Contracts for System Design

Albert Benveniste

Benoît Caillaud

Dejan Nickovic

Roberto Passerone

Jean-Baptiste Raclet

Philipp Reinkemeier

Alberto Sangiovanni-Vincentelli

Werner Damm

Thomas A. Henzinger

Kim G. Larsen

now

the essence of knowledge

Boston — Delft

Foundations and Trends® in Electronic Design Automation

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

A. Benveniste, B. Caillaud, D. Nickovic, R. Passerone, J-B. Raclet, Ph. Reinkemeier, A. Sangiovanni-Vincentelli, W. Damm, T. Henzinger, and K.G. Larsen. *Contracts for System Design*. Foundations and Trends® in Electronic Design Automation, vol. 12, no. 2-3, pp. 124–400, 2018.

This Foundations and Trends® issue was typeset in LaTeX using a class file designed by Neal Parikh. Printed on acid-free paper.

ISBN: 978-1-68083-402-4

Foundations and Trends® in Electronic Design Automation
Volume 12, Issue 2-3, 2018

Editorial Board

Editorial Scope

Topics

Foundations and Trends® in Electronic Design Automation publishes survey and tutorial articles in the following topics:

- System level design
- Behavioral synthesis
- Logic design
- Verification
- Test
- Physical design
- Circuit level design
- Reconfigurable systems
- Analog design
- Embedded software and parallel programming
- Multicore, GPU, FPGA, and heterogeneous systems
- Distributed, networked embedded systems
- Real-time and cyberphysical systems

Information for Librarians

Foundations and Trends® in Electronic Design Automation, 2018, Volume 12, 4 issues. ISSN paper version 1551-3939. ISSN online version 1551-3947. Also available as a combined paper and online subscription.

Foundations and Trends® in Electronic Design Automation
Vol. 12, No. 2-3 (2018) 124–400

DOI: 10.1561/1000000053

now
the essence of knowledge

Contracts for System Design

Albert Benveniste
INRIA, Rennes, France
albert.benveniste@inria.fr

Benoît Caillaud
INRIA, Rennes, France
benoit.caillaud@inria.fr

Dejan Nickovic
Austrian Institute of Technology
Dejan.Nickovic@ait.ac.at

Roberto Passerone
University of Trento, Italy
roberto.passerone@unitn.it

Jean-Baptiste Raclet
IRIT, Toulouse, France
raclet@irit.fr

Philipp Reinkemeier
Offis, Oldenburg, Germany
Philipp.Reinkemeier@offis.de

Alberto Sangiovanni-Vincentelli
University of California at Berkeley
alberto@eecs.berkeley.edu

Werner Damm
Offis and University of Oldenburg, Germany
werner.damm@offis.de

Thomas A. Henzinger
IST Austria, Klosterneuburg
tah@ist.ac.at

Kim G. Larsen
Aalborg University, Denmark
kgl@cs.aau.dk

Full text available at: http://dx.doi.org/10.1561/1000000053

Contents

Abstract

Recently, *contract-based design* has been proposed as an "orthogonal" approach that complements system design methodologies proposed so far to cope with the complexity of system design. Contract-based design provides a rigorous scaffolding for verification, analysis, abstraction/refinement, and even synthesis. A number of results have been obtained in this domain but a unified treatment of the topic that can help put contract-based design in perspective was missing. This monograph intends to provide such a treatment where contracts are precisely defined and characterized so that they can be used in design methodologies with no ambiguity. In particular, this monograph identifies the essence of complex system design using contracts through a mathematical "meta-theory", where all the properties of the methodology are derived from a very abstract and generic notion of contract. We show that the meta-theory provides deep and illuminating links with existing contract and interface theories, as well as guidelines for designing new theories. Our study encompasses contracts for both software and systems, with emphasis on the latter. We illustrate the use of contracts with two examples: requirement engineering for a parking garage management, and the development of contracts for timing and scheduling in the context of the AUTOSAR methodology in use in the automotive sector.

A. Benveniste, B. Caillaud, D. Nickovic, R. Passerone, J-B. Raclet, Ph. Reinkemeier, A. Sangiovanni-Vincentelli, W. Damm, T. Henzinger, and K.G. Larsen. *Contracts for System Design*. Foundations and Trends® in Electronic Design Automation, vol. 12, no. 2-3, pp. 124–400, 2018.
DOI: 10.1561/1000000053.

1

Introduction

1.1 Industrial context

System companies such as automotive, avionics and consumer electronics enterprises are facing significant difficulties due to the exponentially raising complexity of their products coupled with increasingly tight demands on functionality, correctness, and time-to-market. The cost of being late to market or of imperfections in the products is staggering as witnessed by the recent recalls and delivery delays that system industries had to bear. Many challenges face the system community to deliver products that are reliable and effective. Table 1.1, albeit not recent, continues to be a telling example of the main causes and their share in the difficulties related to systems complexity.[1] This table highlights the importance of system integration, where corrections occur late in the design flow and are therefore very costly.

System specification and integration is particularly critical for Original Equipment Manufacturers (OEM) managing the integration and maintenance process with subsystems that come from different suppliers who use different design methods, different software architectures, and different hardware

[1] Source: VDC research, Track 3: Embedded Systems Market Statistics Exhibit II-13 from volumes on automotive/industrial automation/medical, 2008

Table 1.1: Difficulties related to system complexity. The table displays, for each industrial sector, the percentage of tasks delayed and tasks causing delays, for the different phases of system design.

Design task	Tasks delayed automotive	Tasks delayed automation	Tasks delayed medical
System integration test, and verification	63.0%	56.5%	66.7%
System architecture design and specification	29.6%	26.1%	33.3%
Software application and/or middleware development and test	44.4%	30.4%	75.0%
Project management and planning	37.0%	28.3%	16.7%
Design task	Tasks causing delay automotive	Tasks causing delay automation	Tasks causing delay medical
System integration test, and verification	42.3%	19.0%	37.5%
System architecture design and specification	38.5%	42.9%	31.3%
Software application and/or middleware development and test	26.9%	31.0%	25.0%
Project management and planning	53.8%	38.1%	37.5%

Source: VDC research, Track 3: Embedded Systems Market Statistics Exhibit II-13 from volumes on automotive/industrial automation/medical, 2008. http://www.vdcresearch.com/.

platforms. In addition, even inside an OEM itself, complex systems involve a number of different aspects or viewpoints that are generally handled by

different teams using different paradigms and tools. Examples of aspects are system architecture, the functions the system should perform and the services it should deliver, its safety and reliability characteristics, its energy budget, its deployment on an embedded computing platform to name a few.

Contract-based design has as main goal solving the above issues in a rigorous framework.

1.2 Positive impact of contract-based design

Addressing the Complexity of Systems. Several approaches have been developed by research institutions and industry to cope with the exponential growth in systems complexity. Of particular interest to the development of embedded controllers and systems are *layered design* and *component-based design* (used, e.g., in the AUTOSAR[2] standard in the automotive sector, and the ARINC[3] standard in the avionic domain), *model-based development* (supported by important frameworks and tools such as SysML[4] [208] and/or AADL [211] for architecture modeling, and Modelica [133] and Matlab-Simulink [168] for system modeling), *virtual integration* (Ptolemy [124] and Metropolis [100, 68]), and *platform-based design* [100, 114, 221]. There are two basic principles followed by these methods: abstraction/refinement and composition/decomposition. Abstraction and refinement are processes that relate to the flow of design between different layers of abstraction (vertical process) while composition and decomposition operate at the same level of abstraction (horizontal process). Layered design and model-based development focus on the vertical process while component-based design deals principally with the horizontal process. Platform-based design combines the two aspects in a unified framework.

While the above methods have been critical steps in breaking systems complexity, they do not by themselves provide the ultimate answer. Contracts are ideal tools to solidify both vertical and horizontal processes providing the theoretical background to support formal methods in system design. When design is being performed at a considered layer, implicit — and often hidden — assumptions regarding other layers (e.g., computing resources)

[2] http://www.autosar.org/
[3] https://www.aviation-ia.com/product-categories/arinc
[4] http://www.omg.org/spec/SysML/

are typically invoked by the designer. Actual properties of these other layers, however, cannot be compared against these hidden assumptions. Similarly, when components or sub-systems are abstracted via their interfaces in component based design, it is generally not true that such interfaces provide sufficient information for other components to be safely implemented based on this sole interface. By pinpointing responsibilities and making hidden assumptions explicit, contract-based design provides the due discipline, concepts, and techniques to cope with this.

Another challenge for component-based design of embedded systems is to provide interface specifications that address behaviors, not only type properties of interfaces, and are rich enough to cover all phases of the design cycle. This calls for including non-functional characteristics as part of the component interface specifications, which is best achieved by using multiple viewpoints [40, 46, 42]. Contract-based design supports multiple viewpoints by giving a mathematically precise answer to what it means to fuse them.

Addressing OEM-Supplier Chains and Managing Requirements. The management of responsibilities in and design processes across OEM-supplier chains is indeed the core target of contract-based design. By making the explication of implicit assumptions mandatory, contracts help assign responsibilities to a precise stake holder for each design entity. By supporting independent development of the different sub-systems while guaranteeing smooth system integration, they orthogonalize the development of complex systems. Contracts are thus adequate candidates for a technical counterpart of the legal bindings between partners involved in the distributed and concurrent development of a system.

Regarding requirement capture, efforts have been made by paying close attention to book-keeping activities such as the management of the requirement descriptions and corresponding traceability support (e.g., using commercial tools such as Doors[5] in combination with Reqtify[6]) and by inserting, whenever possible, precise formulation and analysis methods and tools. Still, the need for basing requirement engineering on more solid bases is widely

[5] https://www.ibm.com/us-en/marketplace/rational-doors
[6] https://www.3ds.com/fr/produits-et-services/catia/produits/reqtify/

acknowledged. Specifications used for procurement should be precise, unambiguous, and complete. Indeed, a recurrent reason for failures causing deep iterations across supply chain boundaries rests in incomplete characterizations of the conditions for use and environment of the system to be developed by the supplier, such as missing information about failure modes and failure rates, missing information on possible sources of interference through shared resources, and missing boundary conditions. This argument highlights the need of making assumptions on the design context explicit in OEM-supplier commercial contracts. The potentially highest value proposition of a systematic introduction of contracts indeed lies in requirement capture. Already the evaluation results of the industrial partners in the Integrated Project Speeds[7] acclaim the use of contracts for the requirement capture phase to substantially increase the quality of requirements.

By systematically enforcing the explication of assumptions, systems understanding and thus system interface specifications are substantially improved. Thinking in terms of assumptions uncovers early potential incompatibilities, which otherwise would have only been found much later in integration stages. Furthermore, (*i*) the explication of assumptions significantly eases concurrent engineering; assumptions provide a natural way of communication between design teams; (*ii*) the quality improvements in requirements translates directly to improvement of test cases for requirement-based testing; and (*iii*) the effort spent in explicating assumptions translates directly to improvement of test cases for integration testing. Assumptions are easily integrated into industrial design flows for requirement capture, including tools for traceability and change management. Further, formalized contracts allow for a rigorous checking of otherwise easily overlooked inconsistencies between requirements. Formalized contracts allow for "playing out" contracts — a term coined by David Harel [149, 147] — i.e., executing formalized specifications by engines that systematically generate all behaviours possible under the current set of contracts. Such simulation based environments give strong support for checking the completeness of requirements. Finally, vectors for requirement based testing and virtual integration testing can be automatically derived from formalized contracts, again leading to a significant quality improvement. Observers can be automatically generated from

[7] http://cordis.europa.eu/project/rcn/79466_en.htm

formalized contracts and used in model-, software- and hardware-in-the-loop testing, or even integrated into execution platforms e.g. to diagnose failure situations. We will thus in this monograph elaborate in particular on the benefits of formalized contracts for requirement capture.

