

Table of Contents

List of Tables.....	xvi
List of Figures.....	xix
Abstract.....	xxiv
Chapter 1 Introduction.....	1
Chapter 2 Approaches to Software Design.....	7
2.1 Software Design Methods.....	7
2.1.1 Objectives of Software Design Methods.....	8
2.1.2 Methods for Producing Sequential Designs.....	14
2.1.2.1 Structured Analysis and Structured Design.....	14
2.1.2.2 Real-Time Structured Analysis and Structured Design.....	15
2.1.3 Methods for Producing Concurrent Designs.....	16
2.1.3.1 Concurrent Designs From RTSA Models.....	16
2.1.3.2 Concurrent Designs Without RTSA Models.....	17
2.1.3.3 Concurrent Designs For Ada.....	18
2.1.4 Methods for Producing Object-Oriented Designs.....	19
2.1.5 Approaches to Design Scheduling.....	19
2.2 Automating Software Design Methods.....	20
2.2.1 Approaches to Automating Architectural Design.....	20
2.2.1.1 Cluster Analysis.....	21
2.2.1.2 Formal Rule Rewriting Systems.....	22
2.2.1.3 Artificial Intelligence Approaches.....	22
2.2.1.3.1 Rule-based Expert Systems.....	23
2.2.1.3.1.1 Specification-Transformation Expert System.....	23
2.2.1.3.1.2 System Architects Apprentice, Design Assistant.....	24

2.2.1.3.2 Other Artificial Intelligence Approaches.....	26
2.2.2 Approaches to Automating Detailed Design.....	27
2.3 Automatic Programming.....	27
2.4 Summary of Findings.....	29
2.4.1 Importance of Architectural Design.....	29
2.4.2 Benefits of Software Design Methods.....	30
2.4.3 Benefits from Automating Software Design Methods.....	30
Chapter 3 Overview of Research.....	32
3.1 Evaluation of Existing Approaches.....	32
3.2 Generating Concurrent Designs for Real-Time Software.....	37
3.2.1 The Designer's Objectives.....	37
3.2.2 A Software Design Method for Concurrent and Real-Time Systems.....	40
3.2.2.1 COBRA.....	41
3.2.2.2 CODARTS.....	43
3.2.2.2.1 Task Structuring.....	44
3.2.2.2.2 Task-Interface Definition.....	46
3.2.2.2.3 Module Structuring.....	47
3.2.2.2.4 Integrating Tasks and Modules.....	48
3.2.2.2.5 Configuration and Evaluation.....	49
3.3 The Research Problem.....	49
3.3.1 Modeling and Analyzing Specifications.....	50
3.3.2 Modeling Designs and Target Environments.....	52
3.3.3 Modeling Decision-Making Processes.....	52
3.4 A Knowledge-based Approach to Design Generation.....	54
3.4.1 A Conceptual View of the Approach.....	54
3.4.1.1 Design-Process Meta-Knowledge.....	59
3.4.1.2 Decision-Making Knowledge Bases and Design-Decision Rules.....	67
3.5 Proof-of-Concept Prototype.....	71
Chapter 4 A Meta-Model for Specifications.....	74

4.1 The Concept Hierarchy.....	76
4.1.1 Specification Elements.....	79
4.1.1.1 RTSA Syntactic Element Classification.....	80
4.1.1.2 Semantic Element Classification.....	83
4.1.1.2.1 Terminators.....	83
4.1.1.2.2 Solid Transformations.....	84
4.1.1.2.3 Device Interface Objects.....	85
4.1.1.2.4 Control Objects.....	86
4.1.1.2.5 Solid Directed Arcs.....	88
4.1.1.2.6 Dashed Directed Arcs.....	90
4.1.1.2.7 Functions.....	93
4.1.1.2.8 Solid Two-Way Arcs.....	96
4.1.2 Specification Addenda.....	96
4.1.2.1 Aggregation Groups.....	97
4.1.2.2 Exclusion Groups.....	98
4.1.2.3 Locked-State Events.....	100
4.2 Concept Axioms and Inheritance.....	101
4.3 Classification Rules and Concept Classification.....	104
4.4 Information Elicitation.....	110
Chapter 5 A Meta-Model for Concurrent Designs.....	112
5.1 A Diagrammatic View of the Design Meta-Model.....	113
5.2 Modeling Concurrent Designs.....	122
5.2.1 Concurrent Design Entities.....	123
5.2.2 Relationships Among Concurrent Design Entities.....	129
5.2.3 Assumptions Underlying the Design Meta-Model.....	135
5.3 Describing Target Environments.....	137
Chapter 6 Task Structuring.....	141
6.1 Identify Candidate Tasks.....	143
6.1.1 Rules for Identifying Input/Output Tasks.....	143

