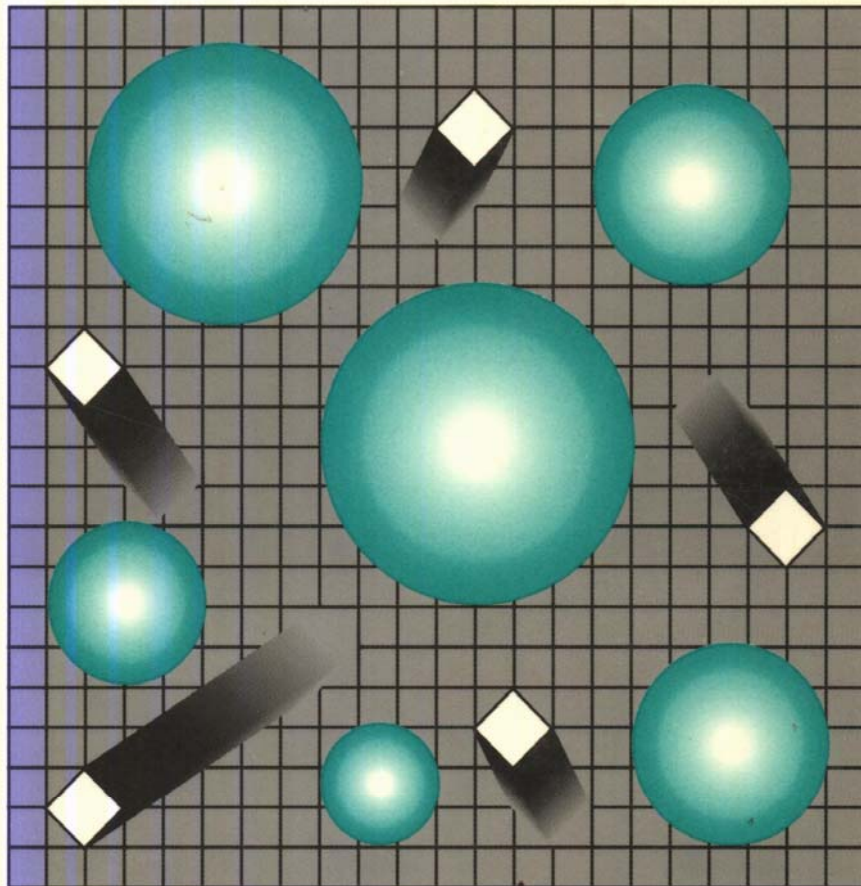


ARTIFICIAL INTELLIGENCE IN ENGINEERING

Edited by Graham Winstanley



Contents

List of Contributors	xiii
List of Trademarks	xiv
Introduction	xv
1: The Engineering Domain and Systems <i>G. Winstanley</i>	1
1.1 Introduction	1
1.2 The Systems Philosophy	2
1.3 Systems Classification	6
1.4 Models	9
1.4.1 Steady-state, Deterministic	10
1.4.2 Steady-state, Non-deterministic	11
1.4.3 Dynamic, Deterministic	11
1.4.4 Dynamic, Non-deterministic	13
1.4.5 Control Models	13
1.4.6 Conceptual Models	15
1.4.7 An Approach to the Building of Models	17
1.5 Methodology	19
1.5.1 Systems Engineering Methodologies	21
1.6 Lifecycle Concepts	23
1.6.1 Software Systems Engineering	24
1.7 AI and Systems	26
1.8 Conclusions	29
References	31
2: Representation and Reasoning <i>F.N. Teskey</i>	33
2.1 Introduction to Domain Models	33
2.2 Relational Data Models	35
2.2.1 Relational Theory	35
2.2.2 Limitations of Relational Models	36
2.2.3 Binary Relations	37
2.3 Semantic Network Models	39
2.3.1 Overview of Semantic Networks	39
2.3.2 Other Frame Relations	41
2.3.3 Procedural Attachments	41
2.3.4 Object-oriented Programming	43
2.3.5 Some Limitations of Semantic Nets	43
2.4 Using Logic for Representation	44
2.4.1 Types of Logic	44
2.4.2 Propositional Logic	44
2.4.3 Predicate Logic	46
2.4.4 Prolog	46

