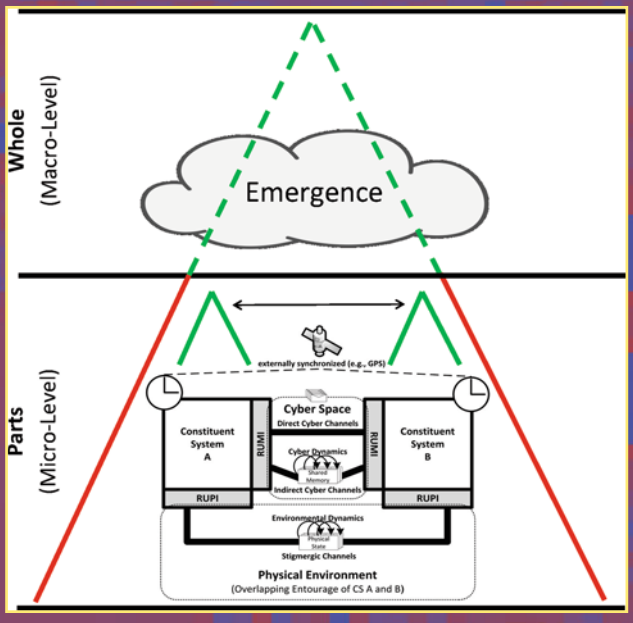
State-of-the-Art Survey

LNCS 10099

Andrea Bondavalli · Sara Bouchenak
Hermann Kopetz (Eds.)

Cyber-Physical Systems of Systems

Foundations – A Conceptual Model and Some Derivations: The AMADEOS Legacy



Springer Open

Lecture Notes in Computer Science 10099

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

More information about this series at http://www.springer.com/series/7408

Andrea Bondavalli · Sara Bouchenak
Hermann Kopetz (Eds.)

Cyber-Physical Systems of Systems

Foundations – A Conceptual Model and Some Derivations: The AMADEOS Legacy

Springer Open

Editors
Andrea Bondavalli
Department of Mathematics and Informatics
University of Florence
Florence
Italy

Sara Bouchenak
INSA Lyon - LIRIS
Lyon
France

Hermann Kopetz
Institute of Computer Engineering
Vienna University of Technology
Vienna
Austria

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-319-47589-9 ISBN 978-3-319-47590-5 (eBook)
DOI 10.1007/978-3-319-47590-5

Library of Congress Control Number: 2016958992

LNCS Sublibrary: SL2 – Programming and Software Engineering

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Foreword

"We are called to be architects of the future, not its victims"
Richard Buckminster-Fuller

This book is about "systems-of-systems". If you search in Google for this term, you get *176,000,000 results in 0.60 seconds* (20.9.2016). This clearly shows the importance and vibrancy of this field! However, it also shows the wide and diverging variety of viewpoints, concepts, and opinions related to systems-of-systems.

Technical systems-of-systems – in the form of networked, independent constituent computing systems temporarily collaborating to achieve a well-defined objective – form the backbone of most of today's infrastructure. The energy grid, most transportation systems, the global banking industry, the water-supply system, military equipment, many embedded systems, and a great number more, strongly depend on systems-of-systems. The correct operation and continuous availability of these underlying systems-of-systems are fundamental for the functioning of our modern society.

Looking at such systems-of-systems, one property clearly stands out: *complexity*. Modern systems-of-systems have reached a degree of structural and behavioral complexity that makes it difficult – in many cases impossible – to understand them. As a consequence, a massive engineering effort and monetary investment are required to design, implement, maintain and evolve many of today's systems-of-systems. Owing to new properties that are introduced when systems-of-systems are formed – such as emergent behavior, especially *unpredictable* emergent behavior – a new element of *risk* is also introduced. Because our dependence on these growing systems-of-systems is nearly total, we need reliable methods, principles, and tools to manage the evolution of our systems-of-systems in today's world of growing complexity, relentless change, and merciless uncertainty. This book is a move forward on this interesting and important path.

The first important step of achieving this objective is the development of an understandable and consistent *set of concepts* describing the systems-of-systems domain. This is not the case in the current state of the art: Therefore, this is the first valuable contribution of this book for the community.

Systems-of-systems become alive by exchanging information and control between their constituent systems and the physical environment via *interfaces*. Interfaces are responsible for many properties in systems-of-systems and therefore need detailed attention: This is the second impressive result of the book – a thorough treatment of interface definition, specification, implementation, and monitoring.

The most fascinating and disturbing phenomenon in systems-of-systems is *emergence*: Behavior or properties that only become active or visible when the constituent systems start cooperating. Emergence has been studied in many contexts and with many objectives: Here we find a consistent theory with important novel concepts, which is applicable to many systems. This is a major research achievement.

