SPECIFICATION AND DESIGN METHODOLOGY FOR REAL-TIME EMBEDDED SYSTEMS

SPECIFICATION AND DESIGN METHODOLOGY FOR REAL-TIME EMBEDDED SYSTEMS

Randall S. Janka

Cadence Design Systems, Inc.
Georgia Tech Research Institute
Georgia Institute of Technology

SPRINGER SCIENCE+BUSINESS MEDIA, LLC

ISBN 978-1-4613-5263-1 ISBN 978-1-4615-0837-3 (eBook)
DOI 10.1007/978-1-4615-0837-3

Library of Congress Cataloging-in-Publication Data

Copyright © 2002 by Springer Science+Business Media New York
Originally published by Kluwer Academic Publishers in 2002
Softcover reprint of the hardcover 1st edition 2002

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, mechanical, photo-copying, recording, or otherwise, without the prior written permission of the publisher, Springer Science+Business Media, LLC.

Printed on acid-free paper.

*“But as for me, I trust in You, O LORD, I say, ‘You are my God.’
My times are in your hand....”
–Psalm 31:14-15a*

*To our awesome God
who engineered this wondrous creation
and our salvation through his son Jesus Christ;
And to my beloved wife
Beverly
and our three incredible children
Alexandra, Bethany, and Victoria*

Table of Contents

LIST OF FIGURES	XI
LIST OF TABLES	XV
LIST OF ACRONYMS	XIX
FOREWORD	XXIII
PREFACE	XXV
ACKNOWLEDGEMENTS	XXIX
1 INTRODUCTION	1
1.1 THE BASIC PROBLEM	1
1.2 A SOLUTION TO THE PROBLEM	4
1.3 CONTRIBUTIONS	5
1.4 ORGANIZATION	8
1.5 CONVERGENCE OF RESEARCH THREADS	9
2 PROBLEM BACKGROUND	11
2.1 BACKGROUND	11
2.2 THE DOMAIN-SPECIFIC PROBLEM	29
2.3 A DOMAIN-SPECIFIC SOLUTION	30
3 SYSTEM REQUIREMENTS & INTRINSIC SDM ASSESSMENT	33
3.1 SYSTEM REQUIREMENTS SPECIFICATION DOMAINS	33
3.2 DOMAIN-RELEVANT MODELS OF COMPUTATION	34
3.3 BEST MOCs FOR DOMAIN-SPECIFIC SPECIFICATION AXES	38
3.4 IMPLICIT FRAMEWORK MOCs	41
3.5 COMPARING THE MONOLITHIC FRAMEWORKS	43

4	QUANTIFIED EXTRINSIC SDM ASSESSMENT	45
4.1	A UNIFIED SPECIFICATION-MODELING METHODOLOGY EVALUATION FRAMEWORK	45
4.2	QUANTIFICATION OF SARKAR BASIS	51
4.3	USING QUANTIFIED BASIS TO CHARACTERIZE CASE SDM FRAMEWORKS	68
4.4	CONCLUSION	77
5	EXTENDING GAJSKI'S SER METHODOLOGY	79
5.1	BACKGROUND	79
5.2	PARALLELS BETWEEN GAJSKI'S SER & OUR ADOI	80
5.3	EXTENDING GAJSKI'S SER TO OUR ADOI	84
6	THE MAGIC SPECIFICATION & DESIGN METHODOLOGY	87
6.1	OVERVIEW OF THE MAGIC METHODOLOGY	87
6.2	ESTABLISHING MODEL CONTINUITY	89
6.3	"RULES"—THE STEPS OF THE MAGIC SDM	92
6.4	"TOOLS"—THE FRAMEWORKS INTEGRATED INTO THE MAGIC SDM	94
6.5	MODEL CONTINUITY VIA MIDDLEWARE	98
7	CASE STUDY: VALIDATING THE MAGIC SDM USING A SAR PROCESSOR APPLICATION	109
7.1	RASSP SAR BENCHMARK OVERVIEW	110
7.2	TABULATE REQUIREMENTS	114
7.3	CAPTURE NON-CONSTRAINT REQUIREMENTS IN AN EXECUTABLE MODEL	116
7.4	BUILD EXECUTABLE WORKBOOK WITH REQUIREMENTS	123
7.5	GATHER BENCHMARKS FOR TOKENS	124
7.6	EXPLORE ALTERNATIVE ARCHITECTURES & TECHNOLOGIES	125
7.7	MAKE DESIGN DECISIONS	138
7.8	CREATE IMPLEMENTATION SPECIFICATION	140
7.9	DIFFICULTIES ENCOUNTERED & OVERCOME	140
7.10	CONCLUSION	144
8	MAGIC QUANTIFICATION & SUMMARY	147
8.1	MODEL CONTINUITY IN THE MAGIC SDM	147
8.2	SARKAR QUANTIFICATION OF MAGIC SDM	152
8.3	SUMMARY	161