6.1.2	Rules for Identifying Internal Tasks.....	146
6.1.2.1	Internal Periodic Tasks.....	148
6.1.2.2	Internal Asynchronous Tasks.....	151
6.2	Allocate Remaining Transformations to Tasks.....	153
6.2.1	Allocating Transformations Based on Control Cohesion.....	153
6.2.2	Allocating Transformations Based on Sequential Cohesion.....	154
6.2.3	Allocating Transformations Based on Functional Cohesion.....	160
6.3	Consider Task Mergers.....	162
6.3.1	Rules for Combining Tasks Based on Mutual Exclusion.....	164
6.3.2	Rules for Combining Tasks Based on Sequential Cohesion.....	166
6.3.3	Rules for Combining Tasks Based on Temporal Cohesion.....	168
6.3.4	Rules for Combining Tasks Based on Task Inversion.....	173
6.3.5	Rationale for Preferred Rule Orderings.....	175
6.4	Consider Resource Monitors.....	177
6.5	Review Task Structure and Consider Renaming Tasks.....	179
Chapter 7	Task Interface Definition.....	181
7.1	Allocate External Task Interfaces.....	184
7.1.1	Rule to Allocate Timer Events.....	184
7.1.2	Rule to Allocate External Events.....	185
7.1.3	Rules for Allocating Data.....	186
7.1.4	Rule to Identify Inter-Task Exchanges.....	187
7.2	Allocate Control and Event Flows.....	188
7.2.1	Rules for Mapping to Software Events.....	189
7.2.2	Rules for Mapping to Tightly-Coupled Messages.....	193
7.2.3	Rule for Mapping to Queued Messages.....	194
7.2.4	Rule for Mapping Ambiguous Control and Event Flows.....	195
7.2.5	Rules for Mapping to Existing Messages.....	198
7.3	Allocate Data Flows.....	200
7.3.1	Rules for Mapping to Tightly-Coupled Messages.....	201

7.3.2	Rules for Mapping to Queued Messages.....	203
7.3.3	Rule for Mapping Ambiguous Data Flows.....	206
7.3.4	Rules for Mapping to Existing Messages.....	208
7.3.5	Rule for Symmetric Message Interfaces.....	212
7.3.6	Rule for Ambiguous Interface to Input/Output Devices.....	214
7.4	Elicit Message Priorities.....	216
7.5	Allocate Queue Interfaces.....	217
7.5.1	Rules for Single-Priority Message Queues.....	218
7.5.2	Rules for Multiple-Priority Message Queues.....	221
7.6	Consider Renaming Task Interface Design Elements.....	224
Chapter 8 Module Structuring.....		225
8.1	Identify Candidate Modules.....	227
8.1.1	Rule for Identifying Device-Interface Modules.....	228
8.1.2	Rules for Identifying Other Interface Modules.....	228
8.1.3	Rule for Identifying State-Transition Modules.....	230
8.1.4	Rule for Identifying Data-Abstraction Modules.....	231
8.1.5	Rules for Identifying State-Driven Modules.....	232
8.2	Allocate Functions to Data-Abstraction Modules.....	235
8.3	Allocate Remaining Transformations to Modules.....	240
8.4	Allocate Isolated Elements to Modules.....	243
8.4.1	Rules to Allocate Isolated Data Stores.....	244
8.4.2	Rules to Allocate Isolated Functions.....	246
8.5	Consider Module Subsumption.....	247
8.6	Determine Module Operations.....	249
8.6.1	Rule to Allocate Arcs Internal to Modules.....	251
8.6.2	Rules to Determine Operations for Interface Modules.....	252
8.6.3	Rule to Determine Operations for State-Transition Modules.....	263
8.6.4	Rules to Determine Operations for Direct Access to Data Stores.....	265
8.6.5	Rules to Determine Operations from Functions.....	268

