Contributions to system-level modelling and simulation of hardware-software architectures of embedded systems.

Sébastien Le Nours

Nantes Université

HABILITATION A DIRIGER DES RECHERCHES

DE NANTES UNIVERSITE

Par

Sébastien LE NOURS

Contributions to system-level modelling and simulation of hardware-software architectures of embedded systems

Présentée et soutenue à NANTES, le 21 novembre 2022
Unité de recherche : IETR UMR CNRS 6164

Composition du Jury :

President of the jury	Abdoulaye Gamatié, Senior Researcher CNRS LIRMM
Reviewer	Florence Maraninchi, Full Professor INP Grenoble
Reviewer	Andy Pimentel, Full Professor University of Amsterdam
Reviewer	Olivier Sentieys, Full Professor University of Rennes 1
Examiner	Liliana Cucu-Grosjean, Senior Researcher INRIA
Examiner	Sébastien Pillement, Full Professor Nantes University

Document prepared for obtaining an accreditation to supervise research
(*Habilitation à diriger des recherches*)

Contributions to system-level modelling and simulation of hardware-software architectures of embedded systems

Sébastien Le Nours

Defended on November 21th, 2022 with jury composed of :

Abdoulaye Gamatié	Senior Researcher	CNRS LIRMM	President of the jury
Florence Maraninchi	Full Professor	INP Grenoble	Reviewer
Andy Pimentel	Full Professor	University of Amsterdam	Reviewer
Olivier Sentieys	Full Professor	University of Rennes 1	Reviewer
Liliana Cucu-Grosjean	Senior Researcher	INRIA	Examiner
Sébastien Pillement	Full Professor	Nantes University	Examiner

Contents

List of figures **7**

List of tables **9**

Summary of research and teaching activities **13**
A Personal record . . . 13
B Summary of teaching activities . . . 15
B.1 Teachings . . . 15
B.2 Responsibilities and services to the community . . . 16
C Summary of research activities . . . 18
C.1 Addressed area of research . . . 18
C.2 Supervision activities . . . 18
C.3 Contracts and research projects . . . 20
C.4 Publications . . . 21
C.5 Member of boards and committees . . . 21
C.6 Animation of research . . . 21

1 Introduction **23**
1.1 Context . . . 23
1.1.1 Evolutions and trends in the design of hardware-software architectures of embedded systems . . . 23
1.1.2 Principles of electronic system-level design . . . 24
1.1.3 Issues and addressed problems . . . 26
1.2 Contributions to the research field . . . 28
1.2.1 Development of activities . . . 28
1.2.2 Contributions to improving system-level simulation efficiency . . . 28
1.2.3 Contributions to the creation of system-level models of hardware-software architectures 29
1.2.4 Contributions to online management of hardware-software architectures . . . 29
1.3 Organization of the document . . . 29
1.4 Publications related to Chapter 1 . . . 30

2 Contributions to system-level modelling and simulation **31**
2.1 Introduction . . . 31
2.2 Addressed issue . . . 31
2.3 Principles of the proposed system-level approach . . . 33
2.4 Creation of state-based models . . . 35
2.5 Generic execution model . . . 37
2.6 Implementation and preliminary results . . . 39
2.7 Conclusion . . . 40
2.8 Supervision . . . 41
2.9 Publications related to chapter 2 . . . 41

3 Communication modelling of multiprocessor architectures **43**
3.1 Introduction . . . 43
3.2 Improvement of simulation efficiency for distributed architecture analysis . . . 44
3.2.1 Characteristics of the studied systems . . . 44
3.2.2 Application of the simulation method . . . 46
3.2.3 Obtained results . . . 47
3.3 Improvement of simulation efficiency for multiprocessor systems-on-chip analysis . . . 47
3.3.1 Characteristics of the studied systems . . . 47
3.3.2 Application of the simulation method . . . 49
3.3.3 Obtained results . . . 51
3.4 Conclusion . . . 52
3.5 Supervision . . . 53
3.6 Publications related to Chapter 3 . . . 53