9 CONCLUSION: DIRECTIONS FOR FURTHER RESEARCH & APPLYING MAGIC TO SOC DOMAIN	163
9.1 APPLIED & BASIC RESEARCH	163
9.2 APPLYING MAGIC CONCEPTS TO THE SOC DOMAIN	165
9.3 VIRTUAL COMPONENT CODESIGN (VCC)	168
9.4 CODESIGN VIS À VIS COVERIFICATION	170
9.5 CONCLUDING COMMENTS	172
APPENDIX A: DETAILS OF VSIPL & MPI MIDDLEWARE	173
A.1 VSIPL: COMPUTATION MIDDLEWARE	173
A.2 MPI: COMMUNICATIONS MIDDLEWARE	180
APPENDIX B: DETAILS OF CASE STUDY	185
B.1 SIMULINK DETAILS	185
B.2 VSIPL CODE GENERATION SUBTLETIES	189
B.3 EARCHITECT DETAILS	189
REFERENCES	203
INDEX	217

List of Figures

Figure 1-1. Cost of specification and design errors increase throughout process (after Figure 1.5 in [1]).	1
Figure 1-2. Basic flow of information needed to support model continuity.	3
Figure 1-3. Convergence of research threads leading to our MAGIC SDM.	9
Figure 2-1. Simplified functional block diagram of “The Processor” in a generic radar system.	12
Figure 2-2. Simplified signal flow and system states representative of the application domain of The Processor of Figure 2-1; the shading shows how such a design might be mapped to an embedded MP target.	13
Figure 2-3. Potential architecture to implement The Processor of Figure 2-1.	15
Figure 2-4. Fundamental perspective of layering software on hardware.	17
Figure 2-5. Expanded model of canonical MP hardware/software codesign layers.	17
Figure 2-6. Canonical model as implemented with recent and current COTS vendor MPs.	18
Figure 2-7. Framework codesign model decoupling computation from configuration.	19
Figure 2-8. Model of Figure 2-6 utilizing standardized middleware.	20
Figure 2-9. Model of Figure 2-7 utilizing standardized middleware.	21
Figure 4-1. Taxonomy of Sarkar’s unified reactive-systems specification methodology attributes (branches) and sub-attributes (leaves).	47
Figure 4-2. Attributes of Sarkar basis.	48
Figure 4-3. Sub-attributes of the language support attribute (in bold).	54
Figure 4-4. Sub-attributes of the complexity control attribute (in bold).	59

Figure 4-5. Sub-attributes of the model continuity attribute (in bold).	63
Figure 4-6. Graphical representation of Sarkar basis attributes.	67
Figure 4-7. Plot of normalized attribute values for the CASE SDMs in Table 4-8.	76
Figure 4-8. Plot of CASE SDMs and Ideal SDM in 3-tuple space.	77
Figure 5-1. Gajski et al.'s three classes and scopes of design methodology.	83
Figure 5-2. Extending Gajski's SER (from Figure 5-1) to our ADOL.	85
Figure 6-1. Simplified diagram of the MAGIC specification and design flow.	89
Figure 6-2. How model continuity is currently lacking in CASE SDMs.	90
Figure 6-3. Establishing model continuity between an executable specification model and a design specification model.	91
Figure 6-4. Simple Simulink model to illustrate code generation.	107
Figure 7-1. SAR block diagram with SAR image processor highlighted in bold italics.	112
Figure 7-2. SAR image formation algorithm flow.	113
Figure 7-3. Simulink model of single threaded version of our SAR processor.	117
Figure 7-4. The model for video-to-baseband conversion.	118
Figure 7-5. Range processing block of our Simulink SAR model.	119
Figure 7-6. Azimuth processing block of our Simulink SAR model.	119
Figure 7-7. Parsing the SAR processor into separate processes.	121
Figure 7-8. Parallelized SAR processing Simulink model where $nPrange=4$ and $nPazimuth=8$.	122
Figure 7-9. Cornerturn required when parallelizing range and azimuth processing	123
Figure 7-10. Token Quantification worksheet (partial screen shot) from executable workbook ($nPrange=4$, $nPazimuth=8$).	129