8.7 Review Module Structure and Consider Renaming Modules.....	277
Chapter 9 Task and Module Integration.....	278
9.1 Determine Module Placements.....	280
9.1.1 Captive Modules.....	281
9.1.1.1 Rules for Placing DIMs for Asynchronous and Periodic Devices..	281
9.1.1.2 Rule for Placing State-Transition Modules.....	283
9.1.1.3 Rules for Placing User and Subsystem Interface Modules.....	284
9.1.2 Shareable Modules.....	285
9.1.2.1 Rule for Placing Data-Abstraction Modules.....	287
9.1.2.2 Rules for Placing Remaining Information-Hiding Modules.....	289
9.2 Link Tasks and External Modules.....	294
9.3 Link External Modules.....	299
Chapter 10 A Prototype COncurrent Designer's Assistant.....	304
10.1 Software Architecture for CODA.....	305
10.2 Knowledge Representation.....	308
10.2.1 Specification Meta-Model.....	309
10.2.2 Design Meta-Model.....	311
10.2.3 Decision-Making Processes and Design-Decision Rules.....	313
10.2.4 Meta-Knowledge.....	314
10.3 CODA's Components.....	316
10.3.1 Command & Query Processor.....	316
10.3.2 Specification Library and Specification Analyzer.....	316
10.3.3 Concept Classifier and Classification Rules.....	317
10.3.4 Design Library and Design Generator.....	317
10.3.5 Target Environment Description Library and Loader.....	318
10.3.6 Hardware and Software Requirements for the Prototype.....	318
10.4 User's View of CODA.....	318
10.4.1 CODA Viewed by a Novice Designer.....	319
10.4.2 CODA Viewed by an Experienced Designer.....	321

10.4.2.1 CODA Commands.....	322
10.4.2.1.1 CODA Enforces Process Constraints.....	322
10.4.2.1.2 CODA Manages Libraries.....	324
10.4.2.1.3 CODA Analyzes Specifications.....	325
10.4.2.1.4 CODA Generates Designs.....	326
10.4.2.1.5 Unimplemented Commands.....	326
10.4.2.2 CODA Queries.....	327
10.4.2.2.1 General Information Queries.....	327
10.4.2.2.2 Querying Design Elements.....	327
10.4.2.2.3 Querying Design Histories.....	329
10.4.2.2.4 Querying Design Relationships.....	329
Chapter 11 Evaluation.....	331
11.1 Summary Evaluation.....	331
11.1.1 Evaluation Against Research Objectives.....	332
11.1.2 Comparison Against Other Approaches.....	334
11.1.3 Strengths and Weaknesses of the Proposed Approach.....	337
11.2 Analysis of the Case Studies.....	339
11.2.1 Modeling and Analysis of Specifications.....	339
11.2.1.1 Semantic Interpretation of Flow Diagrams.....	339
11.2.1.1.1 Problems Classifying Data Flows.....	342
11.2.1.1.2 Problems Classifying Functions.....	343
11.2.1.1.2.1 Tentative Classifications.....	344
11.2.1.1.2.2 Assisted Classifications.....	345
11.2.1.2 Extensions and Restrictions to RTSA.....	346
11.2.1.3 Eliciting Helpful Information.....	347
11.2.1.3.1 Eliciting Node Cardinality.....	348
11.2.1.3.2 Eliciting Locked-State Events.....	348
11.2.1.3.3 Eliciting Exclusion Groups.....	349
11.2.1.3.4 Eliciting Aggregation Groups.....	350