1.3 A bird's eye view of research in contracts

The notion of contract is not new. It was first developed and promoted in the community of software engineering, and more specifically Model Driven Engineering. Actually, Design by Contract is a software engineering technique popularized by Bertrand Meyer [200, 201] following earlier ideas from Floyd-Hoare logic [234, 155]. Floyd-Hoare logic assigns meaning to sequential imperative programs in the form of triples of assertions consisting of a precondition on program states and inputs, a command, and a postcondition on program states and outputs. So far contracts consisting of pre/postconditions naturally fit imperative sequential programming. In situations where programs may operate concurrently, interference on shared variables can occur. *Rely/Guarantee* rules [159] were thus added to interface contracts. Rely conditions state assumptions about any interference on shared variables during the execution of operations by the system's environment. Guarantee conditions state obligations of the operation regarding shared variables.

Despite early contributions by Abadi, Lamport, and Wolper [5, 3], developing contracts for Cyber-Physical Systems [236, 100][8] and Reactive Systems [146, 152, 142, 196], where mathematical behaviors are essential, boomed more recently in the 2000's, when de Alfaro and Henzinger proposed and popularized so-called *interface theories* [105, 103, 8]. Since then, a number of models have been proposed that can be seen as instances of contract theories, either to address a specific technical aspect (e.g., function, timing, and resources), or by following different styles and approaches (Assume/Guarantee contracts or Interfaces).[9]

[8] Distributed physical systems complemented by computing systems.

[9] See the dedicated bibliographical notes in this monograph.

1.4 Contribution of this monograph

This wide diversity in the proposed approaches calls for a clarification of what the essence of a contract theory is. More specifically, we need an abstract and generic theory (a *meta-theory*) of contracts or interfaces that abstracts away how contracts and actual designs are actually represented and still formally defines the following concepts:

- *implementations* and *environments* that conform to the contract; a contract is *consistent* if it possesses legal implementations and *compatible* if it possesses legal environments;
- contract *refinement*, the proper notion of substitutability for contracts;
- *conjunction* of contracts, how to "fuse" different viewpoints;
- *parallel composition* of contracts, how composing (sub-)contracts attached to subsystems yields a system-level contract; the aim is that this parallel composition supports independent development, meaning in particular that composing legal implementations for each subcontract yields a legal implementation for the system-level contract;
- an additional, less essential but still useful concept, is that of *quotient*, which is the adjoint of the notion of parallel composition; how to "patch" an existing design to make it satisfy a new contract.

As the central contribution of this monograph, we thus propose a mathematical *meta-theory of contracts* and specialize it to different existing contract theories and variations thereof. In addition to presenting a number of new results, the monograph has a tutorial value in explaining the role of contracts and interfaces in design. In this respect, we include extensive bibliographical notes with particular attention to the numerous results published since year 2000. Since a number of topics are addressed, we preferred to defer bibliographical studies to the different chapters for each different topic.

The monograph is organized as shown in the Figure 1.1, which shows a dependency map between the different chapters. Chapters 1, 2, and the concluding Chapter 12 address readers who may not be specialists in contracts nor on formal methods (except for the summary of results when describing

the organization of the paper). Chapter 3 is a wide scope discussion of the state of the art—details of recent results are not discussed. Chapter 4, which is a key contribution of this monograph, is more technical but is meant to be self-contained and should be readable by anyone having general skills in mathematics. Chapters 5 to 9 target readers enough exercised in formal methods and, for some parts, even researchers in the field. Some readers may be particularly interested in a particular contract framework and then concentrate on the corresponding chapter. Alternatively, she may be interested in links between frameworks. The two application Chapters 10 and 11 target a wider audience, although they rely on the technical material of previous chapters. A more detailed description of these chapters follows.

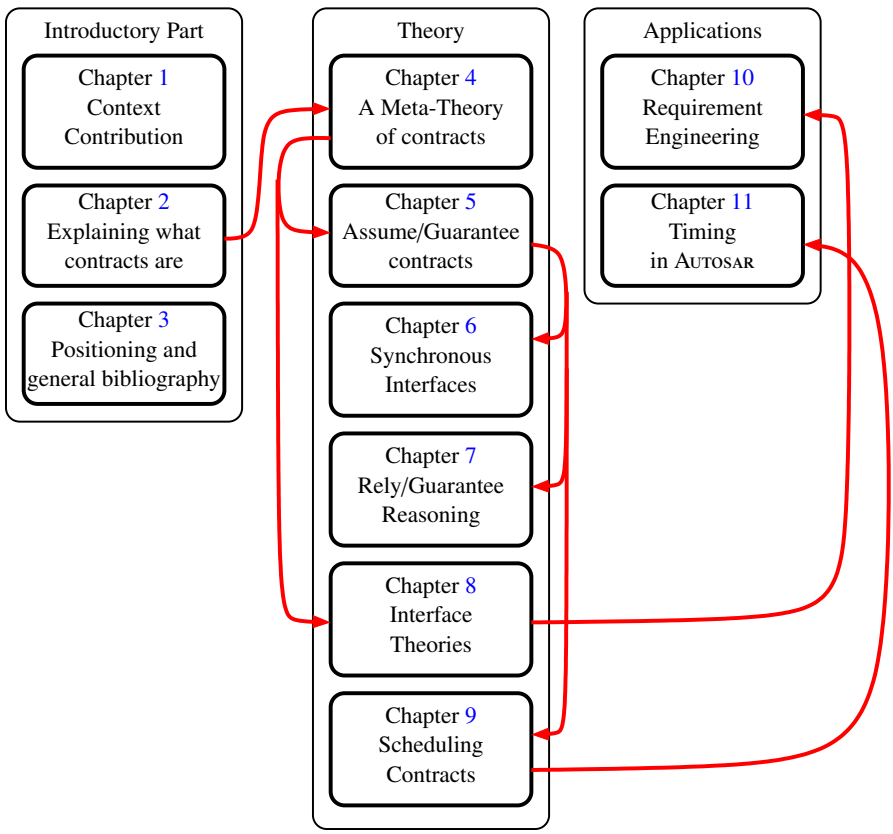

Figure 1.1: Organization of the monograph and dependencies between chapters.

In Chapter 2 we first discuss the requirements on a theory of contracts, based on methodological considerations. In particular we stress the need to support different viewpoints on the system (e.g., operation, function, timing, energy, safety). Then we develop a primer on contracts by using a very simple example requiring only elementary mathematical background. The purpose of this simplistic example is to smoothly and informally introduce the different concepts and operations we need for a contract framework.

Chapter 3 presents a birds eye bibliography of the subject and explains the positioning of our work. So far the links and parallels between the two notions of contract in Object Oriented programming and contract or interface for system design were obscure. In this chapter we draw these two landscapes and pave the way for clarifying the (actually existing) links between these two notions of contract.

Chapter 4 is the cornerstone of this monograph: It presents a new vista on contracts. The so-called "meta-theory" of contracts is introduced and developed in detail. By meta-theory we mean the collection of concepts, operations, and properties that any formal contract framework should offer. Every concrete framework compliant with this meta-theory will inherit these generic properties. The principle of the meta-theory is the definition of a contract as two sets: correct implementations and legal environments. In doing so, we do not assume any particular way of specifying implementations or environments thus making the meta-theory applicable to any contract theory proposed in the literature. Architecture design is greatly facilitated if the framework used allows to re-structure in a different way a system architecture, while preserving its overall semantics (i.e., meaning). A mathematical formalization of this feature is by requiring that the composition operator supporting architecture modeling shall be associative: $(M \times M') \times M'' = M \times (M' \times M'')$, illustrated in Figure 1.2. When applied to contracts, the same property is key in supporting independent development of subsystems by different suppliers with safe system integration. The meta-theory naturally leads, instead, to the consideration of a weaker notion of *sub-associativity,* involving the refinement for its definition — we prove that sub-associativity is sufficient for supporting independent development. Not all concrete contract frameworks possess an associative parallel composition; our results prove that sub-associativity nevertheless holds. We give a tight additional axiom for the

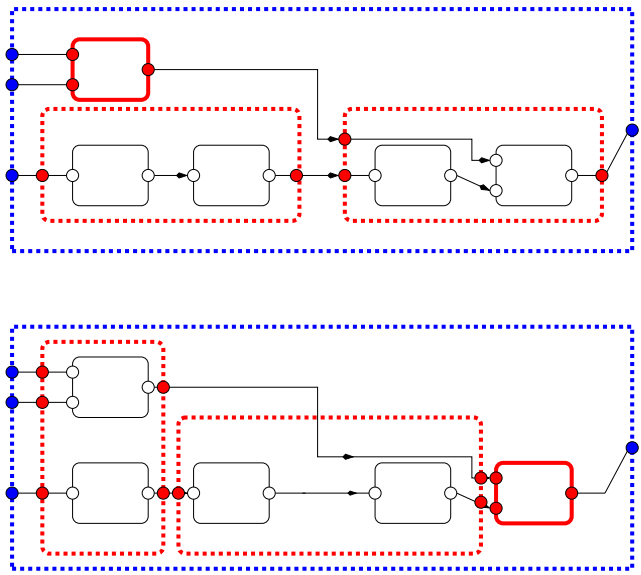

Figure 1.2: Illustrating associativity in architectures. The figures show two architectures using the same set of five components. Components are grouped into different subsystems in the top and bottom architectures. Associativity means that the two obtained architectures should possess identical semantics.

meta-theory ensuring that contract composition is associative and we show that this axiom holds for Assume/Guarantee contracts. We introduce the notion of *quotient,* which supports the practice of patching an existing system to make it satisfy different specifications; the quotient formalizes the concept of "minimal patch". We finally show how abstraction techniques can be defined at the level of the meta-theory, thus specializing to any compliant contract theory. The meta-theory does not specify how components and contracts are effectively represented and manipulated. The subsequent series of chapters presents a panorama of major concrete contract frameworks.

Chapter 5 deals with Assume/Guarantee contracts [40, 46]. This framework is the most straightforward instance of the meta-theory. It presents pairs (A, G) of assumptions and guarantees explicitly, A and G being both expressed as properties. This framework is flexible in that it allows for different styles of description of such properties — computational efficiency depends on the style adopted. In Chapter 6 we relate the Synchronous Interfaces [82]

to the Assume/Guarantee contracts. In Chapter 7 we analyse Rely/Guarantee reasoning [159] used in the area of software engineering and formal methods, to reason about concurrency. We show that this reasoning is also tightly related to Assume/Guarantee contracts. Chapter 8 develops the Interface theories [105, 19], in which assumptions and guarantees are specified by means of a single object: the interface. We revisit the notion of *quotient* for Modal Interfaces [226, 227], to make it the proper specialization of the notion of quotient following the meta-theory. We use this revisited quotient in a holistic methodology for automatically moving from system-level requirements to a set of subcontracts for the different suppliers. We ground on firm bases how Assume/Guarantee contracts can be emulated using Modal Interfaces. Chapter 9 develops a contract framework addressing schedulability analysis, a task involving resource aspects. This framework is subtle because the time and the computing resources both have a strong global flavor.

