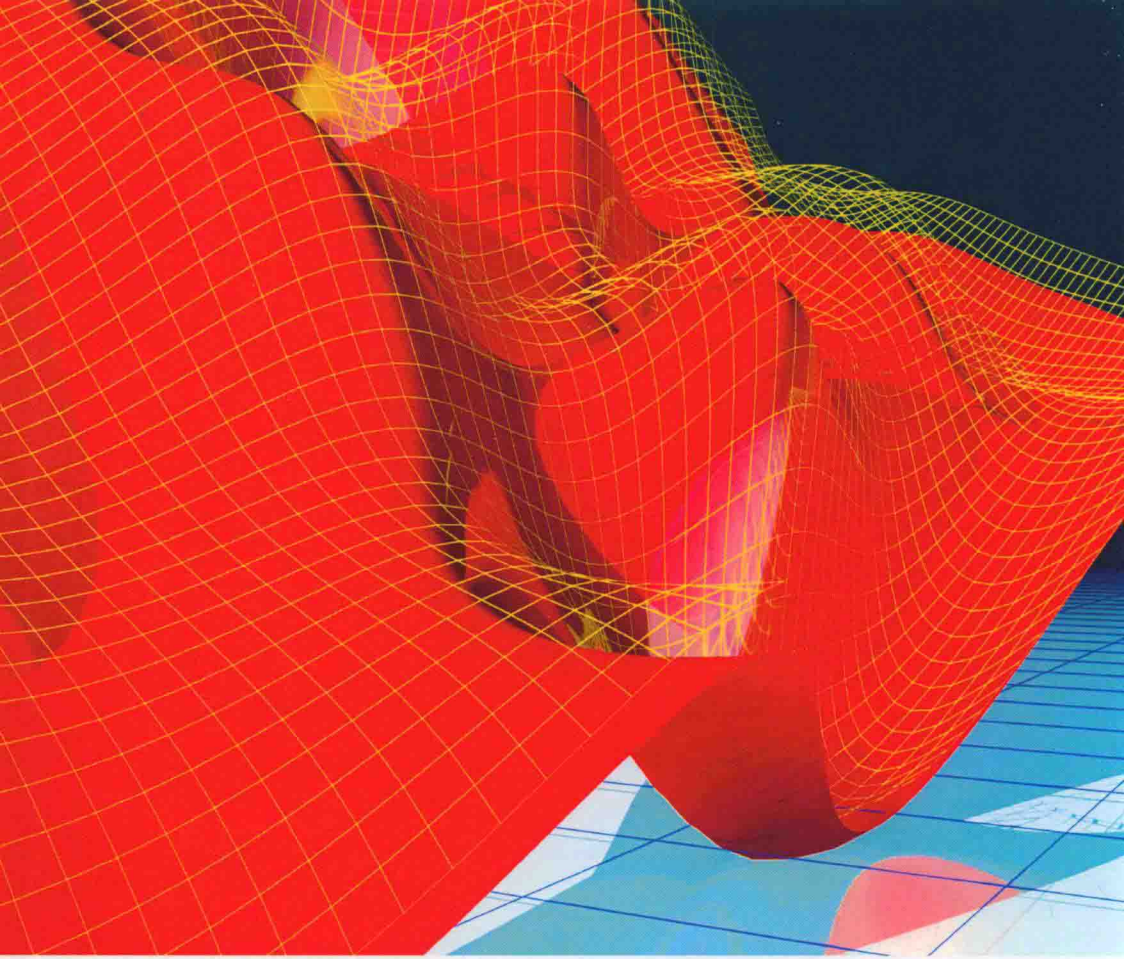

Computational Frameworks

Edited by
Mamadou Kaba Traoré

Systems, Models and Applications

iSTE PRESS

ELSEVIER

Computational Science and Engineering (CSE) is core to all engineering sciences. As a result, computational systems are seeing worldwide growth of services to all user domains: public or private, military or government, education or industry, healthcare or finance, science or entertainment.