4 Modelling and management of computation resources in multiprocessor systems **55**
4.1 Introduction . . . 55
4.2 Modelling and sizing of mobile radiocommunication system architectures . . . 55
4.2.1 Modelling and simulation method for non-functional properties . . . 56
4.2.2 Modelling of dynamic functions . . . 59
4.3 Management strategies of multiprocessor architectures . . . 61
4.3.1 Related work and contribution of our work . . . 62
4.3.2 Proposed online management strategies . . . 63
4.3.3 Online management architecture modelling and simulation . . . 67
4.4 Conclusion . . . 68
4.5 Suepervision . . . 69
4.6 Publications related to Chapter 4 . . . 70

5 Probabilistic modelling and simulation of multiprocessor systems **71**
5.1 Introduction . . . 71
5.2 Establishment of a measurement-based characterization flow for the creation of probabilistic models . . . 72
5.2.1 Proposal . . . 72
5.2.2 Results . . . 74
5.3 Adoption of statistical model-checking analysis methods . . . 76
5.3.1 Proposal . . . 76
5.3.2 Results . . . 78
5.4 Conclusion . . . 80
5.5 Supervision . . . 80
5.6 Publications related to Chapter 5 . . . 81

6 Conclusion and prospects **83**
6.1 Review of the work carried out . . . 83
6.2 Prospects and directions envisaged . . . 84

List of publications **87**

Bibliography **91**

List of Figures

1 Lessons taught in the field of embedded system design. The titles in bold are modules which I assume the responsibility, the titles in italics are modules which I episodically participated. . 16
2 Position of research activities in the field of hardware/software architectures design of embedded systems. 18

1.1 The Y diagram with detailed at the system level, inspired from [16]. The main activities led at the system level are identified. 25
1.2 Illustration of the compromise between accuracy and analysis time for different performance evaluation methods, (a) simulation-based approaches, (b) formal approaches. 26
1.3 Different levels of complexity in shared resources of multiprocessor systems. 27
1.4 Position of the covered research topics, main projects and co-supervised PhD thesis. 28

2.1 (a) Example of a system-level model with three architecture viewpoints: functional, executive and physical, physique, (b) description of functions behaviours with elementary communication and computation statements. The execution semantic related to each statement needs to be defined such as illustrated in Figure 2.7. 32
2.2 Discrete event simulation of the architecture model in the figure 2.1. In this example, the instants associated with the communication of M_2 via the interfaces IF1 and IF2 and the communication bus N are detailed. The instants associated with the communications of the other relations are not detailed for the sake of readability. The shaded part corresponds to a situation of contention on the interface IF1. 33
2.3 (a) Execution of a model with classical simulation approach, (b) execution of a model with instantaneous computation of future simulation instants. 34
2.4 (a) Situation with erroneous prediction of instants when a shared resource is used, (b) principle of the correction method of resource usage instants. 34
2.5 (a) Notation of timed Petri nets used to describe the dependencies between simulation instants, (b) expression of the transition instants in the illustrated network. 36
2.6 Examples of elementary patterns used to describe the dependencies between start and end instants of computation and communication statements. 37
2.7 Examples of patterns related to communication statements: (a) rendez-vous protocole, (b) infinite capacity FIFO protocol. T_{Wr} and T_{Rd} denote the write and read durations. 37
2.8 Examples of built Petri net models to describe the dependencies between start and end instants of communication and computation statements. 38
2.9 (a) Structural and behavioral description of the proposed generic execution model, (b) management procedure of the computed instants. 38
2.10 Influence of the complexity of the calculation method on the achieved simulation speed-up. . 39

3.1 Illustration of levels of abstraction in communication description for the two case studies. . . 43
3.2 Illustration of the distributed architecture studied in the scope of the CIFAER project [J3-1]. 45
3.3 Illustration of exchanged between two HomePlugAV interfaces and evolution of the interface memory usage during the transmission [J3-1]. 45
3.4 Evolution over the simulation time of transaction level and message level models [J3-1]. . . 46

