

# ARTIFICIAL INTELLIGENCE AND HUMAN LEARNING

*Intelligent computer-aided instruction*

Edited by

John Self

Technische Hochschule Darmstadt  
Fachbereich 3  
Institut für Psychologie  
Steubenplatz 12, 6100 Darmstadt

Inv.-Nr. 9108740

LONDON NEW YORK

Chapman and Hall

# Contents

<b>Contributors</b>	<b>xi</b>
<b>Preface</b>	<b>xiv</b>
 <b>PART 1 THEORETICAL ISSUES</b>	
<b>1 Representing complex knowledge in an intelligent machine tutor</b>	<b>3</b>
<i>Beverly P. Woolf</i>	
1.1 Introduction	3
1.2 Knowledge of the domain	4
1.3 Knowledge of teaching	14
1.4 Evaluations	25
1.5 Conclusions	26
 <b>2 Of course ICAI is impossible . . . worse though, it might be seditious</b>	 <b>28</b>
<i>Jim Ridgway</i>	
2.1 Introduction	29
2.2 Problems for ICAI in mathematical education	30
2.3 The needs of mathematical education	41
2.4 Computers in education?	43
2.5 Can ICAI be seditious?	45
2.6 Suggestion: transparent ICAI systems	46
 <b>3 The role of qualitative models in instruction</b>	 <b>49</b>
<i>William J. Clancey</i>	
3.1 What are qualitative models?	49
3.2 Situation-specific models	50
3.3 Simulation or executable models	55
3.4 Representation requirements	56
3.5 Use of models: the central role of diagnosis	60
3.6 Student model assessments	64
3.7 Conclusions	67

<b>4</b>	<b>Methods and mental models in theories of cognitive skill</b>	<b>69</b>
	<i>Stephen J. Payne</i>	
4.1	Introduction	69
4.2	Computational accounts of cognitive skill acquisition: skill as method	70
4.3	Weaknesses in the skill-as-method theories	73
4.4	Mental models	77
4.5	Integrating methods and mental models	79
<b>5</b>	<b>The requirements of conceptual modelling systems</b>	<b>88</b>
	<i>Stefano A. Cerri</i>	
5.1	Introduction	88
5.2	Common mistakes in knowledge engineering	89
5.3	Programming as translating	95
5.4	Conceptual modelling systems	102
5.5	About the architecture of conceptual modelling systems	103
5.6	Conclusions	107
<b>6</b>	<b>The appraisal of an ICAI system</b>	<b>109</b>
	<i>Lindsey Ford</i>	
6.1	Introduction	109
6.2	Overview of system	110
6.3	Appraisal of behaviour	114
6.4	How well have the research aims been met?	119
6.5	Conclusions	122
<b>7</b>	<b>Methods for evaluating micro-theory systems</b>	<b>124</b>
	<i>Tony Priest and Richard M. Young</i>	
7.1	What is a micro-theory system?	124
7.2	Micro-theory systems and ICAI	125
7.3	Micro-theory systems and statistics	126
7.4	Criteria for a desirable measure	127
7.5	Measures for evaluating micro-theories	129
7.6	Summary of the four methods	135
7.7	Applications and conclusions	136
 <b>PART 2 ICAI TECHNIQUES</b>		
<b>8</b>	<b>The role of the episodic memory in an intelligent tutoring system</b>	<b>141</b>
	<i>Gerhard Weber, Gerd Waloszek and Karl Friedrich Wender</i>	
8.1	Some constraints on intelligent tutoring	141
8.2	Levels of programming knowledge	143
8.3	Episodic memory and the role of generalization	144
8.4	An example	146
8.5	Failure-driven learning	153

<b>9</b>	<b>A resolution based method for discovering students' misconceptions</b>	<b>156</b>
	<i>Ernesto Costa, Sylvie Duchénoy and Yves Kodratoff</i>	
9.1	Introduction	156
9.2	Working hypothesis	158
9.3	An example	160
9.4	Conclusion	164
<b>10</b>	<b>Guided discovery tutoring and bounded user modelling</b>	<b>165</b>
	<i>Mark Elsom-Cook</i>	
10.1	Introduction	165
10.2	What is the nature of 'guidance' in tutoring?	168
10.3	How can appropriate guidance be offered?	171
10.4	Conclusions	178
<b>11</b>	<b>The application of machine learning to intelligent tutoring systems</b>	<b>179</b>
	<i>David Gilmore and John Self</i>	
11.1	Introduction	179
11.2	Student modelling methods	180
11.3	Current uses of machine learning in tutoring systems	181
11.4	Concept learning systems	186
11.5	An alternative student model	191
11.6	Conclusions	195
<b>12</b>	<b>Self-organized learning within an intelligent teaching system</b>	<b>197</b>
	<i>Rosemary R. Todd</i>	
12.1	Introduction	197
12.2	An intelligent teaching system	197
12.3	Domain	199
12.4	Self-organized learning	200
12.5	ITSSOL: an idealized version	201
12.6	ITSSOL: current version	205
12.7	The research strategy: a preliminary study	208
12.8	Summary	210
<b>13</b>	<b>Plan recognition and intelligent tutoring systems</b>	<b>212</b>
	<i>Mark R. Woodroffe</i>	
13.1	Introduction	212
13.2	Literature review	213
13.3	A detailed description of FITS-2	216
<b>14</b>	<b>Understanding user behaviour in command-driven systems</b>	<b>226</b>
	<i>John Jones, Mark Millington and Peter Ross</i>	
14.1	Introduction	226

