	The Task File.....	209
B.5	The Interface File.....	210
B.6	Summary.....	212
Appendix C	TANDEM Domain and Platform Drivers	213
C.1	Introduction.....	213
C.2	GDI Control Unit.....	214
C.3	Platform Driver.....	215
C.4	Power Macintosh Platform Driver.....	215
C.5	Domain Driver.....	216
C.6	Power Macintosh Default Text Interface.....	217
C.7	Summary.....	218
Appendix D	Domain Complexity	219
D.1	Introduction.....	219
D.2	Domain Complexity.....	219
D.3	Complexity Formula 1a.....	220
	D.3.1 Proof by Example.....	221
	D.3.2 Proof by Induction for General Node Formula.....	223
	D.3.3 Proof by Induction for General Transition Formula.....	223
D.4	Complexity Formula 1b.....	224
	D.4.1 Proof.....	225
	D.4.2 Example.....	225
D.5	Summary.....	226

Figures

Figure 1.1	ITS development problems.....	5
Figure 2.1	Common ITS model.....	13
Figure 2.2	Semantic network example, a knowledge representation method.....	14
Figure 2.3	The Cardiac Tutor.....	21
Figure 2.4	Geometer's Sketchpad and a tutoring agent for geometric construction.....	27
Figure 2.5	MOBIT Architecture.....	29
Figure 2.6	Defining the behaviour of simulated objects in Rides.....	30
Figure 2.7	CALVIN screen display.....	31
Figure 3.1	ITS development environment overview.....	35
Figure 3.2	Domain knowledge and representation technique separation.....	38
Figure 3.3	Components with domain dependence and independence.....	39
Figure 3.4	Sharing of independent domain knowledge.....	41
Figure 3.5	Two approaches to interface development.....	45
Figure 3.6	ITS interface separation.....	46
Figure 3.7	Separating ITS components.....	49
Figure 4.1	Transition network example.....	57
Figure 4.2	Example POP table.....	59
Figure 4.3	Example maintenance static axiom.....	60
Figure 4.4	Modified POP table and standard static axiom.....	61
Figure 4.5	Further static axiom examples.....	61
Figure 4.6	Procedural net for painting.....	63
Figure 4.7	Basic plan net structure.....	64
Figure 4.8	Plan net and POP table.....	64
Figure 4.9	Condition types.....	65
Figure 4.10	Graphical standard static axioms.....	65
Figure 4.11	Graphical maintenance static axioms.....	65
Figure 4.12	Static axiom rules.....	66
Figure 4.13	Graphical POP with inessential conditions.....	66
Figure 4.14	Graphical POP using maintenance rules.....	66
Figure 4.15	Task overlays on the domain model.....	69
Figure 4.16	Example projection graph for a microwave domain.....	70
Figure 4.17	Preferred path through the projection graph.....	71
Figure 4.18	Indicating encouraged paths through the projection graph.....	72

Figure 4.19	General SCRs in the microwave domain.....	73
Figure 4.20	SCR critical choice points.....	73
Figure 4.21	Example student task models.....	74
Figure 4.22	Task structures over a domain.....	75
Figure 4.23	VCR sub-domain separation.....	76
Figure 4.24	Two independent sub-domains.....	77
Figure 4.25	The relations between VCR sub-domains.....	78
Figure 4.26	Independent VCR sub-domains.....	78
Figure 4.27	Sample VCR domain.....	79
Figure 4.28	Automatic generated VCR sub-domains.....	79
Figure 4.29	Generated sub-domains.....	80
Figure 5.1	Feedback from the <i>why</i> command.....	92
Figure 5.2	Feedback from the <i>how</i> command.....	93
Figure 5.3	Example of feedback dialogue in a car maintenance domain.....	94
Figure 5.4	Example task structure for a car maintenance domain.....	95
Figure 5.5	Example dialogue of task feedback in a car maintenance domain.....	97
Figure 5.6	Teaching strategy and knowledge representation association.....	103
Figure 5.7	Teaching strategy summary.....	105
Figure 6.1	TANDEM welcome screen.....	109
Figure 6.2	Basic authoring sequence.....	111
Figure 6.3	The Natural Laboratory.....	111
Figure 6.4	Automated scheme for developing a procedural task tutor.....	112
Figure 6.5	High level view of the TANDEM methodology.....	113
Figure 6.6	Domain component of TANDEM.....	114
Figure 6.7	Task component of TANDEM.....	114
Figure 6.8	Interface component of TANDEM.....	115
Figure 6.9	TANDEM implementation cycle.....	115
Figure 6.10	The DCE environment.....	118
Figure 6.11	DCE object manipulation tools.....	118
Figure 6.12	DCE environment manipulation tools.....	119
Figure 6.13	DCE object tools.....	119
Figure 6.14	Pointer mode change.....	120
Figure 6.15	The DCE command menu.....	120
Figure 6.16	The DCE windows menu.....	120
Figure 6.17	Top level DCE window for a fax machine domain.....	121