2.5	Searching World Models	48
2.5.1	Pattern Matching	48
2.5.2	Depth or Breadth First	49
2.5.3	Forward and Backward Chaining	50
2.6	Using Logic for Reasoning	50
2.6.1	Automated Theorem-proving	51
2.6.2	Conjunctive Normal Form	52
2.6.3	Resolution	52
2.7	Probabilistic Reasoning	53
2.7.1	Introduction to Probabilistic Reasoning	53
2.7.2	Inference Nets	54
2.7.3	Conditional Probabilities	54
2.7.4	Logical Sufficiency and Logical Necessity	55
2.7.5	Certainty Factors	56
2.8	An Example of a Knowledge Representation Environment	57
2.8.1	Knowledge Representation	57
2.8.2	Knowledge Acquisition	58
2.8.3	Applications	60
2.8.4	Conclusions	60
2.9	The Problem of Common Sense	61
	References	62
3:	Knowledge-based System Shells and Languages <i>G. Winstanley and H.A. Heathfield</i>	65
3.1	Introduction	65
3.2	Expert System Development Toolkits: Environments and Shells	66
3.2.1	KEE	68
3.2.2	Inference ART	71
3.2.3	Goldworks	73
3.2.4	KES	79
3.2.4.1	KES PS	80
3.2.4.2	KES HT	85
3.2.4.3	KES BAYES	87
3.2.5	Crystal	88
3.3	Symbolic Languages	91
3.3.1	Lisp	91
3.3.2	Prolog	99
3.3.3	The Object-oriented Paradigm	104
3.3.3.1	Object-based Modular Structure	104
3.3.3.2	Data Abstraction	105
3.3.3.3	Data Representation	105
3.3.3.4	Inheritance	106
3.3.3.5	Polymorphism	106
3.3.3.6	Automatic Memory Management	106
3.3.3.7	Object-oriented Languages	107
3.3.4	C++	107
3.3.4.1	Objects	108
3.3.4.2	Data Abstraction and Classes	108
3.3.4.3	Inheritance	109
3.3.4.4	Messages and Polymorphism	110
3.3.4.5	Instance Initialisation and Memory Management	111
3.3.4.6	Conclusion	112
3.3.5	Smalltalk	113
3.3.5.1	Data Abstraction and Classes	113
3.3.5.2	Objects	114
3.3.5.3	Messages	114

3.3.5.4	Inheritance	115
3.3.5.5	Memory Management	116
3.3.5.6	The Smalltalk Environment	116
3.3.5.7	Conclusion	116
	Bibliography	117
4:	Knowledge Engineering: Tools and Techniques <i>J.M. Kellett and G. Marshall</i>	119
4.1	Introduction	119
4.1.1	What is Knowledge Engineering?	119
4.1.2	What is Knowledge? Where is it Kept?	121
4.1.3	The Expert and the Knowledge Engineer	123
4.1.4	Knowledge-based Systems and Prototyping	124
4.2	Phases in Knowledge Acquisition	126
4.2.1	Groundwork—Orientation	127
4.2.2	Direct Approaches	128
4.2.2.1	Interview Techniques	128
4.2.2.2	Tutorial Interview	130
4.2.2.3	Teachback Interview	130
4.2.2.4	Introspection	131
4.2.3	Observational Approaches	131
4.2.3.1	Verbal Protocol Analysis	131
4.2.3.2	Dialogues	132
4.2.4	Indirect Approaches	133
4.2.4.1	Repertory Grid Analysis	133
4.2.4.2	Card Sorting	135
4.2.5	Machine-based Approaches	136
4.2.5.1	Machine Induction	136
4.2.5.2	Knowledge Engineering Tools	138
4.2.5.3	TEIRESIAS	138
4.2.5.4	ROGET	140
4.2.5.5	ETS—Expertise Transfer System	140
4.2.5.6	KADS—Knowledge Acquisition and Document Structuring	142
4.2.5.7	KEATS—the Knowledge Engineer's Assistant	143
4.2.5.8	RBFS—the Rule-Based Frame System	144
4.3	Concluding Remarks	146
	References	147
5:	Engineering Design, Manufacture and Test <i>T. Katz</i>	151
5.1	Introduction	151
5.1.1	Introduction to Design, Manufacture and Test	152
5.1.1.1	Design	152
5.1.1.2	Manufacture	154
5.1.1.3	Test and Inspection	154
5.1.2	The Interdisciplinary Nature of Engineering	155
5.2	System Design Approaches	155
5.2.1	Choice of Expert System Type	156
5.2.2	Knowledge Representation	156
5.2.3	Modelling	157
5.2.3.1	Structural and Functional Modelling	157
5.2.3.2	Hierarchies	158
5.2.3.3	Multiple Levels	159
5.2.4	Operational Considerations	160
5.2.4.1	Activity Confinement	162
5.2.4.2	Truth Maintenance	163