Figure 7-11. First order estimation of system latencies, based on middleware token delays in the absence of performance modeling.	133
Figure 7-12. Hardware model used for all of our performance models.	134
Figure 7-13. The software model of our performance model.	135
Figure 7-14. The range process with a template used in all of the blocks.	136
Figure 7-15. Latencies for SAR processing architectures based on performance modeling simulations.	138
Figure 7-16. Mapping window for the SAR processing performance model.	143
Figure 7-17. Scatter/send code that shows our flexible mapping.	143
Figure 7-18. Local variables for range processing.	144
Figure 8-1. How model continuity is currently lacking in CASE SDMs (Figure 6-2).	148
Figure 8-2. Establishing model continuity between an executable specification model and a design specification model (Figure 6-3).	149
Figure 8-3. Establishing model continuity between an executable specification model and a design specification model.	151
Figure 8-4. Plot of normalized attribute values for the CASE SDMs vis à vis the Ideal and MAGIC SDMs from Table 8-11.	160
Figure 8-5. Plot of SDMs in 3-D attribute-space, which shows MAGIC's improvement over CASE SMDs moving towards Ideal SDM.	161
Figure 9-1. Methodological flow of Cadence's Virtual Component Codesign (VCC) infrastructure.	169
Figure 9-2. Conceptual illustration of the complexity of design space exploration.	171
Figure A-1. Data space characteristics and interrelationship.	177
Figure A-2. VSIPL application flow.	179
Figure B-1. Data input block.	186
Figure B-2. Formatting the SAR image for output.	187

Figure B-3. Setting the model execution parameters.	188
Figure B-4. Demux dialog box used for scattering the data for range processing.	188
Figure B-5. Mux dialog box used for gathering range results for the cornerturn.	189
Figure B-6. Project window at the beginning of an eArchitect session.	191
Figure B-7. Opening the supplemental Mercury library to support our MAGIC SDM.	191
Figure B-8. Hardware model of Mercury MCH6 motherboard.	193
Figure B-9. Hardware model of Mercury PPC daughtercard.	194
Figure B-10. One of two compute nodes (CE) on daughtercard.	195
Figure B-11. Contents of a software block's process that performs a delay.	195
Figure B-12. Local variables for FIR processing.	196
Figure B-13. Local variables for the cornerturn.	196
Figure B-14. Local variables for azimuth processing.	196
Figure B-15. Local variables for display processing.	197
Figure B-16. Parameters of the input data source.	198
Figure B-17. Dialog window to set parameters for simulation.	198
Figure B-18. Tool for accessing simulation runs for analysis.	199
Figure B-19. Analysis tools for visualization of simulation data.	199
Figure B-20. After loading our simulation run we see the bottom-line latency.	200
Figure B-21. Tools options in the Analysis Control Panel.	200
Figure B-22. Simulation time-line showing the exact latency of the simulation is 1.207 seconds.	201

List of Tables

Table 2-1. Basic characteristics of COTS MP codesign frameworks.	29
Table 3-1. Summary of models of computation (MoCs) with example frameworks and/or languages	38
Table 3-2. Models of computation implicit in frameworks of §2.1.4.	44
Table 4-1. Language support conceptual models sub-attributes spreadsheet.	70
Table 4-2. Language support analysis techniques sub-attributes spreadsheet.	71
Table 4-3. Representational complexity control sub-attributes spreadsheet.	72
Table 4-4. Developmental complexity control sub-attributes spreadsheet.	73
Table 4-5. Model continuity model integration sub-attributes spreadsheet.	74
Table 4-6. Model continuity implementation sub-attributes spreadsheet.	75
Table 4-7. Raw values for attribute integrations.	75
Table 4-8. Normalized attribute values for the CASE SDMs.	76
Table 5-1. Extending SER design representation and abstraction levels to our ADoI (Board Level).	81
Table 5-2. Language support for conceptual model characteristics of embedded systems.	81
Table 5-3. System design tasks.	84
Table 6-1. Technologies in our application domain.	98
Table 6-2. Summary of transformation rules for code generator.	106
Table 7-1. Summary of system requirements and constraints.	114
Table 7-2. Environmental non-performance characteristics of processor boards.	116