Figure 6.18	Sub-domain separation.....	122
Figure 6.19	POP table definition using plan nets.....	122
Figure 6.20	Initial situation definition.....	123
Figure 6.21	Standard static and maintenance axiom definition.....	123
Figure 6.22	Testing the domain model.....	124
Figure 6.23	Building a task definition.....	125
Figure 6.24	Displaying the task hierarchy.....	126
Figure 6.25	Constructing a projection network.....	126
Figure 6.26	Path selection within a projection network.....	127
Figure 6.27	Displaying the SCRs.....	128
Figure 6.28	Plan net to POP table conversion.....	129
Figure 6.29	Graphical task to task model to STM.....	130
Figure 6.30	DCE save dialog.....	132
Figure 6.31	The GDI in TANDEM.....	134
Figure 6.32	The GDI session process flow.....	135
Figure 6.33	The initial GDI dialog for a Power Macintosh.....	135
Figure 6.34	Pre-tutoring session teaching strategy selection.....	136
Figure 6.35	GDI and domain interface separation.....	137
Figure 6.36	Insulating the GDI.....	138
Figure 6.37	Domain and platform layers in TANDEM.....	138
Figure 6.38	GDI and domain interaction example.....	139
Figure 6.39	GDI independence model.....	140
Figure 7.1	Keeping the GDI domain independent.....	144
Figure 7.2	GDI session manager dialog box.....	145
Figure 7.3	Teaching strategy selection.....	145
Figure 7.4	Teaching strategy summary.....	146
Figure 7.5	Default text based interface.....	146
Figure 7.6	Panafax UF-V40 fax machine.....	148
Figure 7.7	Fax domain definition example.....	148
Figure 7.8	Task hierarchy for <i>Cleaning the scanning area</i>	149
Figure 7.9	Fax machine domain dialogue windows.....	149
Figure 7.10	Fax machine domain session example.....	150
Figure 7.11	Car maintenance domain definition example.....	151
Figure 7.12	Basic task hierarchy for the car domain.....	152
Figure 7.13	Car maintenance interface.....	153
Figure 7.14	Car jack <i>used</i> while under the car.....	153
Figure 7.15	Interface to GDI code.....	154
Figure 7.16a	Car maintenance domain example session.....	155
Figure 7.16b	Car maintenance domain example session.....	156

Figure 7.17	Radio studio domain model example 1.....	158
Figure 7.18	Radio studio domain model example 2.....	158
Figure 7.19	Radio studio interface.....	159
Figure 7.20	Changing the on-air volumes.....	159
Figure 7.21	Selecting a help command.....	160
Figure 7.22	Playing a cart.....	161
Figure 7.23	Starting an empty turntable.....	161
Figure 7.24	Playing a record.....	162
Figure A.1	Plotting nodes around a circle.....	198
Figure A.2	Labelling nodes around a circle.....	198
Figure A.3	First level projection graph.....	199
Figure A.4	Final projection graph.....	200
Figure A.5	Nodes of a projection graph.....	201
Figure A.6	Selecting an appropriate node.....	201
Figure A.7	The final projection graph.....	202
Figure A.8	The plan net projection algorithm.....	202
Figure A.9	TANDEMs projection algorithm.....	204
Figure B.1	Providing GDI domain independence.....	205
Figure B.2	DCE generated files.....	206
Figure B.3	Car domain layout example.....	207
Figure B.4	Layout file code example.....	207
Figure B.5	Selecting the QuickSave option.....	208
Figure B.6	POP table code example.....	208
Figure B.7	Displayed tasks example.....	209
Figure B.8	Displayed SCRs example.....	209
Figure B.9	Example task file.....	210
Figure B.10	Saving custom interface code.....	210
Figure B.11	Default interface code.....	211
Figure B.12	DCE generated interface code.....	211
Figure B.13	Car domain custom interface code.....	212
Figure C.1	Platform and domain drivers.....	213
Figure C.2	GDI control unit code.....	214
Figure C.3	Platform driver code.....	215
Figure C.4	Power Macintosh platform driver.....	215
Figure C.5	Domain driver code.....	216
Figure C.6	Power Macintosh text interface A.....	217
Figure C.7	Power Macintosh text interface B.....	218
Figure D.1	Initial graph examples.....	220
Figure D.2	Adding an independent transition.....	220

Figure D.3	The original graphs.....	221
Figure D.4	Duplication of DE.....	221
Figure D.5	The preserved transitions from G_1	222
Figure D.6	Graph of part of a VCR domain.....	225
Figure D.7	Comparison of representations.....	226