11.2.2	Generation and Representation of Designs.....	350
11.2.2.1	Task Structures.....	350
11.2.2.1.1	Candidate Tasks.....	351
11.2.2.1.2	Remaining Transformations.....	353
11.2.2.1.3	Task Mergers.....	356
11.2.2.1.4	Resource Monitors.....	357
11.2.2.2	Task Interfaces.....	360
11.2.2.2.1	External Task Interfaces.....	360
11.2.2.2.2	Control and Event Flows.....	361
11.2.2.2.3	Data Flows.....	363
11.2.2.2.4	Message Priorities and Queue Interfaces.....	364
11.2.2.3	Module Structures.....	365
11.2.2.3.1	Candidate Modules.....	365
11.2.2.3.2	Functions of Data-Abstraction Modules.....	367
11.2.2.3.3	Remaining Transformations.....	368
11.2.2.3.4	Isolated Elements.....	369
11.2.2.3.5	Module Subsumption.....	370
11.2.2.3.6	Module Operations.....	371
11.2.2.4	Task and Module Integration.....	373
11.2.2.5	Assessment of Automated Design Generation.....	373
11.3	Aspects of the Approach that Proved Difficult to Automate.....	376
11.3.1	Specification Analysis.....	376
11.3.1.1	Aperiodic Functions.....	376
11.3.1.2	Triggered Functions Receiving Data Flows.....	377
11.3.1.3	Stimulus versus Response.....	378
11.3.1.4	Classifying Terminators.....	379
11.3.1.5	Eliciting Required Information.....	379
11.3.1.6	Expressing Cardinality.....	380
11.3.1.7	Specification Addenda.....	380

11.3.1.7.1 Mutual Exclusion.....	381
11.3.1.7.2 Locked-State Events.....	381
11.3.1.7.3 Aggregation.....	382
11.3.2 Design Generation.....	382
11.3.2.1 Allocating Transformations to Existing Tasks and Modules.....	383
11.3.2.2 Combining Tasks with Compatible Periods.....	383
11.3.2.3 Combining Data-Abstraction Modules.....	383
11.3.2.4 Determining Synchronization Requirements.....	384
11.4 Quality of Generated Designs.....	385
11.4.1 Task Structure.....	386
11.4.2 Module Structure.....	387
11.4.3 Task and Module Integration.....	390
11.4.4 Task Interfaces.....	391
11.5 Interpretation of Results.....	393
Chapter 12 Contributions and Future Research.....	396
12.1 Contributions.....	396
12.2 Potential Applications.....	398
12.3 Future Research.....	400
References.....	403
Appendices.....	411
Appendix A. Axioms and Classification Rules for the Specification Meta-Model.....	412
A.1 Axioms for Semantic Concepts.....	413
A.1.1 Axioms for Specification Elements.....	414
A.1.2 Axioms for Specification Addenda.....	432
A.2 Rules for Classifying Semantic Concepts.....	434
A.2.1 Rules for Arc Classification.....	435
A.2.2 Rules for Transformation Classification.....	442
A.2.3 Rules for Stimulus-Response Classification.....	451