Table 7-3. Example of system parameters portion of Token Quantification worksheet.	128
Table 7-4. Example of token quantification and non-IPC latency computation ($nPrange=4$, $nPazimuth=8$).	130
Table 7-5. Token Summary for performance modeling, including latency estimates without accounting for IPC.	132
Table 7-6. Ruling out architectures that do not meet scalability requirement (in black).	132
Table 7-7. Latencies of SAR processor architectures accounting for IPC. (Architectures that do not meet scalability requirement are shaded.)	137
Table 7-8. Assessing our design options, optimizing on minimal board count, N_{boards} .	139
Table 8-1. Normalized language attribute values and the improvement with the MAGIC SDM.	152
Table 8-2. Language support conceptual models sub-attributes spreadsheet including MAGIC SDM.	153
Table 8-3. Language support analysis techniques sub-attributes spreadsheet including MAGIC SDM.	154
Table 8-4. Normalized complexity control attribute values and the improvement with the MAGIC SDM.	155
Table 8-5. Normalized model continuity attribute values and the improvement with the MAGIC SDM.	155
Table 8-6. Representational complexity control sub-attributes spreadsheet including MAGIC SDM.	156
Table 8-7. Developmental complexity control sub-attributes spreadsheet including MAGIC SDM.	157
Table 8-8. Model continuity model integration sub-attributes spreadsheet including MAGIC SDM.	158
Table 8-9. Model continuity implementation sub-attributes spreadsheet including MAGIC SDM.	159

Table 8-10. Raw attributes for CASE SDMs vis à vis Ideal and MAGIC SDMs.	159
Table 8-11. Normalized attributes for CASE SDMs vis à vis Ideal and MAGIC SDMs.	160
Table 8-12. Normalized model continuity attribute values and the improvement with the MAGIC SDM.	161
Table B-1. Subtle VSIPL code generation equivalents.	190

List of Acronyms

ACL	Application Configuration Language
ACT	Autocoding Toolset
ADC	Analog-to-digital converter
ADoI	Application domain of interest
API	Application programming interface
BM-2	Benchmark-2
CAD	Computer-aided design
CASE	Computer-aided system engineering
CE	Compute Element
CFD	Control flow diagram
CFSM	Communicating finite state machine
COTS	Commercial-off-the-Shelf
CPS	Composite program-states
DAC	Digital-to-analog converter
DARPA	Defense Advanced Research Projects Agency
DCDF	Directed-control dataflow
DE	Discrete event
DF	Dataflow
DFD	Dataflow diagram
DFT	Discrete Fourier transform
DSP	Digital signal processing
DSPW	DSP Workstation (MATLAB/Simulink/Stateflow framework)
FFT	Fast Fourier transform
FLOPS	Floating-point operations
FLOPS/s	Floating-point operations per second
FP(s)	Function Point(s)
FSM	Finite state machine
GE	Graphical editor
GEDAE	Graphical Entry, Distributed Application Environment
GFLOPS/s	Gigaflops per second
GPP	General-purpose processor
GUI	Graphical user interface
IP	Intellectual property
IPC	Interprocessor communication
ISPME	Integrated Specification and Performance Modeling Environment