We complement the above chapters devoted to aspects of the theory with two illustration cases. In Chapter 10, we develop and study requirements for a simple parking garage. Its top-level specification comprises several viewpoints, each one consisting of a requirement table. We pay attention to responsibilities by properly identifying assumptions regarding the environment (context of use), and guarantees offered by the system if properly used. We then study the critical design step consisting in producing sub-contracts for each supplier, following an architecture of sub-systems that differs from the top-level architecture — a frequently encountered situation. We go beyond the state-of-the-art by proposing a *synthesis* method and algorithm, by which the sub-contracts are automatically derived, from the top-level contract and the (SysML-like) topological description of the sub-systems architecture. We discuss the use of contracts in formally establishing properties of the requirements such as consistency, compatibility, and completeness. Despite this being a simple example, it is yet much too complex to be dealt with by hand. A Proof of Concept tool was used to support our development. The contract framework used for this study is the Modal Interfaces.

Chapter 11, which is intended to present an industrially-relevant application, addresses a key part of the AUTOSAR development process in use in the automotive industry. AUTOSAR advocates a design methodology by which the functions, structured into tasks, are first designed independently of the

computing and communication infrastructure, assuming a virtual AUTOSAR run time environment. We study the key step by which time budgets are then allocated to tasks and computing resources are assigned. Lack of formal support in AUTOSAR methodology makes this step difficult today. We show the benefit of using contracts for this step. To this end, we develop an adaptation, called *scheduling contracts*, of the Assume/Guarantee contracts.

Finally the concluding chapter summarizes the lessons drawn from this work and analyzes the industrial situation.

References

[1] D.5.1.2 Pilot Project Evaluation Report. Technical report, SPEEDS project consortium, April 2010.

[2] *10 years AUTOSAR: the worldwide automotive standard for E/E systems*. ATZ extra. Springer Vieweg, 2013. available from https://books.google.fr/books?id=DQJKnwEACAAJ.

[3] Martín Abadi and Leslie Lamport. Composing specifications. *ACM Trans. Program. Lang. Syst.*, 15(1):73–132, January 1993.

[4] Martín Abadi and Leslie Lamport. Conjoining specifications. *ACM Trans. Program. Lang. Syst.*, 17(3):507–534, 1995.

[5] Martín Abadi, Leslie Lamport, and Pierre Wolper. Realizable and unrealizable specifications of reactive systems. In Giorgio Ausiello, Mariangiola Dezani-Ciancaglini, and Simona Ronchi Della Rocca, editors, *ICALP*, volume 372 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 1989.

[6] Yael Abarbanel, Ilan Beer, Leonid Gluhovsky, Sharon Keidar, and Yaron Wolfsthal. FoCs – Automatic Generation of Simulation Checkers from Formal Specifications. In E. Emerson and A. Sistla, editors, *Computer Aided Verification*, volume 1855 of *Lecture Notes in Computer Science*, pages 538–542. Springer, Berlin / Heidelberg, 2000.

[7] B. Thomas Adler, Luca de Alfaro, Leandro Dias da Silva, Marco Faella, Axel Legay, Vishwanath Raman, and Pritam Roy. Ticc: A Tool for Interface Compatibility and Composition. In *Proc. of the 18th International Conference on Computer Aided Verification (CAV'06)*, volume 4144 of *Lecture Notes in Computer Science*, pages 59–62. Springer, 2006.

[8] Luca De Alfaro and Thomas A. Henzinger. Interface-based design. In *Engineering Theories of Software Intensive Systems, proceedings of the Marktoberdorf Summer School*. Kluwer, 2004.

[9] Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[10] Rajeev Alur, Limor Fix, and Thomas A. Henzinger. A determinizable class of timed automata. In David L. Dill, editor, *CAV*, volume 818 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 1994.

[11] Rajeev Alur, Limor Fix, and Thomas A. Henzinger. Event-clock automata: A determinizable class of timed automata. *Theor. Comput. Sci.*, 211(1-2):253–273, 1999.

[12] Rajeev Alur and Thomas A. Henzinger. Reactive modules. *Formal Methods in System Design*, 15(1):7–48, 1999.

[13] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *J. ACM*, 49(5):672–713, 2002.

[14] Rajeev Alur, Thomas A. Henzinger, Orna Kupferman, and Moshe Y. Vardi. Alternating refinement relations. In *Proc. of the 9th International Conference on Concurrency Theory (CONCUR'98)*, volume 1466 of *Lecture Notes in Computer Science*, pages 163–178. Springer, 1998.

[15] Rajeev Alur, Thomas A. Henzinger, Freddy Y. C. Mang, Shaz Qadeer, Sriram K. Rajamani, and Serdar Tasiran. MOCHA: modularity in model checking. In *Proc. of the 10th International Conference on Computer Aided Verification (CAV'98), Vancouver, BC, Canada, June 28–July 2, 1998*, volume 1427 of *Lecture Notes in Computer Science*. Alan J. Hu and Moshe Y. Vardi, editors. Springer, 1998, pages 521–525.

[16] Madhukar Anand, Sebastian Fischmeister, and Insup Lee. A comparison of compositional schedulability analysis techniques for hierarchical real-time systems. *ACM Trans. Embedded Comput. Syst.*, 13(1):2, 2013.

[17] Saoussen Anssi, Sébastien Gérard, Stefan Kuntz, and François Terrier. AUTOSAR vs. MARTE for enabling timing analysis of automotive applications. In Iulian Ober and Ileana Ober, editors, *SDL 2011: Integrating System and Software Modeling – 15th International SDL Forum Toulouse, France, July 5–7, 2011. Revised Papers*, volume 7083 of *Lecture Notes in Computer Science*, pages 262–275. Springer, 2011.

[18] Saoussen Anssi, Sara Tucci Piergiovanni, Stefan Kuntz, Sébastien Gérard, and François Terrier. Enabling scheduling analysis for AUTOSAR systems. In *14th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing, ISORC 2011, Newport Beach, California, USA, 28–31 March 2011*, pages 152–159. IEEE Computer Society, 2011.

[19] Adam Antonik, Michael Huth, Kim Guldstrand Larsen, Ulrik Nyman, and Andrzej Wasowski. 20 Years of Modal and Mixed Specifications. *Bulletin of European Association of Theoretical Computer Science*, 1(94), 2008.

[20] Adam Antonik, Michael Huth, Kim Guldstrand Larsen, Ulrik Nyman, and Andrzej Wasowski. Complexity of Decision Problems for Mixed and Modal Specifications. In *FoSSaCS*, pages 112–126, 2008.

[21] Alexandre Arnold, Benoît Boyer, and Axel Legay. Contracts and Behavioral Patterns for SoS: The EU IP DANSE approach. In Kim G. Larsen, Axel Legay, and Ulrik Nyman, editors, *AiSoS*, volume 133 of *EPTCS*, pages 47–66, 2013.

[22] Chetan Arora, Mehrdad Sabetzadeh, Lionel C. Briand, and Frank Zimmer. Requirement boilerplates: Transition from manually-enforced to automatically-verifiable natural language patterns. In Liping Zhao, Julio Cesar Sampaio do Prado Leite, Sam Supakkul, Lawrence Chung, and Ye Wang, editors, *4th IEEE International Workshop on Requirements Patterns, RePa 2014, Karlskrona, Sweden, August 26, 2014*, pages 1–8. IEEE, 2014.

[23] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. MIT Press, Cambridge, 2008.

[24] Felice Balarin, Jerry R. Burch, Luciano Lavagno, Yosinori Watanabe, Roberto Passerone, and Alberto L. Sangiovanni-Vincentelli. Constraints specification at higher levels of abstraction. In *Proceedings of the Sixth IEEE International High-Level Design Validation and Test Workshop (HLDVT01)*, pages 129–133, Monterey, CA, November 7–9, 2001. IEEE Computer Society, Los Alamitos, CA, USA.

[25] Felice Balarin and Roberto Passerone. Functional verification methodology based on formal interface specification and transactor generation. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE06)*, pages 1013–1018, Munich, Germany, March 6–10, 2006. European Design and Automation Association, 3001 Leuven, Belgium.

[26] Felice Balarin and Roberto Passerone. Specification, synthesis and simulation of transactor processes. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26(10):1749–1762, October 2007.

[27] Felice Balarin, Roberto Passerone, Alessandro Pinto, and Alberto L. Sangiovanni-Vincentelli. A formal approach to system level design: Metamodels and unified design environments. In *Proceedings of the Third ACM and IEEE International Conference on Formal Methods and Models for Co-Design (MEMOCODE05)*, pages 155–163, Verona, Italy, July 11–14, 2005. IEEE Computer Society, Los Alamitos, CA, USA.

[28] Krishnakumar Balasubramanian, Aniruddha Gokhale, Gabor Karsai, Janos Sztipanovits, and Sandeep Neema. Developing applications using model-driven design environments. *IEEE Computer*, 39(2):33–40, 2006.

[29] Sebastian S. Bauer, Alexandre David, Rolf Hennicker, Kim Guldstrand Larsen, Axel Legay, Ulrik Nyman, and Andrzej Wasowski. Moving from specifications to contracts in component-based design. In Juan de Lara and Andrea Zisman, editors, *FASE*, volume 7212 of *Lecture Notes in Computer Science*, pages 43–58. Springer, 2012.

[30] Sebastian S. Bauer, Uli Fahrenberg, Line Juhl, Kim G. Larsen, Axel Legay, and Claus R. Thrane. Weighted modal transition systems. *Formal Methods in System Design*, 42(2):193–220, 2013.

[31] Sebastian S. Bauer, Line Juhl, Kim G. Larsen, Axel Legay, and Jirí Srba. Extending modal transition systems with structured labels. *Mathematical Structures in Computer Science*, 22(4):581–617, 2012.

[32] Sebastian S. Bauer, Philip Mayer, Andreas Schroeder, and Rolf Hennicker. On Weak Modal Compatibility, Refinement, and the MIO Workbench. In *Proc. of 16th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'10)*, volume 6015 of *Lecture Notes in Computer Science*, pages 175–189. Springer, 2010.

[33] Andreas Baumgart, Eckard Böde, Matthias Büker, Werner Damm, Günter Ehmen, Tayfun Gezgin, Stefan Henkler, Hardi Hungar, Bernhard Josko, Markus Oertel, Thomas Peikenkamp, Philipp Reinkemeier, Ingo Stierand, and Raphael Weber. Architecture Modeling. Technical report, OFFIS, March 2011. http://ses.informatik.uni-oldenburg.de/download/bib/paper/OFFIS-TR2011_ArchitectureModeling.pdf.

[34] Gerd Behrmann, Alexandre David, and Kim G. Larsen. A tutorial on uppaal. In Marco Bernardo and Flavio Corradini, editors, *Formal Methods for the Design of Real-Time Systems: International School on Formal Methods for the Design of Computer, Communication, and Software Systems, Bertinora, Italy, September 13-18, 2004, Revised Lectures*, pages 200–236, Berlin, Heidelberg, 2004. Springer, Berlin Heidelberg.