5.3	Design	163
5.3.1	Handling Complexity	164
5.3.2	AI-assisted Design	164
5.4	Manufacture	165
5.5	Product Test	166
5.5.1	Introduction to Test and Test Equipment	166
5.5.2	The ATPG—A First Attempt at Solution	167
5.5.2.1	Design for Test (DFT)	168
5.5.2.2	Built-in Self-test (BIST)	168
5.5.3	AI in Digital Test	168
5.5.3.1	Testable Design Expert System (TDES)	169
5.5.3.2	Alvey KB System for High-level Built-in Self-test	169
5.5.3.3	Sequential Circuit Test Search Scheme (SCIRTSS)	169
5.5.3.4	Automatic Programming Approach to Testing	170
5.5.3.5	Hitest	170
5.5.3.6	Multi-level Deep-shallow Integration	171
5.5.3.7	Proof Workbench	172
5.5.4	System Overview	173
5.5.4.1	Test Engineering Techniques in the Functional Test Domain	173
5.5.5	The Test-scheduling Expert System	175
5.5.5.1	Circuit Model	175
5.5.5.2	Register Transfer Logic Rule Base and Circuit Search	176
5.5.5.3	The Search	177
5.5.5.4	Generation of High-level Structure	178
5.5.5.5	Knowledge Base for Test Identification	180
5.5.5.6	Knowledge and Heuristics for Test Schedule	183
5.5.6	The Test Vector Generator and Timing Verifier	183
5.5.6.1	Test Definition	184
5.5.6.2	The Merging and Conflict Resolution Algorithm	184
5.5.7	Test Planner	185
5.5.7.1	Test Planner Function	186
5.5.7.2	Test Data Manipulation	186
5.5.8	Conclusions	188
5.6	Computer Integrated Manufacture—CIM	189
5.6.1	An AI-controlled CIM Model	189
5.7	Factors that Need Consideration before Developing Applications	190
5.7.1	The Future	191
	Bibliography	191
6:	Planning and Project Management <i>P.W. Kuczora</i>	195
6.1	Introduction	195
6.1.1	Planning and AI	195
6.1.2	Project Management	197
6.2	Pure AI Techniques for Planning Systems	199
6.2.1	Strategic and Tactical Planners	199
6.2.2	Traditional (Strategic) AI Planners	199
6.2.2.1	Planning as Search	199
6.2.2.2	Pruning the Search Space	200
6.2.2.3	State Space Search and Means-end Analysis	201
6.2.3	Linear and Non-linear Planners	202
6.2.3.1	Interacting Subproblems, Conjunctive Goals and Constraints	202
6.2.3.2	STRIPS and Linear Planning	203
6.2.3.3	Interaction Detection and Correction in Linear Planners	204
6.2.3.4	Hierarchical and Non-linear Planners	205
6.2.3.5	Hierarchical Search Control and Interaction Detection	207
6.2.4	Backtracking and Plan Repair	208
6.2.4.1	Non-monotonic Reasoning	208