LAN	Local area network
MAGIC	Methodology Applying Generation, Integration, and Continuity
MB/s	Megabytes per second
MCCI	Management Communications and Control, Inc.
MFLOPS	Million floating-point operations
MFLOPS/s	Million floating-point operations per second
MIPS	Million instructions per second
MoC	Model of computation
MP	Multiprocessor
MPI	Message Passing Interface
MPI/RT	Message Passing Interface/Real-Time
MSTI	MPI Software Technology, Inc.
NOW	Network of workstations
OO	Object-oriented
PMS	Processors/Memories/Switches
PS	Program-states
PSM	Programming-state machine
PSS	Program-substates
QoS	Quality-of-service
RASSP	Rapid prototyping of Application Specific Signal Processors
RIPPEN	Real-time Interactive Programming and Processing Environment
RISC	Reduced Instruction Set Computing
RTOS	Real-time operating system
RTW	Real-Time Workshop
RTWG	RASSP Taxonomy Working Group
S/R	Synchronous/reactive
SAN	System-area network
SAR	Synthetic aperture radar
SBC	Single board computer
SPC	Scalable parallel computer
SDE	Software development environment
SDF	Synchronous dataflow
SDL	Specification and Description Language
SDM	Specification and design methodology
SDS	Software development specification
SDE	Software development environment
SER	Specify-Explore-Refine
SiB	System-in-a-box
SLDL	System Level Design Language
SoC	System-on-a-chip

SPMD	Single-program multiple-data
SREM	Software Requirements Engineering Methodology
SRS	System requirements specification
STAP	Space-time adaptive processing
SWAP	Size, weight, and power
TBD	To be defined
TMW	The MathWorks
UI	User interface
VC	Virtual component
VP	Virtual prototype/prototyping
VSIPL	Vector/Signal/Image Processing Library
WAN	Wide area network

Foreword

Specification and design methodology has seen significant growth as a research area over the last decade, tracking but lagging behind VLSI design technology in general and the CAD industry in particular. The commercial rush to market tries to leverage existing technology which fuels CAD design tool development. Paralleling this is very active basic and applied research to investigate and move forward rational and effective methodologies for accomplishing digital design, especially in the field of hardware/software codesign. It is this close relationship between industry and academia that makes close cooperation between researchers and practitioners so important—and monographs like this that combine both abstract concept and pragmatic implementation deftly bridge this often gaping chasm.

It was at the IEEE/ACM Eighth International Symposium on Hardware/Software Codesign where I met the author of this monograph, Dr. Randall Janka, who was presenting some of his recent dissertation research results on specification and design methodology, or as he has so succinctly defined this sometimes ambiguous concept, “the tools and rules.” Where so many codesign researchers are trying to prove out different aspects of codesign and using toy applications to do so, Dr. Janka had developed a complete specification and design methodology and prototyped the infrastructure—and proven its viability, utility, and effectiveness using a demanding real-world application of a real-time synthetic aperture radar imaging processor that was implemented with embedded parallel processors.

It is gratifying to see our own work in general and our Specify-Explore-Refine (SER) design methodology in particular serve as a stimulus to the specification and design methodology contained in this monograph. Dr. Janka, who authored this volume originally as his doctoral dissertation at Georgia Tech, was able to leverage critical core concepts of our SER methodology and then conceive a novel and practical specification and design methodology. An important key contribution of his work includes the practical instantiation of model continuity, an abstract concept that refers to the ability to pass a model of the system under design from specification to design to implementation—without losing any details. Another important contribution is that his “MAGIC” specification and design methodology does account for non-performance constraints such as size, weight, and power as

well as important real-time performance constraints—long before the designer commits to specific hardware and software structural components.

Despite being based on academic work, this book is remarkably practical. It provides an excellent snapshot of the state-of-the-art and gives the reader a good understanding of both the fundamental challenges of specification and design as well as a unified and quantified ability to assess a given methodology. How he has conceived and prototyped the infrastructure to validate his MAGIC methodology will be of value to both researcher and practicing engineer, especially the workarounds used to integrate tools from different vendors and domains. This book serves as a very useful reference in a very dynamic and evolving field.

Professor Daniel Gajski, Director
Center for Embedded Computer Systems
University of California Irvine

Preface

This book is an update and expansion of my doctoral dissertation, which I successfully defended in the School of Electrical and Computer Engineering in the Georgia Institute of Technology (Georgia Tech) in December, 1999, under the title, *A Model-Continuous Specification and Design Methodology for Embedded Multiprocessor Signal Processing Systems*. However, before a practicing engineer quickly puts this book back on the shelf, I should note that this dissertation has deep roots in “the real world” and has many pragmatisms contained within.