[35] Nikola Benes, Jan Kretínský, Kim Guldstrand Larsen, and Jirí Srba. Checking Thorough Refinement on Modal Transition Systems Is EXPTIME-Complete. In Martin Leucker and Carroll Morgan, editors, *ICTAC*, volume 5684 of *Lecture Notes in Computer Science*, pages 112–126. Springer, 2009.

[36] Nikola Benes, Jan Kretínský, Kim Guldstrand Larsen, and Jirí Srba. On determinism in modal transition systems. *Theoretical Computer Science*, 410(41):4026–4043, 2009.

[37] Benoît Caillaud and Jean-Baptiste Raclet. Ensuring Reachability by Design. In *Int. Colloquium on Theoretical Aspects of Computing*, September 2012.

[38] Albert Benveniste and Gérard Berry. The synchronous approach to reactive and real-time systems. *Proceedings of the IEEE*, 79(9):1270–1282, 1991.

[39] Albert Benveniste, Benoît Caillaud, Luca P. Carloni, and Alberto L. Sangiovanni-Vincentelli. Tag machines. In Wayne Wolf, editor, *EMSOFT*, pages 255–263. ACM, 2005.

[40] Albert Benveniste, Benoît Caillaud, Alberto Ferrari, Leonardo Mangeruca, Roberto Passerone, and Christos Sofronis. Multiple viewpoint contract-based specification and design. In *Proceedings of the Software Technology Concertation on Formal Methods for Components and Objects, FMCO'07*, volume 5382 of *Lecture Notes in Computer Science*, pages 200–225. Springer, October 2008.

[41] Albert Benveniste, Benoît Caillaud, and Paul Le Guernic. Compositionality in dataflow synchronous languages: Specification and distributed code generation. *Inf. Comput.*, 163(1):125–171, 2000.

[42] Albert Benveniste, Benoît Caillaud, and Roberto Passerone. Multi-viewpoint state machines for rich component models. In Gabriela Nicolescu and Pieter J. Mosterman, editors, *Model-Based Design for Embedded Systems*, chapter 15, page 487. CRC Press, Taylor and Francis Group, Boca Raton, London, New York, November 2009.

[43] Albert Benveniste, Benoît Caillaud, and Jean-Baptiste Raclet. Application of interface theories to the separate compilation of synchronous programs. In *CDC*, pages 7252–7258. IEEE, 2012.

[44] Albert Benveniste, Paul Caspi, Stephen A. Edwards, Nicolas Halbwachs, Paul Le Guernic, and Robert de Simone. The Synchronous Languages 12 years later. *Proceedings of the IEEE*, 91(1):64–83, 2003.

[45] Albert Benveniste, Dejan Nickovic, and Thomas Henzinger. Compositional Contract Abstraction for System Design. Rapport de recherche RR-8460, INRIA, January 2014. http://hal.inria.fr/hal-00938854.

[46] Luca Benvenuti, Alberto Ferrari, Leonardo Mangeruca, Emanuele Mazzi, Roberto Passerone, and Christos Sofronis. A contract-based formalism for the specification of heterogeneous systems. In *Proceedings of the Forum on Specification, Verification and Design Languages (FDL08)*, pages 142–147, Stuttgart, Germany, September 23–25, 2008.

[47] Gerard Berry. The effectiveness of synchronous languages for the development of safety-critical systems. White paper, Esterel Technologies, 2003.

[48] Nathalie Bertrand, Axel Legay, Sophie Pinchinat, and Jean-Baptiste Raclet. A compositional approach on modal specifications for timed systems. In *Proc. of the 11th International Conference on Formal Engineering Methods (ICFEM'09)*, volume 5885 of *Lecture Notes in Computer Science*, pages 679–697. Springer, 2009.

[49] Nathalie Bertrand, Axel Legay, Sophie Pinchinat, and Jean-Baptiste Raclet. Modal event-clock specifications for timed component-based design. *Sci. Comput. Program.*, 77(12):1212–1234, 2012.

[50] Nathalie Bertrand, Sophie Pinchinat, and Jean-Baptiste Raclet. Refinement and consistency of timed modal specifications. In *Proc. of the 3rd International Conference on Language and Automata Theory and Applications (LATA'09)*, volume 5457 of *Lecture Notes in Computer Science*, pages 152–163. Springer, 2009.

[51] Antoine Beugnard, Jean-Marc Jézéquel, and Noël Plouzeau. Making components contract aware. *IEEE Computer*, 32(7):38–45, 1999.

[52] Dirk Beyer, Arindam Chakrabarti, and Thomas A. Henzinger. Web service interfaces. In Allan Ellis and Tatsuya Hagino, editors, *WWW*, pages 148–159. ACM, 2005.

[53] Purandar Bhaduri and S. Ramesh. Interface synthesis and protocol conversion. *Formal Aspects of Computing*, 20(2):205–224, 2008.

[54] Purandar Bhaduri and Ingo Stierand. A proposal for real-time interfaces in speeds. In *Design, Automation and Test in Europe (DATE'10)*, pages 441–446. IEEE, 2010.

[55] Céline Bigot, Alain Faivre, Jean-Pierre Gallois, Arnault Lapitre, David Lugato, Jean-Yves Pierron, and Nicolas Rapin. Automatic test generation with agatha. In Hubert Garavel and John Hatcliff, editors, *Tools and Algorithms for the Construction and Analysis of Systems: 9th International Conference, TACAS 2003 Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2003 Warsaw, Poland, April 7–11, 2003 Proceedings*, pages 591–596, 2003. Springer, Berlin / Heidelberg.

[56] Simon Bliudze and Joseph Sifakis. The Algebra of Connectors – Structuring Interaction in BIP. *IEEE Trans. Computers*, 57(10):1315–1330, 2008.

[57] R. Bloem and B. Jobstmann. Manual for property-based synthesis tool. Technical Report Prosyd D2.2/3, 2006.

[58] Roderick Bloem, Alessandro Cimatti, Karin Greimel, Georg Hofferek, Robert Könighofer, Marco Roveri, Viktor Schuppan, and Richard Seeber. RATSY – A New Requirements Analysis Tool with Synthesis. In Tayssir Touili, Byron Cook, and Paul Jackson, editors, *CAV*, volume 6174 of *Lecture Notes in Computer Science*, pages 425–429. Springer, 2010.

[59] Eckard Böde, Matthias Büker, Werner Damm, Günter Ehmen, Martin Fränzle, Sebastian Gerwinn, Thomas Goodfellow, Kim Grüttner, Bernhard Josko, Björn Koopmann, Thomas Peikenkamp, Frank Poppen, Philipp Reinkemeier, Michael Siegel, and Ingo Stierand. Design Paradigms for Multi-Layer Time Coherency in ADAS and Automated Driving (MULTIC). In *Forschungsvereinigung Automobiltechnik e.V.*, number 302 in FAT-Schriftenreihe. Verband der Automobilindustrie (VDA), 2017.

[60] Amar Bouali. Xeve, an ESTEREL verification environment. In *Proc. of the 10th International Conference on Computer Aided Verification (CAV'98), Vancouver, BC, Canada, June 28–July 2, 1998*, volume 1427 of *Lecture Notes in Computer Science*. Alan J. Hu and Moshe Y. Vardi, editors. Springer, 1998, pages 500–504.

[61] Gérard Boudol and Kim Guldstrand Larsen. Graphical versus logical specifications. *Theor. Comput. Sci.*, 106(1):3–20, 1992.

[62] Ferenc Bujtor, Sascha Fendrich, Gerald Lüttgen, and Walter Vogler. Nondeterministic modal interfaces. In Giuseppe F. Italiano, Tiziana Margaria-Steffen, Jaroslav Pokorný, Jean-Jacques Quisquater, and Roger Wattenhofer, editors, *SOFSEM 2015: Theory and Practice of Computer Science – 41st International Conference on Current Trends in Theory and Practice of Computer Science, Pec pod Sněžkou, Czech Republic, January 24–29, 2015. Proceedings*, volume 8939 of *Lecture Notes in Computer Science*, pages 152–163. Springer, 2015.

[63] Ferenc Bujtor, Sascha Fendrich, Gerald Lüttgen, and Walter Vogler. Nondeterministic modal interfaces. *Theor. Comput. Sci.*, 642:24–53, 2016.

[64] Ferenc Bujtor and Walter Vogler. Error-pruning in interface automata. In *40th International Conference on Current Trends in Theory and Practice of Computer Science*, SOFSEM 2014, pages 162–173, Nový Smokovec, Slovakia, January 26–29, 2014.

[65] Ferenc Bujtor and Walter Vogler. Failure semantics for modal transition systems. In *14th International Conference on Application of Concurrency to System Design*, ACSD 2014, pages 42–51, Tunis La Marsa, Tunisia, June 23–27, 2014.

[66] Ferenc Bujtor and Walter Vogler. Error-pruning in interface automata. *Theor. Comput. Sci.*, 597:18–39, 2015.

[67] Jerry R. Burch. *Trace Algebra for Automatic Verification of Real-Time Concurrent Systems*. PhD thesis, School of Computer Science, Carnegie Mellon University, August 1992.

[68] Jerry R. Burch, Roberto Passerone, and Alberto L. Sangiovanni-Vincentelli. Overcoming heterophobia: Modeling concurrency in heterogeneous systems. In *Proceedings of the 2nd International Conference on Application of Concurrency to System Design (ACSD01)*, pages 13–32, Newcastle upon Tyne, UK, June 25–29, 2001. IEEE Computer Society, Los Alamitos, CA, USA.

[69] Giorgio C. Buttazzo. *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*. Springer, 2005.

[70] Benoît Caillaud. Mica: A Modal Interface Compositional Analysis Library, October 2011. http://www.irisa.fr/s4/tools/mica.

[71] Benoît Caillaud, Benoît Delahaye, Kim Guldstrand Larsen, Axel Legay, Mikkel Larsen Pedersen, and Andrzej Wasowski. Compositional design methodology with constraint Markov chains. In *Proceedings of the 7th International Conference on Quantitative Evaluation of SysTems (QEST) 2010*. IEEE Computer Society, 2010.

[72] Georgiana Caltais and Bertrand Meyer. On the verification of SCOOP programs. *Sci. Comput. Program.*, 133:194–215, 2017.

[73] Daniela Cancila, Roberto Passerone, Tullio Vardanega, and Marco Panunzio. Toward correctness in the specification and handling of non-functional attributes of high-integrity real-time embedded systems. *IEEE Transactions on Industrial Informatics*, 6(2):181–194, May 2010.

[74] Luca P. Carloni, Roberto Passerone, Alessandro Pinto, and Alberto L. Sangiovanni-Vincentelli. Languages and tools for hybrid systems design. *Foundations and Trends in Electronic Design Automation*, 1(1/2), 2006.

[75] Marco Carloni, Orlando Ferrante, Alberto Ferrari, Gianpaolo Massaroli, Antonio Orazzo, Ida Petrone, and Luigi Velardi. Contract-based analysis for verification of communication-based train control (cbtc) system. In Andrea Bondavalli, Andrea Ceccarelli, and Frank Ortmeier, editors, *Computer Safety, Reliability, and Security: SAFECOMP 2014 Workshops: ASCoMS, DECSoS, DEVVARTS, ISSE, ReSA4CI, SASSUR. Florence, Italy, September 8–9, 2014. Proceedings*, pages 137–146. Springer International Publishing, 2014.