The use of computational technologies (e.g. big data, high performance clusters, databases and information systems, integrated and embedded hardware/software components, smart devices, mobile and pervasive networks, cyber physical systems, etc.) requires many important theoretical, methodological, technical and scaling issues to be studied, monitored, and predicted, in order to save time, reduce costs and add value to social capital.

With a highly selective set of contributions, this book aims to provide an overview of advanced perspectives in this domain, while bridging the gap between frontline research and practical efforts. It is unique in showing the interdisciplinary nature of this area and the way in which it interacts with emerging technologies and techniques.

Mamadou Kaba Traoré currently chairs the distributed software engineering Master degree at Blaise Pascal University in France. His research is on formal specification, symbolic manipulation and code synthesis of simulation models.

ELSEVIER
iSTE PRESS
www.iste.co.uk
9 781785 482564

Edited by
Mamadou Kaba Traoré

Computational Frameworks

Computational Frameworks

Systems, Models and Applications

Edited by

Mamadou Kaba Traoré

ISTE PRESS

ELSEVIER

First published 2017 in Great Britain and the United States by ISTE Press Ltd and Elsevier Ltd

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms and licenses issued by the CLA. Enquiries concerning reproduction outside these terms should be sent to the publishers at the undermentioned address:

ISTE Press Ltd
27-37 St George's Road
London SW19 4EU
UK

www.iste.co.uk

Elsevier Ltd
The Boulevard, Langford Lane
Kidlington, Oxford, OX5 1GB
UK

www.elsevier.com

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

For information on all our publications visit our website at http://store.elsevier.com/

© ISTE Press Ltd 2017
The rights of Mamadou Kaba Traoré to be identified as the author of this work have been asserted by him in accordance with the Copyright, Designs and Patents Act 1988.

British Library Cataloguing-in-Publication Data
A CIP record for this book is available from the British Library
Library of Congress Cataloging in Publication Data
A catalog record for this book is available from the Library of Congress
ISBN 978-1-78548-256-4

此为试读,需要完整PDF请访问: www.ertongbook.com

Computational Frameworks

Introduction

Introduction to Computational Frameworks: From Heterogeneity Challenge to Integrative Levels of Organization

The idea for this volume started at AUSTECH 2015, the first international multiconference in Technology of the African University of Science and Technology (AUST). We had organized it in October 12–14 2015, on the AUST campus in Abuja, Nigeria. AUSTECH 2015 focused on current developments in engineering technologies, and their scientific and industrial applications for development in sub-Saharan Africa. It hosted three symposia, each of them reflecting one of the core technological areas developed at AUST. Among them was the Computer Science Symposium (CSS), a platform to explore and facilitate the dissemination of the most recent advancements in the theories, technologies and applications of computational systems for development (with special emphasis on sub-Saharan Africa).

Computational Science and Engineering (CSE) is core to all engineering sciences. As a result, computational systems are seeing an explosion growth of services to all user domains worldwide: public or private, military or governmental, educational or industry, healthcare or finance and science or entertainment. Beyond their diversity, there is a unity of concerns, which can

---

Introduction written by Mamadou Kaba TRAORÉ.

be expressed through some key research questions. The use of computational technologies (e.g. high performance clusters, Big Data, databases and information systems, integrated and embedded hardware/software components, smart devices, mobile and pervasive networks, cyber physical systems, etc.) requires theoretical, methodological, technical, scaling and many other important issues to be studied, monitored and predicted in order to save costs and time and to add value to social capital.

CSS in AUSTECH 2015 brought together world-class researchers and students to share and discuss current challenges faced in the pursuit of advancing focused knowledge. It quickly appeared from the contributions that trends and key challenges can be seen from the perspective of computational frameworks. The quality of the presentations and the richness of exchanges led us to select four major contributions by well-recognized researchers to be chapters in a pioneering volume on CSE. It is unique in that it presents the views on computational systems, models and applications under one holistic scope. With such a very restrictive and highly selective set of contributions, this handbook aims at providing an overview in very advanced theoretical, methodological, technical and cognitive perspectives in the domain, while bridging the gap between frontline research and practical efforts.