Next, a rich *conceptual model* of generic systems-of-systems, divided into ten viewpoints, is developed. The conceptual model is supported by a SysML profile, covering the ten viewpoints. Especially interesting and innovative parts are the representation and use of *time* in the SoS time package and the handling of *emergence* in the SoS emergence package. As additional material, a three-level architecture description framework for generic systems-of-systems – again well implementing the ten viewpoints – is presented, which can be used in a commercially available graphical tool. This part greatly helps the understanding of systems-of-systems development and documentation.

One of the very strong points of this book is the presentation of time and *synchronized timing* in systems-of-systems. This aspect has not been covered with sufficient theoretical rigor in the existing literature. Topics are global time base and resilient clocks – presenting both innovative research and an excellent tutorial.

Many systems-of-systems are not static, but must adapt to changing requirements, be it changing business requirements or in response to changing environmental or operational parameters of the constituent systems. Timely adaptation of systems-of-systems requires a property that is called *dynamicity*. A theory – based on autonomic computing – and implementation patterns for coping with dynamicity are presented.

This volume is on a research and technology level and throughout the book, interesting and illustrative examples can be found.

The book grew out of a sequence of EU-funded projects of which the described project AMADEOS was the culmination. Most of the book shows an admirable maturity, both of the material and the presentation, and is certainly a source of much more fruitful research.

I wish the reader as much satisfaction in reading this rewarding book as the authors had during writing it!

September 2016 Frank J. Furrer

Preface

The general availability of a powerful communication infrastructure (e.g., the Internet) makes it possible to interconnect existing self-contained computer-systems— called *Constituent Systems (CS)*— that already provide a useful service to their users. Such a composition of a set of independent and autonomous systems that brings about novel services to their stakeholders is called a *System-of-Systems* (SoS). The purpose of building an SoS out of CSs is to realize *emergent services* that go beyond the services provided by any of the isolated CSs. Emergence is thus at the core of SoS engineering.

Consider, for example, a local bank terminal that is connected to the worldwide ATM system. This connection enables the novel service of worldwide accessibility of a local bank account. A tourist in a remote country can withdraw money, denoted in the currency of the host country, from his/her home bank that displays the transaction in the currency of the home bank. Since the exchange rate of the currencies is time dependent, the monetary value of this multi-currency transaction depends on the instant when the transaction is executed. This simple example shows that new issues, such as the appropriate representation of information in differing contexts or the time when an action takes place, play an important role in such an SoS.

The vision of the Internet of Things (IoT) assumes that the networked connection of *smart things*, i.e., *Cyber-Physical System (CPS)* with sensors and actuators that observe and directly influence the physical environment, has the potential to provide disruptive novel services to our society. We call such an integration of stand-alone CPSs that provides services that go beyond the services of any of its isolated CPSs a *Cyber-Physical System of Systems (CPSoS)*.

Consider, for example, a smart grid where a multitude of autonomous energy producers and energy consumers, controlled by their local CPSs and the control systems of the utility companies cooperate to provide a smooth flow of electric energy from producers to consumers. Despite the fact that such a possibly gigantic CPSoS is in a continuous state of evolution, it must provide a dependable service 24 hours a day, seven days a week.

This book on *Systems of Systems* documents the main insights on CPSoS that we gained during our work in the European research project AMADEOS (acronym for Architecture for Multi-criticality Agile Dependable Evolutionary Open System-of-Systems). The objective of this research was to bring *time awareness, dynamicity,* and *evolution* into the design of SoS, to establish a sound *conceptual model* that provides a well-defined language for describing SoS, to investigate the intricate topic of *emergence* in an SoS, and to outline a *generic architectural framework* and an *SoS design methodology*, supported by some prototype tools, for the modeling, development, and evolution of time-sensitive SoS.

The AMADEOS partners from industry (Thales Nederlands, Resiltech, and ENCS) and universities (University of Firenze, Vienna University of Technology and Université Grenoble Alpes) joined efforts to arrive at results that are, on the one hand,

of wide-ranging value in an industrial context, and on the other hand, extend the understanding of SoS beyond the current state of the art in the academic world.

It is the objective of this book to present in a single consistent body the foundational concepts and their relationships in order to form a conceptual basis for the description and understanding of SoS and to go deeper in what we consider the characterizing and distinguishing elements of SoS: time, emergence, evolution, and dynamicity.

The first part of the book is devoted to this conceptual work. We start in Chap. 1 with the set of definitions of the relevant concepts. The need for a new approach and vision of interfaces is the topic of Chap. 2. In Chap. 3 we investigate the phenomenon of emergence in CPSoS, with a definition of emergence in the SoS context, and discuss some properties of emergent phenomena. Chap. 4 provides the definition of the AMADEOS conceptual model that captures SoS basic concepts and their interrelationships, and describes a SysML profile semi-formalization supporting the definition of SoS platform independent models (PIMs).