[76] Marco Carloni, Orlando Ferrante, Alberto Ferrari, Gianpaolo Massaroli, Antonio Orazzo, and Luigi Velardi. Contract modeling and verification with formalspecs verifier tool-suite – application to ansaldo sts rapid transit metro system use case. In Floor Koornneef and Coen van Gulijk, editors, *Computer Safety, Reliability, and Security: SAFECOMP 2015 Workshops, ASSURE, DECSoS. ISSE, ReSA4CI, and SASSUR, Delft, The Netherlands, September 22, 2015, Proceedings*, pages 178–189. Springer International Publishing, 2015.

[77] Franck Cassez, Alexandre David, Emmanuel Fleury, Kim G. Larsen, and Didier Lime. Efficient on-the-fly algorithms for the analysis of timed games. In Martín Abadi and Luca de Alfaro, editors, *CONCUR 2005 – Concurrency Theory: 16th International Conference, CONCUR 2005, San Francisco, CA, USA, August 23–26, 2005. Proceedings*, pages 66–80, Berlin, Heidelberg, 2005. Springer, Berlin / Heidelberg.

[78] Karlis Cerans, Jens Chr. Godskesen, and Kim Guldstrand Larsen. Timed Modal Specification – Theory and Tools. In Costas Courcoubetis, editor, *CAV*, volume 697 of *Lecture Notes in Computer Science*, pages 253–267. Springer, 1993.

[79] Pavol Cerný, Martin Chmelik, Thomas A. Henzinger, and Arjun Radhakrishna. Interface simulation distances. In Marco Faella and Aniello Murano, editors, *GandALF*, volume 96 of *EPTCS*, pages 29–42, 2012.

[80] Arindam Chakrabarti. *A Framework for Compositional Design and Analysis of Systems*. PhD thesis, EECS Department, University of California, Berkeley, Dec 2007.

[81] Arindam Chakrabarti, Luca de Alfaro, Thomas A. Henzinger, Marcin Jurdzinski, and Freddy Y. C. Mang. Interface compatibility checking for software modules. In Ed Brinksma and Kim Guldstrand Larsen, editors, *CAV*, volume 2404 of *Lecture Notes in Computer Science*, pages 428–441. Springer, 2002.

[82] Arindam Chakrabarti, Luca de Alfaro, Thomas A. Henzinger, and Freddy Y. C. Mang. Synchronous and Bidirectional Component Interfaces. In *Proc. of the 14th International Conference on Computer Aided Verification (CAV'02)*, volume 2404 of *Lecture Notes in Computer Science*, pages 414–427. Springer, 2002.

[83] Arindam Chakrabarti, Luca de Alfaro, Thomas A. Henzinger, and Mariëlle Stoelinga. Resource Interfaces. In Rajeev Alur and Insup Lee, editors, *EMSOFT*, volume 2855 of *Lecture Notes in Computer Science*, pages 117–133. Springer, 2003.

[84] Edward Y. Chang, Zohar Manna, and Amir Pnueli. Characterization of temporal property classes. In Werner Kuich, editor, *ICALP*, volume 623 of *Lecture Notes in Computer Science*, pages 474–486. Springer, 1992.

[85] Taolue Chen, Chris Chilton, Bengt Jonsson, and Marta Z. Kwiatkowska. A compositional specification theory for component behaviours. In Helmut Seidl, editor, *ESOP*, volume 7211 of *Lecture Notes in Computer Science*, pages 148–168. Springer, 2012.

[86] Chris Chilton, Bengt Jonsson, and Marta Kwiatkowska. An algebraic theory of interface automata. *Theoretical Computer Science*, 549:146–174, 2014.

[87] Chris Chilton, Bengt Jonsson, and Marta Z. Kwiatkowska. Compositional assume-guarantee reasoning for input/output component theories. *Sci. Comput. Program.*, 91:115–137, 2014.

[88] Chris Chilton, Marta Z. Kwiatkowska, and Xu Wang. Revisiting timed specification theories: A linear-time perspective. In Marcin Jurdzinski and Dejan Nickovic, editors, *FORMATS*, volume 7595 of *Lecture Notes in Computer Science*, pages 75–90. Springer, 2012.

[89] E. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999.

[90] Edmund M. Clarke, David E. Long, and Kenneth L. McMillan. Compositional model checking. In *LICS*, pages 353–362, 1989.

[91] Joey W. Coleman and Cliff B. Jones. A structural proof of the soundness of rely/guarantee rules. *J. Log. Comput.*, 17(4):807–841, 2007.

[92] Patrick Cousot and Radhia Cousot. Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints. In Robert M. Graham, Michael A. Harrison, and Ravi Sethi, editors, *POPL*, pages 238–252. ACM, 1977.

[93] Patrick Cousot and Radhia Cousot. Abstract interpretation and application to logic programs. *J. Log. Program.*, 13(2&3):103–179, 1992.

[94] Patrick Cousot and Radhia Cousot. Abstract Interpretation Frameworks. *J. Log. Comput.*, 2(4):511–547, 1992.

[95] Przemyslaw Daca, Thomas Henzinger, Willibald Krenn, and Dejan Nickovic. Compositional specifications for ioco testing. In *ICST*, 2014.

[96] Loris Dal Lago, Orlando Ferrante, Roberto Passerone, and Alberto Ferrari. Dependability assessment of SOA-based CPS with contracts and model-based fault injection. *IEEE Transactions on Industrial Informatics*, 2017.

[97] Werner Damm. Controlling Speculative Design Processes Using Rich Component Models. In *Fifth International Conference on Application of Concurrency to System Design (ACSD 2005)*, pages 118–119, St. Malo, France, June 2005.

[98] Werner Damm and David Harel. LSCs: Breathing life into message sequence charts. *Formal Methods in System Design*, 19(1):45–80, 2001.

[99] Werner Damm, Hardi Hungar, Bernhard Josko, Thomas Peikenkamp, and Ingo Stierand. Using contract-based component specifications for virtual integration testing and architecture design. In *Design, Automation and Test in Europe, DATE 2011, Grenoble, France, March 14–18, 2011*, pages 1023–1028. IEEE, 2011.

[100] Abhijit Davare, Douglas Densmore, Liangpeng Guo, Roberto Passerone, Alberto L. Sangiovanni-Vincentelli, Alena Simalatsar, and Qi Zhu. METROII: A design environment for cyber-physical systems. *ACM Transactions on Embedded Computing Systems*, 12(1s):49:1–49:31, March 2013.

[101] Alexandre David, Kim Guldstrand Larsen, Axel Legay, Ulrik Nyman, and Andrzej Wasowski. ECDAR: An Environment for Compositional Design and Analysis of Real Time Systems. In *Proc. of the 8th International Symposium on Automated Technology for Verification and Analysis (ATVA'10)*, volume 6252 of *Lecture Notes in Computer Science*, pages 365–370, 2010.

[102] Alexandre David, Kim Guldstrand Larsen, Axel Legay, Ulrik Nyman, and Andrzej Wasowski. Timed I/O automata: A complete specification theory for real-time systems. In *Proc. of the 13th ACM International Conference on Hybrid Systems: Computation and Control (HSCC'10)*, pages 91–100. ACM, 2010.

[103] Luca de Alfaro. Game Models for Open Systems. In *Verification: Theory and Practice*, volume 2772 of *Lecture Notes in Computer Science*, pages 269–289. Springer, 2003.

[104] Luca de Alfaro, Leandro Dias da Silva, Marco Faella, Axel Legay, Pritam Roy, and Maria Sorea. Sociable Interfaces. In *Proc. of the 5th International Workshop on Frontiers of Combining Systems (FroCos'05)*, volume 3717 of *Lecture Notes in Computer Science*, pages 81–105. Springer, 2005.

[105] Luca de Alfaro and Thomas A. Henzinger. Interface automata. In *Proc. of the 9th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE'01)*, pages 109–120. ACM Press, 2001.

[106] Luca de Alfaro and Thomas A. Henzinger. Interface theories for component-based design. In Thomas A. Henzinger and Christoph M. Kirsch, editors, *EMSOFT*, volume 2211 of *Lecture Notes in Computer Science*, pages 148–165. Springer, 2001.

[107] Luca de Alfaro, Thomas A. Henzinger, and Mariëlle Stoelinga. Timed Interfaces. In *Proc. of the 2nd International Workshop on Embedded Software (EMSOFT'02)*, volume 2491 of *Lecture Notes in Computer Science*, pages 108–122. Springer, 2002.

[108] Benoît Delahaye. *Modular Specification and Compositional Analysis of Stochastic Systems*. PhD thesis, Université de Rennes 1, 2010.

[109] Benoît Delahaye, Benoît Caillaud, and Axel Legay. Probabilistic Contracts: A Compositional Reasoning Methodology for the Design of Stochastic Systems. In *Proc. 10th International Conference on Application of Concurrency to System Design (ACSD), Braga, Portugal*. IEEE, 2010.

[110] Benoît Delahaye, Benoît Caillaud, and Axel Legay. Probabilistic contracts: a compositional reasoning methodology for the design of systems with stochastic and/or non-deterministic aspects. *Formal Methods in System Design*, 38(1):1–32, 2011.

[111] Benoît Delahaye, Uli Fahrenberg, Thomas A. Henzinger, Axel Legay, and Dejan Nickovic. Synchronous interface theories and time triggered scheduling. In Holger Giese and Grigore Rosu, editors, *FMOODS/FORTE*, volume 7273 of *Lecture Notes in Computer Science*, pages 203–218. Springer, 2012.

[112] Benoît Delahaye, Joost-Pieter Katoen, Kim Guldstrand Larsen, Axel Legay, Mikkel L. Pedersen, Falak Sher, and Andrzej Wasowski. Abstract Probabilistic Automata. In Ranjit Jhala and David A. Schmidt, editors, *VMCAI*, volume 6538 of *Lecture Notes in Computer Science*, pages 324–339. Springer, 2011.

[113] Benoît Delahaye, Kim G. Larsen, Axel Legay, Mikkel L. Pedersen, and Andrzej Wasowski. Consistency and refinement for interval markov chains. *J. Log. Algebr. Program.*, 81(3):209–226, 2012.

[114] Douglas Densmore, Roberto Passerone, and Alberto L. Sangiovanni-Vincentelli. A platform-based taxonomy for ESL design. *IEEE Design and Test of Computers*, 23(5):359–374, May 2006.

[115] AUTOSAR development cooperation. Specification of Timing Extensions. http://www.autosar.org, Release 4.2.1, 2014.

[116] David L. Dill. *Trace Theory for Automatic Hierarchical Verification of Speed-Independent Circuits*. ACM Distinguished Dissertations. MIT Press, 1989.

[117] Nicolás D'Ippolito, Dario Fischbein, Marsha Chechik, and Sebastián Uchitel. MTSA: The Modal Transition System Analyser. In *Proc. of the 23rd IEEE/ACM International Conference on Automated Software Engineering (ASE'08)*, pages 475–476. IEEE, 2008.

[118] Laurent Doyen, Thomas A. Henzinger, Barbara Jobstmann, and Tatjana Petrov. Interface theories with component reuse. In *Proceedings of the 8th ACM & IEEE International conference on Embedded software, EMSOFT'08*, pages 79–88, 2008.

[119] Dumitru Potop-Butucaru and Stephen Edwards and Gérard Berry. *Compiling Esterel*. Springer, 2007. ISBN: 0387706267.