I.1. Computational science and the challenge of heterogeneity

CSE is a relatively new discipline, commonly heralded as the third mode of discovery (next to theory and experimentation) [ODE 14, RAY 11]. Computational science is often described as fusing numerical methods (with a special focus on Modeling & Simulation (M&S)), computer systems (software and hardware, with a special focus on High Performance Computing) and information science (with a special focus on data management). Today, it is an essential component of modern research in different areas (computational physics, computational biology, computational chemistry, etc.) where it often bridges computer science, mathematics and domain-specific knowledge. It calls for the use of some computational approach that may also require the design of the supporting computational system or the use of an existing one. This is where computational frameworks come into play.

In all forms of CSE, there is a common barrier to the design and use of computational systems, which we call the heterogeneity challenge. Indeed, computational approaches are confronted to a universal complexity due to the need to integrate heterogeneous components into a homogeneous whole (i.e. the framework). Such heterogeneity can take at least one of the following forms:

– disparity of abstractions (often referred to as multiabstraction or multilevel) [BEN 98];

– diversity of details (often referred to as multiresolution) [DAV 93];

– variety of scales (often referred to as multiscale) [TAO 09];

– multiplicity of perspectives (often referred to as multiview or multiperspective) [REI 14];

– distribution of geographic locations (often referred to as multisite) [YAN 08].

As a way to draw attention how difficult this challenge is, the literature often resorts to the use of the prefix "multi". Terms like "multiphysics", "multiobjective" or "multiparadigm" are generalizations (and sometimes accumulations) of these forms of heterogeneity and their underlying complexity. From the literature, we can also derive the following four key requirements to overcome this barrier:

– a system-theoretic approach is needed for a deep and structured understanding of the phenomena involved. Bernard Zeigler has established a foundational system-theoretical framework suitable for systems analysis and design [ZEI 76, ZEI 84, ZEI 00]. In Chapter 1 of this volume, he explores, beyond the frontiers of individual system design, how to efficiently integrate them, leading to the design of System of Systems (SoSs). While a classic system design results in the building of a component that has its operational and managerial independence, the design of a cluster of such components (that, in addition, can be geographically distributed) addresses the evolutionary development issue, i.e. the emergence of functionalities that none of the component can (or was designed to) solve alone. Such emergence can be positive (i.e. desired) or negative (i.e. not wished). The concept of pathways is introduced to allow distributed individual-based

此为试读,需要完整PDF请访问: www.ertongbook.com

tracking and to coordinate individual systems toward SoSs capable of completing specific desired goals/subgoals;

– a holistic approach is necessary for a sound cognitive management of the system or the SoS identified. That is what Andreas Tolk highlights in Chapter 2. In the same way that, technical integration is needed for SoSs, there is a parallel effort to do for the cognitive integration of multiple disciplines, with three levels of alignment: multidisciplinarity, interdisicplinarity and transdisciplinarity. The chapter shows how proper data concepts have to be semantically aligned and processes be synchronized. This is formalized in a Conceptual Interoperability Model;

– validity is always a critical (and often non-trivial) issue. David Hill and colleagues show in Chapter 4 that a common pitfall to all computational approaches is numerical reproducibility, i.e. the obtention from one environment to the other of the same execution results for a given computational experiment. This chapter discusses sources of non-reproducibility, specifically in the sensitive context of high-performance simulation, and makes recommendations to achieve a sound scientific validity;

– it is increasingly compulsory to use performant infrastructures to support computations as they are getting heavier with new areas and demands of CSE. Abderazak Ben Abdallah points out in Chapter 3 that conventional single-core-based designs are no longer suitable to satisfy high-performance and low-power consumption demands in new embedded applications. Designers, thus, turn toward the very hard task of integrating multiple cores on a single chip. The chapter explores alternatives to achieve such goal and examines their pros and cons.