6.2.4.2	Beam Search	210
6.2.4.3	Depth-first (Chronological) Backtracking	210
6.2.4.4	Dependency-directed Search and Truth Maintenance	210
6.2.4.5	Multiple Belief Spaces	211
6.2.5	Script-based and Case-based Planning	212
6.2.6	Opportunistic Planners	213
6.2.7	Mixed-AI-planners	214
6.2.8	Meta-planning	214
6.3	Knowledge-based 'AI-levered' and Non-AI Systems	216
6.3.1	Domain-specific Expert Systems	217
6.3.2	Hybrid 'AI-levered' Project Management Systems	218
6.3.2.1	The PLATFORM Experiments at Stanford	218
6.3.2.2	Project Management at Brighton—PIPPA and XPERT	222
6.3.3	Non-AI-planning in Uncertain Domains	225
6.4	Developments in Engineering Project Management	228
6.4.1	Lessons Learned from PLATFORM	228
6.4.2	Open Architectures and Graphical Interfaces	229
6.4.3	Integrated Applications for Project Management	231
6.5	Appendix: A History of AI-planning Systems	234
6.5.1	Graph Traverser	234
6.5.2	General Problem Solver (GPS)	235
6.5.3	STRIPS	235
6.5.4	ABSTRIPS	236
6.5.5	HACKER	236
6.5.6	INTERPLAN	236
6.5.7	NOAH	236
6.5.8	NONLIN	237
6.5.9	MOLGEN (Friedland, 1979)	237
6.5.10	MOLGEN (Stefik, 1981)	237
6.5.11	SIPE	238
6.5.12	DEVISER	238
6.5.13	BB1	238
6.5.14	O-PLAN	239
6.5.15	COMTRAC-O	239
6.5.16	Knowledge Intensive Planner—KIP	239
6.5.17	PLANEX	239
	References	240
7:	Computer Vision in Industry <i>R. Thomas</i>	245
7.1	Industrial Constraints and Vision System Performance	245
7.1.1	Introduction and Overview	245
7.1.2	CV as an Industrial Discipline	250
7.1.2.1	Existing Industrial CV	251
7.1.3	Performance Criteria	253
7.2	Vision Systems and Systems Engineering	255
7.2.1	Industrial CV Taxonomy	255
7.2.1.1	Visual Inspection	255
7.2.1.2	Robotic Guidance	256
7.2.2	A Systemic Approach	256
7.3	Visual Inspection	260
7.3.1	A Two-dimensional Perspective	260
7.3.1.1	Scene Constraints	260
7.3.1.2	Image Acquisition	261
7.3.1.3	Image Preprocessing	262
7.3.1.4	Segmentation	266
7.3.1.5	Feature Extraction and Classification	268
7.3.2	Visual Inspection: Measurement and Inspection Applications	269

7.3.2.1	Dimensional Inspection	269
7.3.2.2	Surface Inspection	274
7.3.2.3	Electronic Circuit and Layout Inspection	276
7.4	Robotic Guidance	279
7.4.1	Three-dimensional Representations: Model-based Recognition	279
7.4.1.1	Image Representation	279
7.4.1.2	Two-dimensional Modelling and Matching	280
7.4.1.3	Three-dimensional Modelling and Matching	281
7.4.1.4	Three-dimensional Modelling and Matching to Two-dimensional Image Data	281
7.4.2	Visual Guidance Applications	285
7.4.2.1	Manufacturing	285
7.4.2.2	Electronic Assembly	288
7.4.2.3	Robot Bin-picking	289
7.5	Summary	290
	Bibliography	290
8:	Developments in Artificial Intelligence Reasoning <i>A.B. Hunter</i>	295
8.1	Introduction	295
8.2	Uncertainty Reasoning	297
8.2.1	Fuzzy and Many-valued Logic	300
8.2.2	Qualitative Representation of Uncertainty in Prolog	302
8.2.3	Modal Logic	304
8.2.4	Discussion	306
8.3	Default and Non-monotonic Reasoning	307
8.3.1	Default Reasoning	308
8.3.2	Non-monotonic Logic	309
8.3.3	Circumscription	309
8.3.4	The Closed World Assumption and Negation-as-failure	310
8.3.5	Discussion	312
8.4	Temporal Reasoning	312
8.4.1	Tense Logic	312
8.4.2	Allen's Temporal Logic	316
8.4.3	McDermott's Temporal Logic	318
8.4.4	Shoham's Temporal Logic	319
8.4.5	Event Calculus	320
8.4.6	Discussion	322
8.5	Spatial Reasoning	323
8.5.1	Situation Calculus in the 'Blocks-world'	323
8.5.2	SPACES	325
8.5.3	Discussion	326
8.6	Causal and Action Reasoning	326
8.6.1	Action and Dynamic Logics	328
8.6.2	Modal Action Logic	329
8.6.3	Naive Physics	331
8.6.4	Qualitative Reasoning	332
8.6.5	Discussion	333
8.7	Conclusions	333
	References	
9:	Mathematical Foundations of Artificial Intelligence <i>D.K. Bose and S.W. Ellacott</i>	337
9.1	Introduction	337
9.2	Theory of Sets	337