[120] Arvind Easwaran, Insup Lee, Oleg Sokolsky, and Steve Vestal. A compositional scheduling framework for digital avionics systems. In *15th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, RTCSA 2009, Beijing, China, 24-26 August 2009*, pages 371–380. IEEE Computer Society, 2009.

[121] Cindy Eisner. PSL for Runtime Verification: Theory and Practice. In Oleg Sokolsky and Serdar Tasiran, editors, *RV*, volume 4839 of *Lecture Notes in Computer Science*, pages 1–8. Springer, 2007.

[122] Cindy Eisner, Dana Fisman, John Havlicek, Michael J.C. Gordon, Anthony McIsaac, and David Van Campenhout. Formal Syntax and Semantics of PSL – Appendix B of Accellera LRM January 2003. Technical report, IBM, 2003.

[123] Cindy Eisner, Dana Fisman, John Havlicek, Yoad Lustig, Anthony McIsaac, and David Van Campenhout. Reasoning with temporal logic on truncated paths. In Warren A. Hunt Jr. and Fabio Somenzi, editors, *CAV*, volume 2725 of *Lecture Notes in Computer Science*, pages 27–39. Springer, 2003.

[124] Johan Eker, Jörn W. Janneck, Edward A. Lee, Jie Liu, Xiaojun Liu, J. Ludvig, Stephen Neuendorffer, S. Sachs, and Yuhong Xiong. Taming heterogeneity – the ptolemy approach. *Proc. of the IEEE*, 91(1):127–144, 2003.

[125] Avner Engel, Michael Winokur, Gert Dï£¡hmen, and Marc Enzmann. Assumptions / promises – shifting the paradigm in systems-engineering. 18:58–78, 06 2008.

[126] Ling Fang, Takashi Kitamura, Thi Bich Ngoc Do, and Hitoshi Ohsaki. Formal model-based test for autosar multicore rtos. In *Software Testing, Verification and Validation (ICST), 2012 IEEE Fifth International Conference on*, pages 251–259, April 2012.

[127] Nico Feiertag, Kai Richter, Johan Nordlander, and Han Jonsson. A compositional framework for end-to-end path delay calculation of automotive systems under different path semantics. In *IEEE Real-Time System Symposium (RTSS), Workshop on Compositional Theory and Technology for Real-Time Embedded Systems*, November 2008.

[128] Orlando Ferrante, Roberto Passerone, Alberto Ferrari, Leonardo Mangeruca, and Christos Sofronis. BCL: a compositional contract language for embedded systems. In *Proceedings of the 19th IEEE International Conference on Emerging Technologies and Factory Automation*, ETFA14, Barcelona, Spain, September 16–19, 2014.

[129] Orlando Ferrante, Roberto Passerone, Alberto Ferrari, Leonardo Mangeruca, Christos Sofronis, and Massimiliano D'Angelo. Monitor-based run-time contract verification of distributed systems. In *Proceedings of the 9th IEEE International Symposium on Industrial Embedded Systems*, SIES14, Pisa, Italy, June 18–20, 2014.

[130] G. Feuillade. Modal specifications are a syntactic fragment of the Mu-calculus. Research Report RR-5612, INRIA, June 2005.

[131] Dario Fischbein and Sebastián Uchitel. On correct and complete strong merging of partial behaviour models. In *Proc. of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering (SIGSOFT FSE'08)*, pages 297–307. ACM, 2008.

[132] Frédéric Boussinot and Robert de Simone. The Esterel language. *Proceedings of the IEEE*, 79(9):1293–1304, 1991.

[133] Peter Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. Wiley, 2003.

[134] Tayfun Gezgin, Raphael Weber, and Maurice Girod. A Refinement Checking Technique for Contract-Based Architecture Designs. In *Fourth International Workshop on Model Based Architecting and Construction of Embedded Systems, ACES-MB'11*, volume 7167 of *Lecture Notes in Computer Science*. Springer, October 2011.

[135] Jens Chr. Godskesen, Kim Guldstrand Larsen, and Arne Skou. Automatic verification of real-time systems using Epsilon. In Son T. Vuong and Samuel T. Chanson, editors, *PSTV*, volume 1 of *IFIP Conference Proceedings*, pages 323–330. Chapman & Hall, 1994.

[136] G. Gössler and J.-B. Raclet. Modal Contracts for Component-based Design. In *Proc. of the 7th IEEE International Conference on Software Engineering and Formal Methods (SEFM'09)*. IEEE Computer Society Press, November 2009.

[137] Susanne Graf, Roberto Passerone, and Sophie Quinton. Contract-based reasoning for component systems with rich interactions. In Alberto L. Sangiovanni-Vincentelli, Haibo Zeng, Marco Di Natale, and Peter Marwedel, editors, *Embedded Systems Development: From Functional Models to Implementations*, volume 20 of *Embedded Systems*, chapter 8, pages 139–154. Springer, New York, 2014.

[138] Susanne Graf and Sophie Quinton. Contracts for BIP: Hierarchical Interaction Models for Compositional Verification. In John Derrick and Jüri Vain, editors, *FORTE*, volume 4574 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2007.

[139] Orna Grumberg and David E. Long. Model checking and modular verification. *ACM Trans. Program. Lang. Syst.*, 16(3):843–871, 1994.

[140] Liangpeng Guo, Qi Zhu, Pierluigi Nuzzo, Roberto Passerone, Alberto L. Sangiovanni-Vincentelli, and Edward A. Lee. Metronomy: a function-architecture co-simulation framework for timing verification of cyber-physical systems. In *Proceedings of the International Conference on Hardware/Software Codesign and System Synthesis*, CODES14, pages 24:1–24:10, New Delhi, India, October 12–17, 2014. ACM, New York, NY, USA.

[141] Imene Ben Hafaiedh, Susanne Graf, and Sophie Quinton. Reasoning about Safety and Progress Using Contracts. In *Proc. of ICFEM'10*, volume 6447 of *LNCS*, pages 436–451. Springer, 2010.

[142] Nicolas Halbwachs. *Synchronous programming of reactive systems*. Kluwer Academic, 1993.

[143] Nicolas Halbwachs, Fabienne Lagnier, and Christophe Ratel. Programming and Verifying Real-Time Systems by Means of the Synchronous Data-Flow Language Lustre. *IEEE Trans. Software Eng.*, 18(9):785–793, 1992.

[144] Nicolas Halbwachs, Fabienne Lagnier, and Pascal Raymond. Synchronous observers and the verification of reactive systems. In Maurice Nivat, Charles Rattray, Teodor Rus, and Giuseppe Scollo, editors, *AMAST*, Workshops in Computing, pages 83–96. Springer, 1993.

[145] Nicolas Halbwachs and Pascal Raymond. Validation of synchronous reactive systems: From formal verification to automatic testing. In P. S. Thiagarajan and Roland H. C. Yap, editors, *ASIAN*, volume 1742 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 1999.

[146] David Harel. Statecharts: A visual formalism for complex systems. *Sci. Comput. Program.*, 8(3):231–274, 1987.

[147] David Harel, Hillel Kugler, Shahar Maoz, and Itai Segall. Accelerating smart play-out. In Jan van Leeuwen, Anca Muscholl, David Peleg, Jaroslav Pokorný, and Bernhard Rumpe, editors, *SOFSEM*, volume 5901 of *Lecture Notes in Computer Science*, pages 477–488. Springer, 2010.

[148] David Harel, Robby Lampert, Assaf Marron, and Gera Weiss. Model-checking behavioral programs. In Samarjit Chakraborty, Ahmed Jerraya, Sanjoy K. Baruah, and Sebastian Fischmeister, editors, *EMSOFT*, pages 279–288. ACM, 2011.

[149] David Harel and Rami Marelly. *Come, Let's Play: Scenario-Based Programming Using LSCs and the Play-Engine*. Springer-Verlag, 2003. http://www.wisdom.weizmann.ac.il/~harel/ComeLetsPlay.pdf.

[150] David Harel, Assaf Marron, and Gera Weiss. Behavioral programming. *Commun. ACM*, 55(7):90–100, 2012.

[151] David Harel, Assaf Marron, Guy Wiener, and Gera Weiss. Behavioral programming, decentralized control, and multiple time scales. In Cristina Videira Lopes, editor, *SPLASH Workshops*, pages 171–182. ACM, 2011.

[152] David Harel and Amir Pnueli. On the development of reactive systems. In K. R. Apt, editor, *Logic and Models for Verification and Specification of Concurrent Systems*, volume F13 of *NATO ASI Series*, pages 477–498. Springer-Verlag, 1985.

[153] H. Heinecke, W. Damm, B. Josko, A. Metzner, H. Kopetz, A. Sangiovanni-Vincentelli, and M. Di Natale. Software Components for Reliable Automotive Systems. In *Design, Automation and Test in Europe, 2008. DATE '08*, pages 549–554, March 2008.

[154] Thomas A. Henzinger and Dejan Nickovic. Independent implementability of viewpoints. In Radu Calinescu and David Garlan, editors, *Monterey Workshop*, volume 7539 of *Lecture Notes in Computer Science*, pages 380–395. Springer, 2012.

[155] C. A. R. Hoare. An axiomatic basis for computer programming. *Commun. ACM*, 12(10):576–580, 1969.

[156] Alan J. Hu and Moshe Y. Vardi, editors. *Computer Aided Verification, 10th International Conference, CAV'98, Vancouver, BC, Canada, June 28–July 2, 1998, Proceedings*, volume 1427 of *Lecture Notes in Computer Science*. Springer, 1998.

[157] Yanhong Huang, João F. Ferreira, Guanhua He, Shengchao Qin, and Jifeng He. Deadline analysis of AUTOSAR OS periodic tasks in the presence of interrupts. In Lindsay Groves and Jing Sun, editors, *Formal Methods and Software Engineering – 15th International Conference on Formal Engineering Methods, ICFEM 2013, Queenstown, New Zealand, October 29–November 1, 2013, Proceedings*, volume 8144 of *Lecture Notes in Computer Science*, pages 165–181. Springer, 2013.

[158] INCOSE. Incose systems engineering handbook, 2010. http://www.incose.org/ProductsPubs/products/sehandbook.aspx.

[159] Cliff B. Jones. Specification and design of (parallel) programs. In *IFIP Congress*, pages 321–332, 1983.

[160] Cliff B. Jones. Wanted: a compositional approach to concurrency. In *Programming Methodology,* Annabelle McIver and Caroll Morgan, Eds., pages 1–15. Springer Verlag, 2000.

[161] Cliff B. Jones and Ian J. Hayes. Possible values: Exploring a concept for concurrency. *J. Log. Algebr. Meth. Program.*, 85(5):972–984, 2016.

[162] Cliff B. Jones, Ian J. Hayes, and Robert J. Colvin. Balancing expressiveness in formal approaches to concurrency. *Formal Asp. Comput.*, 27(3):475–497, 2015.

[163] Clifford B. Jones. *Systematic software development using VDM (2. ed.)*. Prentice Hall International Series in Computer Science. Prentice Hall, 1991.

[164] Bengt Jonsson and Kim Guldstrand Larsen. Specification and refinement of probabilistic processes. In *Logic in Computer Science (LICS)*, pages 266–277. IEEE Computer, 1991.

[165] Bernhard Josko, Qin Ma, and Alexander Metzner. Designing embedded systems using heterogeneous rich components. 18:558–576, 06 2008.