Computational frameworks are a systematic and assessable way to answer the heterogeneity challenge. Here, we suggest four integrative levels of organization where computational frameworks can address each of the above-mentioned requirements.

I.2. Computational frameworks and the integrative levels of organization

Surprisingly, despite its wide use in the literature, the notion of "computational framework" has no specific and commonly agreed definition, while numerous definitions can be found for terms such as "framework"

(https://en.wikipedia.org/wiki/Framework), "software framework" [GAC 03], "architecture framework" [HIL 04] and many others. Here, we try to provide working definitions, as we claim the term of "computational framework" is used to express various levels of understanding of one same general concept, and that those levels can be structured in a four-layer hierarchy.

As presented in Figure I.1, a computational framework can relate to:

– computational (hardware) systems, i.e. platforms designed to serve as computing systems. This ranges from original computer architectures to computer networks and high-performance architectures. This is the perspective of Chapter 3;

– computational (software) programs, where the basic concerns are modeling and programming. Chapter 4 can be seen from that perspective, with a focus on software's impact on reproducibility;

– computational theories elaborated for problem solving. This ranges from numerical approaches (numerical analysis, operations research, etc.) and logic to artificial intelligence and M&S. Chapter 1 can be placed in this perspective;

– computational applications (i.e. domain-specific methodologies, computations and general results). Chapter 2 can be seen from that perspective, with a generic discourse on multidisciplinary, interdisciplinary and transdisciplinary concepts and illustrations in the specific context of medical simulation;

– a combination of any 2, 3 or 4 or the formers. Sets in Figure I.1 are arranged such that the following are possible: (1)+(2), (1)+(3), (1)+(4), (1)+(2)+(3), (1)+(2)+(4), (1)+(3)+(4) and (1)+(2)+(3)+(4). This suggests that, at one hand, although frameworks can be defined and limited to each set, the ultimate aim of a computational system is to run on top of a given execution platform. On the other hand, when a computational system is associated with any component of any other set, the result is a computational system (e.g. a computational program running on top of a computational hardware system forms with the latter a computational system). Therefore, all associations involving level (1) are meaningful in that the core of the resulting system may be just a hardware system, a software system on top of

a hardware system, a theory implemented in a software and running on a hardware, etc.