3.5 Illustration of the studied multiprocessor systems. They are made of an application described following the SDF model of computation and a platform with multiple tiles connected through a communication bus and a shared memory. 48
3.6 (a) Coding example of the procedure *WriteTokens* for writing n tokens, (b) state-transition diagram describing the sequence of writing n tokens into the shared memory. The shaded rectangles highlight the states during which the shared resources (communication bus, memories) are accessed. 48
3.7 Execution of models describing the writing of n tokens on channel C_1 and the reading of n tokens from channel C_0 in the example of Figure 3.5. (a) At the transfer level, the basic delays and penalties caused by contentions are highlighted. (b) At the message level, the communication times (denoted T_W and T_R) are calculated during the simulation. 50
3.8 Timed Petri net for writing n tokens on channel C_1 and reading n tokens from channel C_0. Communication is considered at transaction level with the FCFS arbitration policy. The shaded rectangles correspond to the situation during which the shared resources (bus, memory) are accessed. 50
3.9 Illustration of the organization of the message level communication model described in the SystemC language. This model uses the calculation method presented in Chapter 2. 51
3.10 (a) Modeling of the studied applications: Sobel filter and JPEG decoder. (b) Experiment platform formed by seven tiles, a shared memory and a communication bus. 52

4.1 Modelling and evolution of the computation load according to different approaches. Δt_1 and Δt_2 denote the interval of time between two successive data to be processed. Situation (b) corresponds to a pipeline execution of the architecture. 56
4.2 (a) Organisation of an LTE data frame in the time and frequency domains, (b-1) organization of the functions of the physical layer in reception, (b-2) possible allocation of functions on two types of computing resources. 57
4.3 Observations of the occupation of computation and memory resources for an architecture based on a dedicated hardware resource [50], [J4-1]. 58
4.4 Description of the behaviour of the communication interfaces according to the dynamic function principle [J4-1]. 60
4.5 Evolution of the usage of resources according to an operating scenario that modifies the applications executed as well as the network available [J4-2]. 60
4.6 Observation of the computation loads induced by the physical layer functions of the E-UTRA and WiFi protocols [J4-2]. 61
4.7 (a) Example of an application described according to the SDF model of computation, (b) Computation resources usage for two distinct mappings associated with the application app_1. 62
4.8 (a) Position of the online manager in relation to the applications running within the same cluster, (b) Process for selecting the frequency of the cluster operation and defining task allocations. 63
4.9 Different methods of allocating timeslots based on prepared execution traces: (a) prepared execution traces for two active applications, (b) combining traces according to the FCFS method, (c) combining traces according to the LASP method, (d) combining traces according to the proposed GAPVC method. 64
4.10 Hierarchical organization for managing multiple applications running on a platform consisting of multiple clusters. 66
4.11 Comparison of average power relative to the platform configuration 8×8 [J4-3]. For this comparison, no application re-allocation is necessary. 67
4.12 Simulation principle using the traces established during each new operating situation. 68
4.13 A high-level model of an application and associated online manager within the Intel CoFluent Studio environment. 68
4.14 Simulation of dynamic power consumption for different operating frequencies and different possible allocations of the considered application [C4-4]. The simulation is conducted within the Intel CoFluent Studio environment using the proposed principle. 69