A.2.4 Rules for Ambiguous-Function Classification.....	453
Appendix B. Automobile Cruise Control and Monitoring System Case Study.....	456
B.1 Analyzing the Specification.....	459
B.1.1 Classifying the Specification.....	459
B.1.2 Eliciting Additional Information.....	461
B.1.3 Checking Classifications and Axioms.....	462
B.1.4 Annotated Data/Control Flow Diagrams.....	462
B.2 Generating the Design.....	475
B.2.1 Structuring Tasks.....	477
B.2.1.1 Identifying Candidate Tasks.....	477
B.2.1.2 Allocating Remaining Transformations.....	479
B.2.1.3 Considering Task Mergers.....	480
B.2.1.4 Completing Task Structuring.....	482
B.2.2 Structuring Modules.....	482
B.2.2.1 Identifying Candidate Modules.....	484
B.2.2.2 Allocating Functions to DAMs.....	486
B.2.2.3 Allocating Isolated Elements.....	487
B.2.2.4 Considering Module Subsumption.....	487
B.2.2.5 Completing Module Structuring.....	488
B.2.3 Integrating Tasks and Modules.....	490
B.2.4 Defining Task Interfaces.....	491
B.2.4.1 Allocating External Task Interfaces.....	491
B.2.4.2 Allocating Control and Event Flows.....	494
B.2.4.3 Allocating Data Flows.....	495
B.2.4.4 Eliciting Message Priorities and Defining Queue Interfaces.....	495
B.2.4.5 Completing Task-Interface Definition.....	496
B.2.5 The Completed Design.....	496
B.2.5.1 Creating the Software Architecture Diagram.....	496
B.2.5.2 Assessing the Design.....	502

B.3 Design Generated for a Novice Designer.....	503
B.3.1 Generating the Design.....	504
B.3.2 The Completed Design.....	505
Appendix C. Robot Controller System Case Study.....	509
C.1 Robot Controller System, Version 1.....	509
C.1.1 Analyzing the Specification.....	511
C.1.1.1 Classifying Concepts in the Specification.....	511
C.1.1.2 Eliciting Additional Information and Verifying Concepts.....	512
C.1.1.3 Annotated Data/Control Flow Diagram.....	513
C.1.2 Generating the Design.....	516
C.1.2.1 Structuring Tasks.....	516
C.1.2.2 Structuring Modules.....	518
C.1.2.3 Integrating Tasks and Modules.....	520
C.1.2.4 Defining Task Interfaces.....	520
C.1.2.5 The Completed Design.....	524
C.1.3 A Design for Target Environments without Message Queues.....	526
C.2 Robot Controller System, Version 2.....	527
Appendix D. Elevator Control System Case Study.....	533
D.1 Elevator Control System for a Small Building.....	534
D.1.1 Analyzing the Specification.....	534
D.1.1.1 Classifying Concepts.....	534
D.1.1.2 Eliciting Additional Information and Verifying the Specification.....	536
D.1.1.3 Annotated Data/Control Flow Diagram.....	537
D.1.2 Generating the Design.....	543
D.1.2.1 Structuring Tasks.....	543
D.1.2.2 Structuring Modules.....	543
D.1.2.3 Integrating Tasks and Modules.....	546
D.1.2.4 Defining Task Interfaces.....	546
D.1.2.5 The Completed Design.....	546

D.1.3 Differentiating Queued Messages by Priority.....	552
D.1.3.1 Priority Message Queuing Services Unavailable.....	553
D.1.3.2 Priority Message Queuing Services Available.....	553
D.2 Elevator Control System for a Large Building.....	555
Appendix E. Remote Temperature Sensor Case Study.....	559
E.1 Analyzing the Specification.....	560
E.1.1 Evaluating the Original Data Flow Diagram.....	560
E.1.2 Correcting and Reevaluating the Data Flow Diagram.....	562
E.1.3 Eliciting Additional Information.....	566
E.2 Generating the Design.....	567
E.2.1 Structuring Tasks.....	567
E.2.2 Defining Task Interfaces.....	570
E.2.3 The Task Architecture.....	571
E.2.4 Structuring Modules.....	571
E.2.5 Integrating Tasks and Modules.....	573
E.2.6 The Completed Design.....	573
E.3 CODA Unaided.....	576
E.3.1 Analyzing the Specification.....	577
E.3.2 The Completed Design.....	577
Appendix F. Index Of Terms.....	580
F.1 Terms Defined Within The Specification Meta-Model.....	580
F.1.1 Specification Elements.....	580
F.1.2 Specification Addenda.....	584
F.2 Terms Defined Within The Design Meta-Model.....	584
F.2.1 Design Entities.....	584
F.2.2 Design Relationships.....	585
Curriculum Vitae.....	588