Part 2 of the book deals with the engineering framework developed in AMADEOS and the technical solutions adopted to deal with time, dynamicity, and evolution. More precisely, Chap. 5 defines the overall tool-supported AMADEOS Architectural Framework (AF), with its main building blocks and interfaces. Chap. 6 elaborates on the role of time and clocks in SoS and presents the design of a Resilient Master Clock (RMC), a hardware–software solution that has been developed in the course of the AMADEOS project. The final chapter in this part presents the AMADEOS dynamicity management and the different Monitor–Analyze–Plan–Execute (MAPE) components.

Part 3 of the book contains case studies of smart grid applications to demonstrate the suitability of the AMADEOS methodology for the design of an advanced industrial SoS.

In the following, we present a short overview of the main points covered in each one of the chapters of this book.

The objective of Chap. 1 is the development of a set of coherent concepts and the associated terms that can be used by domain experts to communicate their ideas about SoS. We start form the fundamental notion of a *Constituent System (CS)* that is time aware and consider an SoS as an integration of a finite number of CSs that are independent and operable, and which are networked together for a period of time to achieve a certain higher goal. The CSs interact by the timely exchange of information items (we call them Itoms) across *Relied Upon Interfaces* (RUI). An Itom is an atomic triple of data, explanation of the data and time. It follows a detailed model of time, based on Newtonian physics, and time measurement by digital clocks is presented. This model of time is used to define the notion of the state of a system at a given instant as the totality of the information items from the past that can have an influence on the future behavior of a system. We then discuss the characteristics of three basic communication mechanisms in cyberspace: datagrams, event-triggered positive acknowledgment, and retransmission protocols, as well as time-triggered protocols, followed by an elaboration of the information flow across stigmergic channels in the physical environment. After a short passage on interfaces (which are discussed at length in Chap. 2) the concepts of *dynamicity* and *evolution* are treated in the last section of Chap. 1.

The focus of Chap. 2 is on the important role that interfaces play in the control of the cognitive complexity of the models that explain the behavior of an SoS. The

boundaries among the CSs within an SoS are formed by the RUI of the CSs. The precise specifications of the syntactic, semantic, and temporal properties of these RUIs hide the internals of a CS implementation and are a good example for the application of the well-known divide and conquer principle. Interface layers allow for the discussion of system interface properties at different abstraction levels. Three interface layers are introduced: the *cyber-physical layer*, the *informational layer*, and the *service layer*. The cyber-physical layer is concerned with the reliable transport of context-sensitive bit-patterns across RUIs, both via messages in cyber-space and stigmergic channels in the physical environment. The informational layer abstracts from the context-sensitivity and the concrete technical implementations of the cyber-physical layer. The service layer structures the behavior of a system into a set of capabilities enabling management of dynamicity and evolution at the interface level. The specification of the execution semantics of an RUI assumes a frame-based synchronous data flow model. In many SoS the connections between the RUIs of the CSs are not static, but dynamic. The sections in Chap. 2 on dynamicity and managed evolution give details on the reaction and reconfiguration capabilities of an SoS that can be considered in the CPSoS design at the interface level.

Chap. 3 deals with the important topic of *emergence* in CPSoS. As quoted earlier, emergence is at the core of SoS engineering. The essence of the concept of emergence is aptly communicated by the following quote, attributed to Aristotle: "The whole is greater than the sum of its parts." The interactions of *parts* (the CSs) can generate a *whole* (the SoS) with unprecedented properties that go beyond the properties of any of its constituent parts. The immense varieties of inanimate and living entities that are found in our world are the result of emergent phenomena that have a small number of elementary particles at their base. After a lengthy discussion about the importance of multi-level hierarchies in the models of nearly decomposable complex systems, the following definition of emergence is presented: *A phenomenon of a whole at the macro-level is emergent if and only if it is of a new kind with respect to the non-relational phenomena of any of its proper parts at the micro level.* In the following sections of Chap. 3 the concepts of *downward causation* and *supervenience* are explained and it is conjectured that in a multi-level hierarchy emergent phenomena are likely to appear at the macro-level when there is a causal loop formed between the micro-level that forms the whole and the whole (i.e., the ensemble of parts) that constrains the behavior of the parts at the micro-level. A schema for the classification of emergent phenomena is presented and four concrete examples of emergent phenomena in computer systems are given. In the final section of Chap. 3, the focus is on the analysis of *detrimental emergent phenomena* in safety-critical CPSoS.