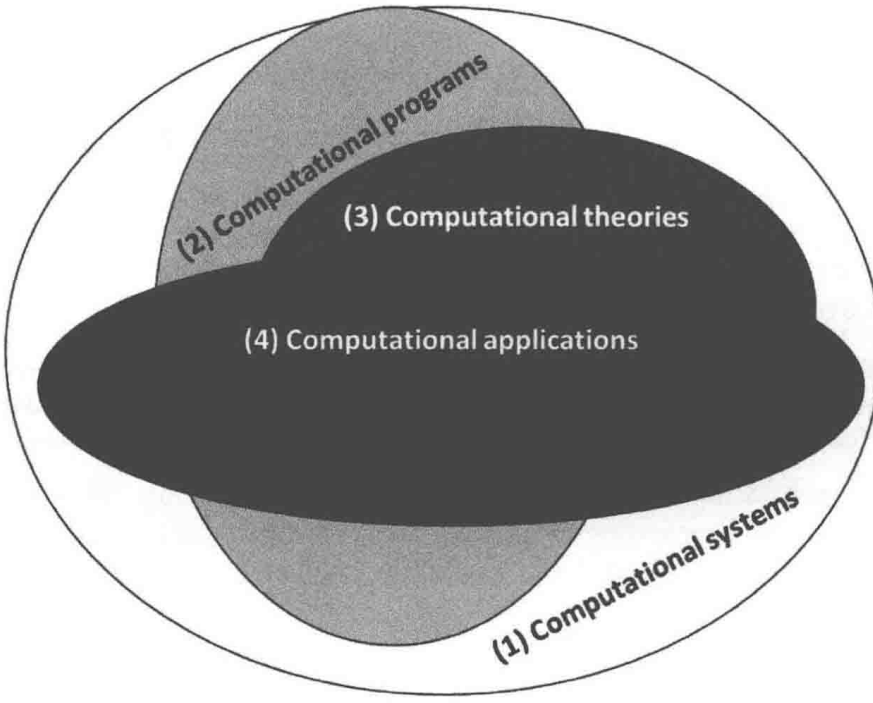

Figure I.1. *Computational systems components*

We propose the following working definitions:

DEFINITION 1.– A computational framework is a set of concepts, properties, rules and tools that can be of a theoretical, methodological, technical or technological nature, which are glued in a normative architecture, to ease and possibly automate the problem-solving process partly or entirely.

DEFINITION 2.– CSE deals with the development and application of computational frameworks to solve complex problems.

What Figure I.1 suggests in an intuitive way is made explicit in Figure I.2, where we show the integrative nature of the organization in which computational frameworks can be formalized.

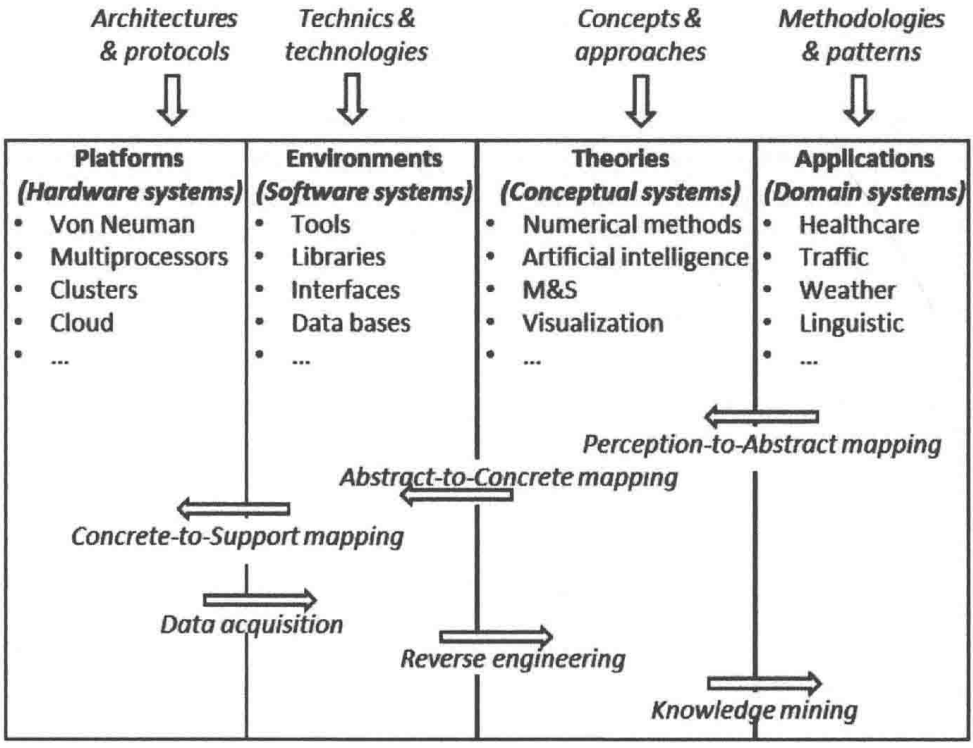

Figure I.2. *Integrative levels of organization for computational frameworks*

Computational frameworks can be defined at a:

– applications level, as in [CHA 13] for healthcare, [ALV 10] for civil engineering, [FLA 10] for metabolic networks, [ONE 11] for dramatic art and many others;

– theories level, as in [DOR 02] for geometry, [GAS 09] for nonlinear algebra, [ZEI 76] for discrete event simulation and many others;

– (software) environments level, as in [OMG 01] for distributed computing, [GAM 95] for object-oriented design, [STE 09] for Java code synthesis and many others;

– (hardware) platforms level, as in [ANG 06] for the design of multiprocessors on-chip, [PAT 01] for dynamic optimization of hardware systems, [FLE 13] for code acceleration on homogeneous multicores and many others.