This book reflects not only my research focus of the five years preceding the completion of my dissertation, but as a practicing engineer since the late 1970s, it also reflects experience and heuristics resulting from design, development, and research interests and activities of roughly twenty years. I began this dissertation while working as an advanced software tools engineer at Mercury Computer Systems in Chelmsford, Massachusetts, and completed it while working as a Senior Research Engineer in the Radar Signal Processing branch of the Georgia Tech Research Institute, the applied R&D organization within Georgia Tech. While appropriately analytical and “scholarly,” my dissertation contains many pragmatisms that originated in that amorphous region of “the real world” where I have worked for years.

Design and its various facets such as specification, analysis, implementation, optimization, etc., are inherently complex, and therefore it is difficult to wrap one’s brain around even one of these facets. I recently coined a definition of “the system engineer” when asked to speak at this year’s IEEE International Workshop on Rapid System Prototyping (RSP), which explores trends in rapid product development of computer-based systems:

The systems engineer is that person who has been around a while and has done hard time designing hardware, software, and firmware—and has even met a few schedules! But truth be told, system engineers learn more from the schedules missed than the schedules met.

I have done my share of missing and meeting schedules in all three of these implementation domains and in different application domains, which include industrial controllers, missile-borne radar seekers, satellite modem testbeds,

and various signal and image processing systems that required embedded parallel processing power. All of these systems were real-time embedded systems with their accompanying performance and non-performance constraints.

Consequently, this book contains a great deal of material that will be of interest to both the researcher and the practitioner. I have been gratified by the number of hits my website has received at the Georgia Tech Research Instituteⁱ (GTRI), which GTRI so graciously left up after I left GTRI to work for Cadence Design Systems, Inc. It is this interest in my dissertation and its dualism that made me decide to leave it essentially unaltered when Kluwer convinced me to publish it as a monograph. The front material (Dedication and Acknowledgements) remains unchanged except for including those people at Cadence and Kluwer as well as Professor Daniel Gajski who deserve my gratitude. The first seven chapters remain unaltered, except for cosmetic changes to meet Kluwer's style requirements or to update references. The last chapter (Chapter 8) of my dissertation has been changed to enlarge upon the section regarding directions for further research (especially system-on-a-chip codesign) and to add additional pertinent references.

The new chapter in this book (Chapter 9) begins to answer questions raised at the end of my dissertation (originally §8.4) since two years has passed since completing it. Both before and since completing the dissertation, I have had the privilege of publishing and presenting over a dozen papers at various refereed research and technology conferences regarding my "MAGIC" specification and design methodology and various aspects surrounding it, gaining valuable feedback from the community. I've also made a major transition from working in the "system-in-a-box" ("SiB") world (embedded parallel processing) to the "system-on-a-chip" ("SoC") world. Ironically, while I was working on my MAGIC methodology and its prototype infrastructure for use in the SiB community, Cadence Design Systems was working on a conceptually similar infrastructure targeting the SoC community, known as "VCC." The similarities between these two infrastructures is striking since they were developed independently and on opposite coasts here in the United States.

While technology and the technological marketplace evolve rapidlyⁱⁱ, the fundamental specification and design methodological issues such as

ⁱ<http://seal-www.gtri.gatech.edu/rjanka>.

ⁱⁱI should note that there have been some changes both in the industry and tools used in my infrastructure prototype. The performance modeling framework I used, eArchitect, was produced by Viewlogic, which has now become Innoveda. The MathWorks has also released a significant upgrade of Simulink (R12.x) that has eliminated the radar data square data referencing workarounds. The MathWorks also has dropped the use of the moniker "DSP

specification, design analysis, models of computation, model continuity, and application domain specificity have not changed. Technology evolves dramatically, but design fundamentals do not. With this said I now commend the reader to the original dissertation summary and the following book for his/her edification:

The process of designing large real-time embedded signal processing systems is plagued by a lack of coherent specification and design methodology. A canonical waterfall design process is commonly used to specify, design, and implement these systems with commercial-off-the-shelf (COTS) multiprocessing (MP) hardware and software. Powerful frameworks exist for each individual phase of this canonical design process, but no single methodology exists which enables these frameworks to work together coherently, i.e., allowing the output of a framework used in one phase to be consumed by a different framework used in the next phase.