14.2	Existing techniques for plan recognition	227
14.3	Why recognize plans and goals?	229
14.4	User modelling in command-driven systems	230
14.5	Conclusion	235
<b>15</b>	<b>SCALD—towards an intelligent authoring system</b>	<b>236</b>
	<i>Rod Nicolson</i>	
15.1	Introduction	236
15.2	The design for an IAS	237
15.3	The SCALD-1 prototype IAS	240
15.4	Current status and projected developments of the SCALD project	248
15.5	Comparison of the IAS cycle with the 'standard' program development cycle	251
15.6	Summary	253
	<b>PART 3 ICAI APPLICATIONS</b>	
<b>16</b>	<b>Design choices for an intelligent arithmetic tutor</b>	<b>257</b>
	<i>Tim O'Shea, Rick Evertsz, Sara Hennessy, Ann Floyd, Mike Fox and Mark Elsom-Cook</i>	
16.1	Introduction	257
16.2	Background to project	258
16.3	Children's conceptions of arithmetic	259
16.4	Modelling the student	263
16.5	Representation techniques for modelling	265
16.6	Methodology	267
16.7	Diagnostic consultants	268
16.8	Mathematical microworlds	270
16.9	Supporting formal and informal arithmetic	272
16.10	Conclusions	274
<b>17</b>	<b>A framework for the design of a writer's assistant</b>	<b>276</b>
	<i>Mike Sharples and Claire O'Malley</i>	
17.1	Introduction	276
17.2	Writing skills	278
17.3	Assistance for the writer	281
17.4	Existing support and teaching systems	283
17.5	Framework for a writer's assistant	283
17.6	Conclusions	290
<b>18</b>	<b>Modelling the students' errors in the ELECTRE tutor</b>	<b>291</b>
	<i>Michel Caillot</i>	
18.1	Introduction	291

18.2	Outline of the ELECTRE system	292
18.3	Modelling students' knowledge	293
18.4	Conclusion	298
<b>19</b>	<b>Interfaces that adapt to the user</b>	<b>300</b>
	<i>Martin Cooper</i>	
19.1	Introduction	300
19.2	Self-adaptive user interfaces	301
19.3	An adaptive interface for electronic mail	303
19.4	Discussion	307
19.5	Conclusions and further work	308
<b>20</b>	<b>Coaching in help systems</b>	<b>310</b>
	<i>Joost Breuker</i>	
20.1	Introduction	310
20.2	A help system for UNIX-Mail	311
20.3	Functions of help systems and the IPS world	317
20.4	Teaching expertise: strategies and tactics	325
<b>21</b>	<b>Question answering and explanation giving in on-line help systems</b>	<b>338</b>
	<i>J. Roger Hartley and Michael Smith</i>	
21.1	Introduction: some requirements of an intelligent help system	338
21.2	Some issues in explanation giving	340
21.3	The classification of users' activities and questions	343
21.4	The interpretation of users' questions	346
21.5	Question answering	350
21.6	Concluding comments	358
<b>22</b>	<b>A Pascal program checker</b>	<b>361</b>
	<i>Mark Elsom-Cook and Benedict du Boulay</i>	
22.1	Introduction	361
22.2	Overall goals of the program checker	362
22.3	A stand-alone syntax error checker	363
22.4	Future development	373
<b>23</b>	<b>Modelling programmers' intentions</b>	<b>374</b>
	<i>W. Lewis Johnson</i>	
23.1	Introduction	374
23.2	An overview of PROUST's intention-based analysis	375
23.3	Empirical evaluation of PROUST	383
23.4	Current and future directions	386

<b>24</b>	<b>Bridge: an intelligent tutor for thinking about programming</b>	<b>391</b>
	<i>Jeffrey Bonar and Robert Cunningham</i>	
24.1	Introduction	391
24.2	Design issues for an intelligent programming tutor	392
24.3	An overview of the Bridge system	394
24.4	Using Bridge with students	408
24.5	Future plans	408
	<b>References</b>	<b>410</b>
	<b>Index</b>	<b>429</b>