9.2.1	Collections of Objects	337
9.2.2	Set Notation	338
9.2.3	Set Manufacture	338
9.2.4	Order	341
9.2.5	Cartesian Products	342
9.3	Relations	343
9.3.1	Order Relations	343
9.3.2	Equivalence Relations	344
9.3.3	Converse Relations and 01-matrices	344
9.4	Functions	345
9.5	Graph Theory	345
9.5.1	Graphs and Problem Representation	346
9.5.1.1	The Decanting Problem	346
9.5.1.2	Instant Insanity	348
9.5.1.3	Representation of Computer Programs	350
9.5.1.4	Graphs in Pattern Recognition	350
9.5.1.5	Expert Systems and Circular Arguments	350
9.5.1.6	'is_a' Graphs and Non-monotonicity	351
9.5.2	Fundamentals of Graph Theory	352
9.5.2.1	New Graphs from Old	352
9.5.2.2	Computer Representations of Graphs	354
9.5.2.3	Paths and Components	355
9.5.2.4	Trees	356
9.5.2.5	Fundamental Cycles	358
9.5.2.6	An Algorithm for Spanning Trees and Fundamental Cycles	361
9.5.2.7	Other Aspects of Graph Theory	363
9.6	The Mathematics of Reasoning	363
9.6.1	Formal Reasoning and Mathematical Logic	363
9.6.1.1	Production Rules and Implication	364
9.6.1.2	Logical Negation and the Contrapositive	364
9.6.1.3	Truth Tables	365
9.6.1.4	The \wedge and \vee Operators	366
9.6.2	Predicate Calculus	368
9.6.2.1	Variables and Quantifiers	368
9.6.2.2	Relations and Clauses	370
9.6.3	Reasoning with Uncertainty	370
9.6.3.1	Fuzzy Reasoning	370
9.6.3.2	Probability Theory and the Laws of Probability	373
9.6.3.2.1	Conditional probability and Bayes' theorem	375
9.6.3.3	The Pragmatic Compromise	377
	Bibliography	377
10:	Computer Architectures for Artificial Intelligence <i>N.J. Avis</i>	379
10.1	Introduction	379
10.2	Sequential Implementations	380
10.2.1	The Semantic Gap	380
10.2.2	Procedural Programming Paradigm	381
10.2.3	Functional Programming Paradigm	381
10.2.4	Control Flow <i>versus</i> Functional Paradigm	382
10.2.5	Support for Functional Languages	383
10.2.5.1	Efficient Function Handling	383
10.2.5.2	Maintaining the Environment	384
10.2.5.3	Structured Memory	384
10.2.5.3.1	Self-defining objects	384
10.2.5.3.2	List representation and access	385
10.2.5.3.3	Heap maintenance	386
10.2.6	Lisp Platforms	388

10.2.6.1	General Purpose: Conventional von Neumann Machines	389
10.2.6.2	The Workstation	389
10.2.6.3	Personal Computers	389
10.2.6.4	Performance Issues	390
10.2.6.5	Lisp Machines	390
10.2.6.5.1	The Symbolics 3600	391
10.2.6.5.2	Other Lisp machines	392
10.2.6.6	Take the RISC	393
10.2.6.6.1	A RISC for Lisp	393
10.2.6.6.2	Other RISCs	394
10.3	Parallel Implementations	394
10.3.1	Parallel Processing	394
10.3.1.1	Processor Configurations	395
10.3.1.1.1	What is a hypercube?	396
10.3.1.2	Performance Considerations	398
10.3.1.3	Types of Parallel Processors	398
10.3.1.4	General-purpose Parallel Processors	399
10.3.1.4.1	Shared memory	399
10.3.1.4.2	Message passing	400
10.3.1.4.3	Shared memory <i>versus</i> message passing	401
10.3.1.4.4	The Transputer	401
10.3.1.4.5	Programming considerations	401
10.3.1.4.6	Commercial message-passing parallel processors	402
10.3.1.4.7	The Intel iPSC	402
10.3.1.5	Language-based Parallel Processors	406
10.3.1.5.1	Reduction computation	407
10.3.1.5.2	ALICE	408
10.3.1.6	Massively Parallel Computers	412
10.3.1.6.1	The Connection Machine	413
10.4	Future Directions—Connectionism	416
10.4.1	Connectionist Systems	418
	Bibliography	419
	Index	423