This lack of coherence usually leads to design errors that are not caught until well in to the implementation phase. Since the cost of redesign increases as the design moves through these three stages, redesign is the most expensive if not performed until the implementation phase, thus making the current incoherent methodology costly. This dissertation shows how designs targeting COTS MP technologies can be improved by providing a coherent coupling between these frameworks, a quality known as “model continuity.”

We have developed a new specification and design methodology (SDM) which accomplishes the requirements specification, design exploration, and implementation of COTS MP-based signal processing systems by using powerful commercial frameworks that are intelligently integrated into a single domain-specific SDM. Our integration establishes model continuity by using autogenerated computation (VSIPL) and communication (MPI) standards-based middleware. We have dubbed our new SDM MAGIC, an acronym for “Methodology Applying Generation, Integration, and Continuity.”

To measure improvement, we have developed an analytical means of measuring SDMs in our domain by quantifying Sarkar’s unified basis for evaluating specification-modeling methodologies. We measured computer-aided system engineering (CASE) SDMs capable of generating real-time code and our own MAGIC SDM, and found the MAGIC SDM was much closer to ideal than the CASE SDMs. We have also validated the MAGIC SDM and demonstrated its efficacy with a real-world benchmark. In so doing

Workstation” to refer to the MATLAB/Simulink/Stateflow/RTW suite. The COTS multiprocessing hardware industry is also going through expected evolution with the players changing, though Mercury Computer Systems still leads the way. It is a similar story in the COTS multiprocessing CASE frameworks market segment.

we also demonstrated that the MAGIC SDM is clearly superior to both VHDL virtual prototyping and the CASE-based SDMs that must commit to an implementation technology before performing design analysis. We also consider further research directions.

Randall S. Janka
Atlanta, Georgia, USA

Acknowledgements

This is the section that is written last and that I have been looking forward to the most. It should be written last, because it is only at the end of such a long road can one appreciate all who have helped him travel down that road all the way to its end. My road is long, having begun sixteen years ago, so I owe a debt of gratitude to many.

It has been an absolute blessing to be a part of Georgia Tech, especially being able to work full-time for the Georgia Tech Research Institute (GTRI). There are two people at Georgia Tech who have been key in my completing this dissertation and for whom I am extremely grateful. The first is my advisor, Dr. Linda Wills, who has been phenomenal in helping me pull order out of chaos. Besides providing excellent mentoring in how to develop my thesis, she has been incredibly supportive and encouraging. Finding editorial comments with time stamps of one or three in the morning says a lot about her dedication. Her enthusiasm and professionalism are exemplary. The other is my boss, Dr. Mark Richards, who hired me into GTRI and made sure that I got into their special program to complete my dissertation by giving me a year off half-time while receiving full-time compensation. He has been an advocate, mentor, supporter, and has also been patient with me while I finished this dissertation. His technical acumen, integrity, and concern for his people are remarkable.

There are others at Georgia Tech who have been most helpful. I am thankful for both Dr. Dale Ray and Dr. David Hertling for their oversight. Dr. Vijay Madisetti has provided some invaluable literature references and search directions, for which I am thankful. I am very grateful for those who have served on my committee, which includes Dr. Henry Owen, Dr. Mark Richards, Dr. Linda Wills, and Dr. Sudhakar Yalamanchili. They have been a great source of insight and support. Thanks also to Marilou Mycko and Glenna Thomas who provide amazing administrative support, often under great pressure.

There are others at GTRI who have been quite helpful while completing my dissertation. They include my lab director, Dr. Bob Trebits of SEAL and my division director Guy Morris of RSD. Janice Rogers of HQ has been creative and flexible in providing coverage for dissertation completion as well as conferences. Others at GTRI who have been extremely helpful over the last couple of years include support staff like Melanie Price, Karen Everson and her computer support staff. Fellow GTRI research colleagues

who have been both helpful and just plain encouraging include Dr. Byron Keel, Dr. Chris Barnes, Dr. Bill Holm, and especially Bill Marshall, my fellow traveler on the long road with whom I've had the pleasure of commiserating evenings, late at night, and weekends. Congratulations, Dr. Marshall—and thanks for all your help beating the network into submission so I could do all my performance modeling simulations.