5.1 Measurement-based characterisation flow for the establishment of probabilistic models for the temporal properties analysis of MPSoC architectures. 72
5.2 Process of characterization and probabilistic modelling of computation and communication resources. 73
5.3 Distribution of the TL simulation models. 75
5.4 Extended modelling flow to allow application of SMC methods. 77
5.5 Details about the generation and simulation flow of instrumented and monitored SystemC models. 77
5.6 Distribution of measured data and estimates obtained by simulation for the experiment *Sobel4a* [47]. 79
5.7 Average values expressed in cyces, obtained by measurement on a real target, simulation of 1 000 000 iterations of a probabilistic model and simulation controlled by the SMC approach. The values given in parentheses indicate the deviation from the mean value measured for the first configuration of the platform [47]. 79
5.8 Distribution of measured data and estimates obtained by simulation for the experiment *Sobel4b* [47]. 80
5.9 Average values expressed in cyces, obtained by measurement on a real target, simulation of 1 000 000 iterations of a probabilistic model and simulation controlled by the SMC approach. The values given in parentheses indicate the deviation from the mean value measured for the second configuration of the platform [47]. 80

List of Tables

1 Number of hours and distribution of teachings at the University of Nantes for the 2019-2020 academic year. 15

2.1 Tested configurations and associated simulation results [J2-1]. 40
2.2 Summary of the case studies using the proposed modelling and simulation approach. These case studies are presented in the indicated chapters. The indicated speed-up factors are obtained without degrading the accuracy of the predictions. 40

3.1 Observed simulation duration (in seconds) for execution of transaction and message level models. The models were executed on a Intel Core2 Dual (2.66 GHz) running under Windows7 [C3-2]. 47
3.2 Comparison of average execution durations obtained by measurements on a real taget and by simulation of transfer and message level models. The values are expressed in clock cycles. . 52
3.3 Execution durations on real target and simulation durations (in HH:MM:SS) for 1 000 000 of application iterations according to different possible allocations. 53

4.1 Comparison of multi-core platform online management strategies for 511 different operating situations. 65

Prologue

This document presents my research activities since my recruitment at the University of Nantes in September 2004. These activities were successively carried out within the IREENA laboratory (Institute for Research in Electronics and Electrotechnics of Nantes Atlantique) until 2012, then within the IETR laboratory (Institute of Electronics and Digital Technologies). The scope of these activities corresponds to the design of hardware and software architectures of embedded electronic systems. In this context, my research activities have focused on the definition of methods favouring the modelling and simulation of hardware and software architectures in order to optimize them under time, cost and energy constraints.

In this document, during the presentation of each of the covered topics, I report on the framework of this work, the nature of the contributions made and the approach undertaken. In doing so, I also explain my role in the supervision of the students who participated and contributed to this work and more broadly the collaborations set up. This synthesis tries to reconcile clarity and precision of the explanations.

This document has three main parts. The first part corresponds to a detailed CV, summarizing in particular my professional experience with regard to the various activities carried out in research and teaching. The second part details in five chapters the research activities carried out since obtaining the doctoral degree, retracing the most significant results. Finally, the third part presents an assessment of the activities carried out and establishes a potential research program taking into account the identified prospects. Attached to this document is a detailed list of my publications.

Summary of research and teaching activities

A Personal record

Sébastien Le Nours
Associate Professor at Nantes Université
Polytech Nantes — Laboratory : IETR, UMR CNRS 6164
Email : sebastien.le-nours@univ-nantes.fr — Rue Christian Pauc, 44306 Nantes
http://www.ietr.fr/sebastien-le-nours — Phone : +33 2 40 68 30 53

PROFESSIONAL EXPERIENCES

Associate Professor — University of Nantes, FR
Sept. 2004 - Present

- Teaching at Polytech Nantes, the engineering school of the University of Nantes. Responsibilities within the Electronics and Digital Technologies (ETN) department of Polytech Nantes.
- Research at the Institute of Electronics and Digital Technologies (IETR), UMR CNRS 6164. Specialization in the modelling and design of hardware and software architectures of embedded systems.