[166] Line Juhl, Kim G. Larsen, and Jirí Srba. Modal transition systems with weight intervals. *J. Log. Algebr. Program.*, 81(4):408–421, 2012.

[167] Gilles Kahn. The Semantics of Simple Language for Parallel Programming. In *IFIP Congress*, pages 471–475, 1974.

[168] S. Karris. *Introduction to Simulink with Engineering Applications*. Orchard Publications, 2006.

[169] Gabor Karsai, Janos Sztipanovitz, Akos Ledczi, and Ted Bapty. Model-integrated development of embedded software. *Proceedings of the IEEE*, 91(1), January 2003.

[170] Orna Kupferman and Moshe Y. Vardi. Modular model checking. In Willem P. de Roever, Hans Langmaack, and Amir Pnueli, editors, *COMPOS*, volume 1536 of *Lecture Notes in Computer Science*, pages 381–401. Springer, 1997.

[171] Kai Lampka, Simon Perathoner, and Lothar Thiele. Analytic real-time analysis and timed automata: a hybrid methodology for the performance analysis of embedded real-time systems. *Design Automation for Embedded Systems*, 14(3):193–227, 2010.

[172] Kai Lampka, Simon Perathoner, and Lothar Thiele. Component-based system design: analytic real-time interfaces for state-based component implementations. *STTT*, 15(3):155–170, 2013.

[173] Leslie Lamport. Proving the correctness of multiprocess programs. *IEEE Trans. Software Eng.*, 3(2):125–143, 1977.

[174] Kim G. Larsen, Axel Legay, Louis-Marie Traonouez, and Andrzej Wasowski. Robust synthesis for real-time systems. *Theor. Comput. Sci.*, 515:96–122, 2014.

[175] Kim Guldstrand Larsen. Modal specifications. In *Automatic Verification Methods for Finite State Systems*, volume 407 of *Lecture Notes in Computer Science*, pages 232–246. Springer, 1989.

[176] Kim Guldstrand Larsen, Ulrik Nyman, and Andrzej Wasowski. Interface Input/Output Automata. In Jayadev Misra, Tobias Nipkow, and Emil Sekerinski, editors, *FM*, volume 4085 of *Lecture Notes in Computer Science*, pages 82–97. Springer, 2006.

[177] Kim Guldstrand Larsen, Ulrik Nyman, and Andrzej Wasowski. Modal I/O Automata for Interface and Product Line Theories. In *Programming Languages and Systems, 16th European Symposium on Programming, ESOP'07*, volume 4421 of *Lecture Notes in Computer Science*, pages 64–79. Springer, 2007.

[178] Kim Guldstrand Larsen, Ulrik Nyman, and Andrzej Wasowski. On Modal Refinement and Consistency. In *Proc. of the 18th International Conference on Concurrency Theory (CONCUR'07)*, pages 105–119. Springer, 2007.

[179] Kim Guldstrand Larsen, Bernhard Steffen, and Carsten Weise. A constraint oriented proof methodology based on modal transition systems. In *Proc. of the 1st International Workshop on Tools and Algorithms for Construction and Analysis of Systems (TACAS'95)*, volume 1019 of *Lecture Notes in Computer Science*, pages 17–40. Springer, 1995.

[180] Kim Guldstrand Larsen and Bent Thomsen. A Modal Process Logic. In *Proceedings of the Third Annual Symposium on Logic in Computer Science (LICS'88)*, pages 203–210. IEEE, 1988.

[181] Kim Guldstrand Larsen and L. Xinxin. Equation solving using modal transition systems. In *Proceedings of the 5th Annual IEEE Symp. on Logic in Computer Science, LICS'90*, pages 108–117. IEEE Computer Society Press, 1990.

[182] Hoa Thi Thieu Le and Roberto Passerone. Refinement-based synthesis of correct contract model decompositions. In *Proceedings of the 12th ACM-IEEE International Conference on Formal Methods and Models for System Design*, MEMOCODE14, Lausanne, Switzerland, October 19–21, 2014.

[183] Hoa Thi Thieu Le, Roberto Passerone, Uli Fahrenberg, and Axel Legay. A tag contract framework for heterogeneous systems. In *Proceedings of the 12th International Workshop on Foundations of Coordination Languages and Self Adaptive Systems*, FOCLASA13, Malaga, Spain, September 11, 2013.

[184] Hoa Thi Thieu Le, Roberto Passerone, Uli Fahrenberg, and Axel Legay. Tag machines for modeling heterogeneous systems. In *Proceedings of the 13th International Conference on Application of Concurrency to System Design*, ACSD13, pages 186–195, Barcelona, Spain, July 8–10, 2013.

[185] Thi Thieu Hoa Le, Roberto Passerone, Uli Fahrenberg, and Axel Legay. Contract-based requirement modularization via synthesis of correct decompositions. *ACM Transactions on Embedded Computing Systems*, 15(2):33:1–33:26, 2016.

[186] Thi Thieu Hoa Le, Roberto Passerone, Uli Fahrenberg, and Axel Legay. A tag contract framework for modeling heterogeneous systems. *Science of Computer Programming*, 115–116:225–246, 2016.

[187] Ákos Lédeczi, Arpad Bakay, Miklos Maroti, Péter Völgyesi, Greg Nordstrom, Jonathan Sprinkle, and Gabor Karsai. Composing domain-specific design environments. *IEEE Computer*, 34(11):44–51, 2001.

[188] Akos Ledeczi, Miklos Maroti, Arpad Bakay, Gabor Karsai, Jason Garrett, Charles Thomason, Greg Nordstrom, Jonathan Sprinkle, and Peter Volgyesi. The generic modeling environment. In *Proceedings of the IEEE International Workshop on Intelligent Signal Processing (WISP2001)*, Budapest, Hungary, May 24–25 2001.

[189] Jane W.S. Liu. *Real-Time Systems*. Prentice Hall, 2000.

[190] Martin Lukasiewycz, Reinhard Schneider, Dip Goswami, and Samarjit Chakraborty. Modular scheduling of distributed heterogeneous time-triggered automotive systems. In *Proceedings of the 17th Asia and South Pacific Design Automation Conference, ASP-DAC 2012, Sydney, Australia, January 30–February 2, 2012*, pages 665–670. IEEE, 2012.

[191] Gerald Lüttgen and Walter Vogler. Modal interface automata. *Logical Methods in Computer Science*, 9(3), 2013.

[192] Gerald Lüttgen and Walter Vogler. Richer interface automata with optimistic and pessimistic compatibility. *ECEASST*, 66, 2013.

[193] Nancy A. Lynch. Input/output automata: Basic, timed, hybrid, probabilistic, dynamic, .. In Roberto M. Amadio and Denis Lugiez, editors, *CONCUR*, volume 2761 of *Lecture Notes in Computer Science*, pages 187–188. Springer, 2003.

[194] Nancy A. Lynch and Eugene W. Stark. A proof of the kahn principle for input/output automata. *Inf. Comput.*, 82(1):81–92, 1989.

[195] Zohar Manna and Amir Pnueli. *The temporal logic of reactive and concurrent systems – specification*. Springer, 1992.

[196] Zohar Manna and Amir Pnueli. *Temporal verification of reactive systems: Safety*. Springer, 1995.

[197] Hervé Marchand, Patricia Bournai, Michel Le Borgne, and Paul Le Guernic. Synthesis of Discrete-Event Controllers Based on the Signal Environment. *Discrete Event Dynamic Systems*, 10(4):325–346, 2000.

[198] Hervé Marchand and Mazen Samaan. Incremental Design of a Power Transformer Station Controller Using a Controller Synthesis Methodology. *IEEE Trans. Software Eng.*, 26(8):729–741, 2000.

[199] Shanmuga Priya Marimuthu and Samarjit Chakraborty. A framework for compositional and hierarchical real-time scheduling. In *12th IEEE Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2006), 16–18 August 2006, Sydney, Australia*, pages 91–96. IEEE Computer Society, 2006.

[200] Bertrand Meyer. Applying "design by contract". *IEEE Computer*, 25(10):40–51, October 1992.

[201] Bertrand Meyer. *Touch of Class: Learning to Program Well Using Object Technology and Design by Contract*. Springer, 2009.

[202] Antoine Miné. Weakly Relational Numerical Abstract Domains. Phd, Ecole Normale Supérieure, département d'informatique, Dec 2004. http://www.di.ens.fr/~mine/these/these-color.pdf.

[203] M.W. Maier. Architecting Principles for Systems of Systems. *Systems Engineering*, 1(4):267–284, 1998.

[204] Walid A. Najjar, Edward A. Lee, and Guang R. Gao. Advances in the dataflow computational model. *Parallel Computing*, 25(13-14):1907–1929, 1999.

[205] Radu Negulescu. *Process Spaces and the Formal Verification of Asynchronous Circuits*. PhD thesis, University of Waterloo, Canada, 1998.

[206] Nicolas Halbwachs and Paul Caspi and Pascal Raymond and Daniel Pilaud. The synchronous data flow programming language Lustre. *Proceedings of the IEEE*, 79(9):1305–1320, 1991.

[207] P. Nuzzo, A. Sangiovanni-Vincentelli, X. Sun, and A. Puggelli. Methodology for the design of analog integrated interfaces using contracts. *IEEE Sensors Journal*, 12(12):3329–3345, Dec. 2012.

[208] Object Management Group (OMG). System modeling language specification v1.1. Technical report, OMG, 2008.

[209] Object constraint language, version 2.0. OMG Available Specification formal/06-05-01, Object Management Group, May 2006.

[210] The Design Automation Standards Committee of the IEEE Computer Society, editor. *1850-2010 – IEEE Standard for Property Specification Language (PSL)*. IEEE Computer Society, 2010.

[211] J. Hudak P. Feiler, D. Gluch. The Architecture Analysis and Design Language (AADL): An Introduction. *Software Engineering Institute (SEI) Technical Note, CMU/SEI-2006-TN-011*, February 2006.

[212] Roberto Passerone. *Semantic Foundations for Heterogeneous Systems*. PhD thesis, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, CA 94720, May 2004.

[213] Roberto Passerone, Luca de Alfaro, Thomas A. Henzinger, and Alberto L. Sangiovanni-Vincentelli. Convertibility verification and converter synthesis: Two faces of the same coin. In *Proceedings of the 20th IEEE/ACM International Conference on Computer-Aided Design (ICCAD02)*, pages 132–139, San Jose, California, November 10–14, 2002. IEEE Computer Society, Los Alamitos, CA, USA.

[214] Roberto Passerone, Jerry R. Burch, and Alberto L. Sangiovanni-Vincentelli. Refinement preserving approximations for the design and verification of heterogeneous systems. *Formal Methods in System Design*, 31(1):1–33, August 2007.

[215] Roberto Passerone, Imene Ben Hafaiedh, Susanne Graf, Albert Benveniste, Daniela Cancila, Arnaud Cuccuru, Sébastien Gérard, Francois Terrier, Werner Damm, Alberto Ferrari, Leonardo Mangeruca, Bernhard Josko, Thomas Peikenkamp, and Alberto Sangiovanni-Vincentelli. Metamodels in Europe: Languages, tools, and applications. *IEEE Design and Test of Computers*, 26(3):38–53, May/June 2009.