Very importantly, computational frameworks can also be defined to address the level-to-level mapping issue, i.e.:

– perception-to-abstraction mapping, which allows the development of an application framework on top of a theoretical one, linking high-level concepts of the application domain to lower level concepts of the theoretical approach (examples are [BER 00] that bridges computer graphics and partial differential equations, [SOC 98] that bridges vision and geometric algebra and [LEE 99] that bridges network security and data mining);

– abstract-to-concrete mapping, which allows the translation of a theoretical framework's elements into their implementation counterparts (examples are [PUE 99] in interface development, [ATK 03] in model-driven system design and numerous computer-aided software engineering tools [FUG 93]);

– concrete-to-support mapping, which allows the deployment of software components onto hardware hosts (examples are the MPI framework [MPI 94] for distributed programming with general languages through message interfacing, and [FER 98] for parallel programming with Java language).

Similarly, some computational frameworks are defined to reverse this level-to-level stream, therefore using the lower level information to build the higher level. That is the case for:

– data acquisition frameworks, which go from platforms (generally an ecosystem of devices) to databases [VIL 13, CED 12, MER 13];

– reverse engineering frameworks, which go from codes to conceptual representations [BRU 14, ROY 08];

– knowledge mining frameworks, which enrich application knowledge with properties discovered/established at the theoretical level [PAL 12, VAN 05, FAY 96].

More importantly, advanced frameworks aggregate several levels and level-to-level mappings. For example, the Common Component Architecture is a standard software architecture allowing plug-and-play composition of scientific applications [RAY 11]. The High-Level Architecture (HLA) framework combines M&S theoretical framework with distributed computing environment framework [FUJ 98]. Lee *et al.* [LEE 03] extend the HLA framework to Discrete Event System Specification theoretical

framework, on the one hand, and to the transportation application framework, on the other hand. Smarsly *et al.* [SMA 13] combine data acquisition (reverse process) and abstract-to-concrete mapping (numerical methods). Cui and Franchetti [CUI 12] cover the four integrative levels, bridging distribution systems application with high-performance computing through Monte Carlo simulation and a MATLAB-based solver.

I.3. Reading the volume

Few books are specifically devoted to computational frameworks. This volume will probably be one of the firsts to do so in a way that shows the interdisciplinary essence of this area and how emerging technologies and techniques can take part in it. The perspectives presented in this chapter provide at least three ways of reading the book:

1) a trail that follows the cognitive-to-technical process of understanding going from very conceptual levels to very technological ones (therefore reading Chapter 2, then Chapter 1, then Chapter 4 and then Chapter 3);

2) a trail that follows key requirements for a sound approach to CSE (therefore reading Chapter 1, then Chapter 2, then Chapter 4 and then Chapter 3);

3) a trail that follows the structural aspect of computational frameworks going from core components to periphery (therefore reading Chapter 3, then Chapter 4, then Chapter 1 and then Chapter 2).

It is very likely that students, domain researchers and professionals will find this volume increasingly interesting if they undergo several readings, adopting different trails from one reading to the other, thus grabbing more and more insights. Indeed, each of the book's chapters opens many research perspectives that call for further developments, leading to new research directions.

I.4. Bibliography

[ALV 10] ALVANCHI A., LEE S., ABOURIZK S., "Modelling framework and architecture of hybrid simulation and discrete event simulation for construction", *Computer Aided Civil and Infrastructure Engineering*, vol. 26 no. 2, pp. 1–15, 2010.

此为试读,需要完整PDF请访问: www.ertongbook.com