A critical element of my research included the integration of commercial products into my methodology. I have had the pleasure of dealing directly with the developers of bleeding edge state-of-the-art COTS multiprocessing hardware and software that have been so integral to my research. I have been most fortunate in being able to have received gratis the hardware and software that was the best of breed as well as what I specified. I am extremely grateful to Mercury Computer Systems not just for the privilege of being able to work there for two years, but especially for their generous contributions to GTRI of hardware and software to support my research and our project work. Thanks to Barry Isenstein, Arlan Pool, Dave Toms, Karen Lauro, and the customer support team. It has been a privilege and pleasure to have worked directly and indirectly with the technical staff at The MathWorks over the years, especially recently on this dissertation. I am grateful for their software donations and for the interchanges with Dr. Don Orofino and Rick Drohan. I am especially thankful for Stuart McGarrity for his help with the SAR Simulink modeling. Viewlogic has been generous in both their software donations and assistance, and I thank Binoy Yohannan and especially Mark Hepburn for his help with the performance modeling using eArchitect. Thanks, also, to Christopher Robbins and Carl Ecklund of MCCI for PGM ACT, and to W. Bernard Schaming and William Lundgren of Lockheed Martin ATL for GEDAE.

It has been a real pleasure working in the middleware arena, co-chairing the VSIPL Forum and peripherally supporting the MPI/RT Forum. Those whose insights and contributions to VSIPL have helped make it a genuine computation middleware standard include Dr. David Schwartz, Randall Judd, Dr. James Lebak, Dr. Sharon Sacco, Steve Paavola, and James Kenny. I am especially grateful for Dan Campbell of GTRI who has so ably stepped in and helped me with the Test Suite so I could co-chair the VSIPL Forum and for the times I had to occasionally retreat because of my dissertation.

I am told it is extremely unusual to be able to work right in the core of one's dissertation research, yet the unusual is what God does best, and that is what He has enabled me to do. I received a great opportunity to return to the commercial world in general, and to Cadence Design Systems, Inc., in particular. Working with the caliber of talent at such a company has been like doing post-doc work. I am extremely grateful for the privilege and pleasure of working with those who have developed a commercial SoC-targeted

MAGIC-like infrastructure, Virtual Component Codesign (VCC), which is already defining next-generation system-level specification and design. I am grateful to be able to work with people like Frank Schirrmeister, Dr. Stan Krolikoski, Grant Martin, Mark Peters, Dr. Robert Juliano, Greg Maxey, and so many others in the Codesign and Coverification Technology group of the Systems and Functional Verification division.

I must also express gratitude to Professor Daniel Gajski of the University of California Irvine for not only inspiring my own research through his many books and papers, but also for his personal encouragement and especially for consenting to write the foreword to this book. I am also thankful to Carl Harris and his staff at Kluwer Academic Publishers for their help in converting an academic dissertation into a monograph of interest to a much larger audience.

Jesus said, “The first shall be last,” so I close these acknowledgements with expressing the deepest of gratitude to my wife, Beverly, who has been a radiant woman of God through the tough times of completing my doctorate. Her belief in me, encouragement, joyful disposition (especially her singing), and never complaining even during crunch times is a testimony to her heart and faith. My three daughters have also paid a price for their daddy to finish this dissertation. While Alexandra, Bethany, and Victoria made it clear they missed their daddy, they have persevered and been patient. Their notes, prayers, hugs, and kisses kept me going on more than one occasion.

Most of all I want to thank God who has been so incredibly faithful to His promises and to me (Psalm 37:3-6). I knew that He would have to intervene in His own sovereign way for me to complete this dissertation. I decided after my twins were born, since I was a disciple of Jesus, a husband, and a father, that finishing my graduate work would require some unconventional creativity on God’s part. It seemed that finishing my doctorate and most recently my dissertation would have to be part of a full-time job since there would be no other way that I could be the disciple, husband, or father Jesus calls men to be. So God moved me to Atlanta to work for GTRI. Yet even with this blessing, it has been a challenge. I thank God for the inspiration, example, faithfulness, conviction, prayers and love of the awesome Atlanta Church of Christ and especially Sid Howell and Wayne Sisco for being “brothers who have stuck closer than a friend” (Proverbs 18:24). Thank you Father, for your faithfulness and for holding my hand.

*“If the LORD delights in a man's way, He makes his steps firm;
though he stumble, he will not fall, for the LORD upholds him with His
hand.”*

—Psalm 37:23-24