[216] Paul Le Guernic and Thierry Gautier and Michel Le Borgne and Claude Le Maire. Programming real-time applications with Signal. *Proceedings of the IEEE*, 79(9):1321–1336, 1991.

[217] Marie-Agnès Peraldi-Frati, Arda Goknil, Morayo Adedjouma, and Pierre Yves Gueguen. Modeling a BSG-E automotive system with the timing augmented description language. In Tiziana Margaria and Bernhard Steffen, editors, *Leveraging Applications of Formal Methods, Verification and Validation. Applications and Case Studies – 5th International Symposium, ISoLA 2012, Heraklion, Crete, Greece, October 15–18, 2012, Proceedings, Part II*, volume 7610 of *Lecture Notes in Computer Science*, pages 111–125. Springer, 2012.

[218] S. Perathoner, K. Lampka, and L. Thiele. Composing heterogeneous components for system-wide performance analysis. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2011*, pages 1–6, march 2011.

[219] Linh T. X. Phan, Jaewoo Lee, Arvind Easwaran, Vinay Ramaswamy, Sanjian Chen, Insup Lee, and Oleg Sokolsky. CARTS: a tool for compositional analysis of real-time systems. *SIGBED Review*, 8(1):62–63, 2011.

[220] I. Pill, B. Jobstmann, R. Bloem, R. Frank, M. Moulin, B. Sterin, M. Roveri, and S. Semprini. Property simulation. Technical Report Prosyd D1.2/1, 2005.

[221] Alessandro Pinto, Alvise Bonivento, Alberto L. Sangiovanni-Vincentelli, Roberto Passerone, and Marco Sgroi. System level design paradigms: Platform-based design and communication synthesis. *ACM Transactions on Design Automation of Electronic Systems*, 11(3):537–563, July 2006.

[222] Klaus Pohl, Manfred Broy, Heinrich Daembkes, and Harald Hï£¡nninger. *Advanced Model-Based Engineering of Embedded Systems: Extensions of the SPES 2020 Methodology*. 01 2016.

[223] Terry Quatrani. *Visual modeling with Rational Rose 2000 and UML (2nd ed.)*. Addison-Wesley Longman Ltd., Essex, UK, UK, 2000.

[224] Jean-Baptiste Raclet. *Quotient de spécifications pour la réutilisation de composants*. PhD thesis, Ecole doctorale Matisse, université de Rennes 1, November 2007.

[225] Jean-Baptiste Raclet. Residual for Component Specifications. In *Proc. of the 4th International Workshop on Formal Aspects of Component Software (FACS'07)*, 2007.

[226] Jean-Baptiste Raclet, Eric Badouel, Albert Benveniste, Benoît Caillaud, Axel Legay, and Roberto Passerone. Modal interfaces: Unifying interface automata and modal specifications. In *Proceedings of the Ninth International Conference on Embedded Software (EMSOFT09)*, pages 87–96, Grenoble, France, October 12–16, 2009.

[227] Jean-Baptiste Raclet, Eric Badouel, Albert Benveniste, Benoît Caillaud, Axel Legay, and Roberto Passerone. A modal interface theory for component-based design. *Fundamenta Informaticae*, 108(1-2):119–149, 2011.

[228] Jean-Baptiste Raclet, Eric Badouel, Albert Benveniste, Benoît Caillaud, and Roberto Passerone. Why are modalities good for interface theories? In *Proc. of the 9th International Conference on Application of Concurrency to System Design (ACSD'09)*. IEEE Computer Society Press, 2009.

[229] Jan Reineke and Stavros Tripakis. Basic problems in multi-view modeling. In *TACAS*, 2014.

[230] Philipp Reinkemeier, Albert Benveniste, Werner Damm, and Ingo Stierand. Contracts for schedulability analysis. In Sriram Sankaranarayanan and Enrico Vicario, editors, *Formal Modeling and Analysis of Timed Systems – 13th International Conference, FORMATS 2015, Madrid, Spain, September 2–4, 2015, Proceedings*, volume 9268 of *Lecture Notes in Computer Science*, pages 270–287. Springer, 2015.

[231] Philipp Reinkemeier and Ingo Stierand. Compositional timing analysis of real-time systems based on resource segregation abstraction. In Gunar Schirner, Marcelo Götz, Achim Rettberg, Mauro Cesar Zanella, and Franz J. Rammig, editors, *Embedded Systems: Design, Analysis and Verification – 4th IFIP TC 10 International Embedded Systems Symposium, IESS 2013, Paderborn, Germany, June 17-19, 2013. Proceedings*, volume 403 of *IFIP Advances in Information and Communication Technology*, pages 181–192. Springer, 2013.

[232] Philipp Reinkemeier and Ingo Stierand. Real-Time Contracts – A Contract Theory Considering Resource Supplies and Demands. Reports of SFB/TR 14 AVACS 100, SFB/TR 14 AVACS, July 2014. http://www.avacs.org.

[233] Richard Payne and John Fitzgerald. Evaluation of Architectural Frameworks Supporting Contract-Based Specification. Technical Report CS-TR-1233, Computing Science, Newcastle University, UK, Dec 2010. available from http://www.cs.ncl.ac.uk/publications/trs/papers/1233.pdf.

[234] Robert W. Floyd. Assigning meaning to programs. In J.T. Schwartz, editor, *Proceedings of Symposium on Applied Mathematics*, volume 19, pages 19–32, 1967.

[235] A. Sangiovanni-Vincentelli, S. Shukla, J. Sztipanovits, G. Yang, and D. Mathaikutty. Metamodeling: An emerging representation paradigm for system-level design. *IEEE Design and Test of Computers*, 26(3):54–69, May/June 2009.

[236] Alberto Sangiovanni-Vincentelli, Werner Damm, and Roberto Passerone. Taming Dr. Frankenstein: Contract-based design for cyber-physical systems. *European Journal of Control*, 18(3):217–238, 2012.

[237] D. Schmidt. Model-driven engineering. *IEEE Computer*, pages 25–31, February 2006.

[238] Lui Sha, Tarek Abdelzaher, Karl-Erik Årzén, Anton Cervin, Theodore Baker, Alan Burns, Giorgio Buttazzo, Marco Caccamo, John Lehoczky, and Aloysius K. Mok. Real Time Scheduling Theory: A Historical Perspective. *Real-Time Systems*, 28(2-3):101–155, 2004.

[239] Mary Sheeran, Satnam Singh, and Gunnar Stålmarck. Checking safety properties using induction and a sat-solver. In Warren A. Hunt and Steven D. Johnson, editors, *Formal Methods in Computer-Aided Design: Third International Conference, FMCAD 2000 Austin, TX, USA, November 1–3, 2000 Proceedings*, pages 127–144, Berlin, Heidelberg, 2000. Springer, Berlin / Heidelberg.

[240] Mary Sheeran and Gunnar Stålmarck. A tutorial on stålmarck's proof procedure for propositional logic. *Formal Methods in System Design*, 16(1):23–58, Jan 2000.

[241] Insik Shin and Insup Lee. Compositional real-time scheduling framework. In *Proceedings of the 25th IEEE Real-Time Systems Symposium (RTSS 2004), 5-8 December 2004, Lisbon, Portugal*, pages 57–67. IEEE Computer Society, 2004.

[242] German Sibay, Sebastian Uchitel, and Víctor Braberman. Existential Live Sequence Charts Revisited. In *ICSE 2008: 30th International Conference on Software Engineering*. ACM, May 2008.

[243] Joseph Sifakis. Component-Based Construction of Heterogeneous Real-Time Systems in Bip. In Giuliana Franceschinis and Karsten Wolf, editors, *Petri Nets*, volume 5606 of *Lecture Notes in Computer Science*, page 1. Springer, 2009.

[244] Eugene W. Stark. A proof technique for rely/guarantee properties. In S. N. Maheshwari, editor, *FSTTCS*, volume 206 of *Lecture Notes in Computer Science*, pages 369–391. Springer, 1985.

[245] Ingo Stierand, Philipp Reinkemeier, and Purandar Bhaduri. Virtual integration of real-time systems based on resource segregation abstraction. In Axel Legay and Marius Bozga, editors, *Formal Modeling and Analysis of Timed Systems – 12th International Conference, FORMATS 2014, Florence, Italy, September 8–10, 2014. Proceedings*, volume 8711 of *Lecture Notes in Computer Science*, pages 206–221. Springer, 2014.

[246] Ingo Stierand, Philipp Reinkemeier, Tayfun Gezgin, and Purandar Bhaduri. Real-time scheduling interfaces and contracts for the design of distributed embedded systems. In *8th IEEE International Symposium on Industrial Embedded Systems, SIES 2013, Porto, Portugal, June 19–21, 2013*, pages 130–139. IEEE, 2013.

[247] Nikolay Stoimenov, Samarjit Chakraborty, and Lothar Thiele. Interface-based design of real-time systems. In Samarjit Chakraborty and Jörg Eberspächer, editors, *Advances in Real-Time Systems (to Georg Färber on the occasion of his appointment as Professor Emeritus at TU München after leading the Lehrstuhl für Realzeit-Computersysteme for 34 illustrious years)*, pages 83–101. Springer, 2012.

[248] Walter Storm. Solving sudoku using simulink design verifier – a model checking example. In *AIAA InfotechAerospace 2010*. American Institute of Aeronautics and Astronautics, April 2010.

[249] Lothar Thiele, Ernesto Wandeler, and Nikolay Stoimenov. Real-time interfaces for composing real-time systems. In Sang Lyul Min and Wang Yi, editors, *Proceedings of the 6th ACM & IEEE International conference on Embedded software, EMSOFT 2006, October 22–25, 2006, Seoul, Korea*, pages 34–43. ACM, 2006.

[250] Stavros Tripakis, Ben Lickly, Thomas A. Henzinger, and Edward A. Lee. On relational interfaces. In *Proc. of the 9th ACM & IEEE International conference on Embedded software (EMSOFT'09)*, pages 67–76. ACM, 2009.

[251] Stavros Tripakis, Ben Lickly, Thomas A. Henzinger, and Edward A. Lee. A theory of synchronous relational interfaces. *ACM Trans. Program. Lang. Syst.*, 33(4):14, 2011.

[252] Sebastián Uchitel and Marsha Chechik. Merging partial behavioural models. In *Proc. of the 12th ACM SIGSOFT International Symposium on Foundations of Software Engineering (SIGSOFT FSE'10)*, pages 43–52. ACM, 2004.

[253] Machiel van der Bijl, Arend Rensink, and Jan Tretmans. Compositional testing with ioco. In Alexandre Petrenko and Andreas Ulrich, editors, *Formal Approaches to Software Testing, Third International Workshop on Formal Approaches to Testing of Software, FATES 2003, Montreal, Quebec, Canada, October 6th, 2003*, volume 2931 of *Lecture Notes in Computer Science*, pages 86–100. Springer, 2003.

[254] Ernesto Wandeler and Lothar Thiele. Interface-based design of real-time systems with hierarchical scheduling. In *12th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2006), 4–7 April 2006, San Jose, California, USA*, pages 243–252. IEEE Computer Society, 2006.

[255] Jos Warmer and Anneke Kleppe. *The Object Constraint Language: Getting Your Models Ready for MDA*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2003.

[256] Elizabeth S. Wolf. *Hierarchical Models of Synchronous Circuits for Formal Verification and Substitution*. PhD thesis, Department of Computer Science, Stanford University, October 1995.