Invited researcher — University of Queensland, AUS
Oct. 2012 - Oct. 2013

Assistant Professor — University of South Brittany, FR
Sept. 2003 - Aug. 2004

PhD researcher — INSA Rennes, FR
Oct. 2000 - Oct. 2003

FORMATIONS UNIVERSITAIRES

Doctoral thesis in electronics — Rennes, FR
Institut National des Sciences Appliquées (INSA) — *Oct. 2000 – Oct. 2003*

- **PhD Thesis topic:** *Study, optimization and implementation of MC-CDMA systems on heterogeneous architectures*. Jury : E. Martin, D. Roviras (reviewers), M. Auguin, J.P. Calvez, M. Jézéquel (examiners), J. Citerne (supervisor), F. Nouvel, J.F. Hélard (co-supervisors).

Master degree in electronic — Lille, France
Université de Lille I — *Sept. 1999 – Sept. 2000*

- **Master Thesis topic:** *Design of clock systems based on CMOS micro-resonator for embedded applications*. B. Stefanelli (supervisor).

Engineer degree in electronic — Brest, France
Institut Supérieur de l'Electronique et du Numérique (ISEN) — *Sept. 1997 – Sept. 2000*

TEACHING ACTIVITIES

Main responsibilities of teaching modules

SoC Design (ETN5)	*ETN Department*
Hardware-software codesign (ETN5)	
Digital circuit design (ETN4)	
Microprocessor systems (ETN4)	
Hardware-Software Architectures Codesign (M2)	*International Master program*
Performance Evaluation of Embedded Systems (M2)	

Main responsibilities and services to the community	ETN Department
Member of the department council	*2004-Present*
Head of end-of-studies internships (ETN5)	*2016-Present*
Responsible for the first year of the engineering cycle (ETN3)	*2006-2012*
Responsible for international internships	*2005-2006*

RESEARCH ACTIVITIES

Research areas

System level design of embedded systems. Performance evaluation of hardware-software architectures. Prototyping of embedded systems.

Research projects and contracts

Coordinator for the University of Nantes of the international RFI WISE pSSim4AI project with the OFFIS institute, Germany	*2020-2023*
Coordinator for the University of Nantes of the Hubert Curien Partnership (PHC) PETA-MC with the OFFIS institute, Germany	*2019-2020*
Coordinator for the University of Nantes of the international project RFI WISE PETA-MC	*2019-2020*
Coordinator of an industrial contract with Intel company	*2016-2017*
Visiting researcher at the University of Queensland, Australia	*2012-2013*
Coordinator for the University of Nantes of the ANR CIFAER project	*2007-2011*
Coordinator for the University of Nantes of a GDR ISIS Young Researcher project	*2007*

Publications

9 international journals, 1 national journal

28 international conferences

5 publications in national national conferences

6 research reports

Student supervision

4 PhD thesis defended, 1 PhD thesis in progress

7 Master thesis defended

Participation in 2 thesis juries as examiner and 3 juries as guest member

Main responsibilities and services to the community

Elected member of the Polytech Nantes research council	*2012-2015*
Member elected to the board of the IREENA laboratory	*2008-2011*
Recruitment committees of the 61st section University of Nantes/Ecole Centrale de Nantes specialist commission	*2005-2007*

Member of recruitment committees for the University of Nantes, INSA Rennes, University of South Brittany

Organization of conferences DASIP 2007, SDR-WinnComm 2009, 2010.

Participation in the activities of the GDR (national research consortium) ISIS (image and signal processing) and SoC2 (system-on-chip and embedded systems)

B Summary of teaching activities

B.1 Teachings

Since my recruitment in 2004 at the University of Nantes, my teaching activities have mainly been carried out at Polytech Nantes, the engineering school of the University of Nantes. These activities are divided into two distinct formations: within the Electronics and Digital Technologies (ETN) department preparing for the engineering degree and within the framework of the International Master program in Wireless Embedded Technologies (WET) (formerly Electronic Systems and Electrical Engineering, SEGE) from the University of Nantes. I participate to these two formations through different modules related to the field of embedded computer systems. By way of example, Table 1 indicates the distribution of courses I taught during the 2019-2020 academic year.