Chap. 4 covers the AMADEOS SysML profile for SoS conceptual modeling. The focus is on the definition of a SysML profile as a modeling support for representing the AMADEOS SoS conceptual model. The basic SoS concepts and their relationships are modeled using a SysML semi-formal representation according to different viewpoints, which represent the key perspectives of AMADEOS: structure, dynamicity and evolution, dependability and security, time, multi-criticality and emergence. Finally, a Smart Grid household scenario is introduced to exemplify the application of the profile and to instantiate the basic SoS concepts to a concrete case study from the Smart Grid domain, focusing on the architecture and emergence viewpoints.

Chap. 5 introduces the overall tool-supported *AMADEOS Architectural Framework (AF)* with its main building blocks and interfaces. The high-level representation of the AMADEOS AF is shown as a pyramid made of four different layers, namely, mission, conceptual, logical, and implementation. Apart from the mission block, all the remaining levels are organized in slices, each corresponding to a specific viewpoint. The following viewpoints of an SoS are explored: structure, dependability, security, emergence, and multi-criticality. Finally, for SoS modeling, a supporting facility tool based on Blockly is demonstrated. Blockly is a visual Domain-Specific Language (DSL) and has been adopted to ease the design of SoS by means of a simple and intuitive user interface; thus requiring minimal technology expertise and support for the SoS designer.

Chap. 6 stipulates that a global notion of time with known precision, shared by all CSs, is essential for the dependable operation of an SoS. Such a global notion of time is needed to specify the temporal properties of interfaces, to enable the interpretation of timestamps in the different CSs, to limit the validity of real-time data, to synchronize input and output actions across CSs, to provide conflict-free resource allocation, to perform prompt error detection, and to strengthen security protocols. Since CSs can join and leave the SoS dynamically, external clock synchronization is the preferred alternative in an SoS. Such an external clock synchronization can be based on the standardized time signal distributed worldwide by Global Navigation Satellite Systems (GNSS), such as GPS, Galileo or GLONASS. Since a GNSS time signal can become unavailable, a resilient master clock is proposed to extend the holdover interval after a time-signal failure. In AMADEOS a prototype of such a resilient GPS disciplined master clock has been developed and tested. The chapter describes the design, implementation and validation of this resilient master clock prototype.

Chap. 7 is devoted to the management of dynamicity in SoS. The well-known *Monitor–Analyze–Plan–Execute (MAPE)* control loop, developed by IBM in the context of autonomic computing, provides the framework for the management of dynamicity. When a Service Level Agreement (SLA) and its associated Service Level Objectives (SLOs) are associated with the service of a managed element, the MAPE control loop guarantees that these SLOs are met. If this is not the case, a new plan is calculated and used to reconfigure the system. This chapter presents AMADEOS dynamicity management and the components of MAPE.

Finally, Chap. 8 contains three case studies from the smart grid domain to demonstrate the viability of the AMADEOS approach to the design of SoS. The three case studies, electric vehicle charging, household management, and an integrated case study that combines the first two together with ancillary services, are modeled by using the AMADEOS Architectural Framework (AF) and the AMADEOS tool set. We utilize the four levels of the AMADEOS AF – mission, conceptual, logical, and implementation – as well as the seven viewpoints that have been defined: structure, dynamicity, evolution, dependability and security, time, multi-criticality, and emergence.

Modeling complex and pervasive infrastructures like the one used as case study clearly highlights how the support of a precise conceptual model and of specific tools for its instantiation is fundamental for a sound and comprehensive codification of the various properties of the whole. At design time the identification of causal loops in the lower levels of the hierarchy, enabled by the support for simulation through model execution, is a mandatory step to identify possible emergent behaviors at the higher

levels. In fact, such behaviors may lead, also in the future evolution of the SoS, to a violation of system requirements. A correct representation of the environment has also proven to be necessary. Finally, global time awareness and monitoring are fundamental for the early detection and for containing the effect of detrimental emergence phenomena at run time.

Although the chapters of the book are arranged in a logical order, an effort has been made to keep each chapter self-contained. The book contains also a glossary of all the terms and concepts used to ease reading and provide a reference for relevant terms in the domain of SoS.

This book can be used as a textbook or supplemental reading for advanced teaching on SoS, their concepts, and their design. In addition to this book, a set of slides is available that helps a lecturer in the development of the teaching material for an advanced course on Systems of Systems, (http://rcl.dsi.unifi.it/projects/amadeos/amadeosteachingmaterial).

Finally, we would like to thank all the following experts for reviewing and helping to improve this book: Wilfried Elmenreich (Alpen-Adria-Universität Klagenfurt, Austria), and Wilfried Steiner (TTTech, Austria), Lorenzo Falai (Resiltech), Leonardo Montecchi (University of Florence), and Antoine Boutet (LIRIS).

Hermann Kopetz
Andrea Bondavalli
Sara Bouchenak

Acknowledgments

This work was partially supported by the FP7-610535-AMADEOS project.