Semester	Level	Title	Lecture	Tutorial	Practical	Project
s.7	ETN4	**Circuit design**[1]	3,75	17,5	42	
		Microprocessor systems[1]	2,5	17,5	21	
		3rd year internship supervision				1
s.8	ETN4	Real-time system design		13,5		
		Transversal project				15
s.9	ETN5	**SoC design**[1]	4,5	15	9	
		HW-SW codesign[1]	9		18	
		Embedded system architectures			9	
		Technical projects				25
		4th year internship supervision				11
s.9	Master2	**Circui design methodology**[1]		12	12	
		Performance evaluation[1]		7,5		
s.10	ETN5	Final year internship supervision				26,5

Table 1: Number of hours and distribution of teachings at the University of Nantes for the 2019-2020 academic year.

These different courses are divided into lectures, tutorials, practical work and projects. Since 2004, my teaching load corresponds to an average annual volume of 275 hours (tutorial hours). I describe here more particularly the courses for which I am currently responsible or for which I significantly contribute.

- Design of digital circuits: this course focuses on methods, languages and tools for the design, simulation and production of digital circuits. I am responsible of lectures, tutorials, practical work as well as introduction sessions for incoming students in the fourth year of the ETN department. Over the years, the changes that I have made to this teaching relate to different aspects: teaching in English for a group of tutorials, introduction sessions for incoming international students, evolution of the progress of the project in order to ensure better acquisition with students, rewriting and reorganization of course material, addition of online tests for student self-assessment.
- Microprocessor systems: this course focuses on learning the operation of hardware and software resources that constitute a microprocessor system. I provide lectures, tutorials and practical work. Over the years, I have developed this teaching on different aspects: teaching in English for a tutorial group, change of microcontroller boards and development tools used in practical work, management of the electronic boards loan to the students (Microchip SAMD21), rewriting and reorganization of the course material, the content of the tutorials and the project covered in practical work, addition of online tests for student self-assessment.
- SoC design: this course focuses on learning advanced methods for designing systems on chips. This teaching is provided within a last year option of the ETN department (entitled Real-Time Embedded Systems). I provide the lectures, the follow-up of the case-studies approached by the students and the practical work. Over the years, I have developed this teaching on the following aspects: rewriting and

[1] Modules for which I am responsible

reorganization of the course material, use of new FPGA-based electronic boards (Xilinx ML402, Xilinx Zedboard), setting up of practical work sessions on the design of SoC using the Mentor Graphics and Xilinx Vivado suite of tools, conference by an engineer from industry on the subject of circuit testing.

- Hardware/software codesign: this course focuses on methods and languages (SystemC) for the joint design and verification of hardware/software resources of embedded systems. This teaching is provided within a last year option of the ETN department. I provide lectures and practical work. Over the years, I have developed this teaching on the following aspects: rewriting and reorganization of the course material, use of new FPGA-based electronic boards (Xilinx ML402, Xilinx Zedboard), set up of new practical works on SystemC.
- Architecture of embedded systems: this course focuses on the use of advanced resources of a microcontroller (sensor management, optimization of consumption, use of a DMA controller). This teaching is provided within the last Real-Time Embedded System option of the ETN department. For this teaching, I set up and I take care of the practical work sessions. I also set up a conference given by an engineer from industry on the subject of power management within embedded systems.
- Digital circuit design methodology: this course focuses on methods, languages and tools for the design, simulation and production of digital circuits. This teaching is given in English to students of the Master WET program. I set up this module and I provide lectures and practical work.
- Evaluation of the performance of embedded system architectures: this course focuses on learning the methods used to analyze the temporal properties of embedded system architectures (discrete event simulation, formal analysis methods). This teaching is given in English to students of the Master WET program. It is intended to be complementary to the teaching of codesign provided in the engineering degree with a particular focus on certain research works in this field. I set up this module and I provide the lectures.

In summary, my teaching activities focus on the use and design of the typical hardware and software resources of embedded systems. The following figure illustrates the distribution of the various lessons carried out, also positioning those carried out more episodically since 2004.

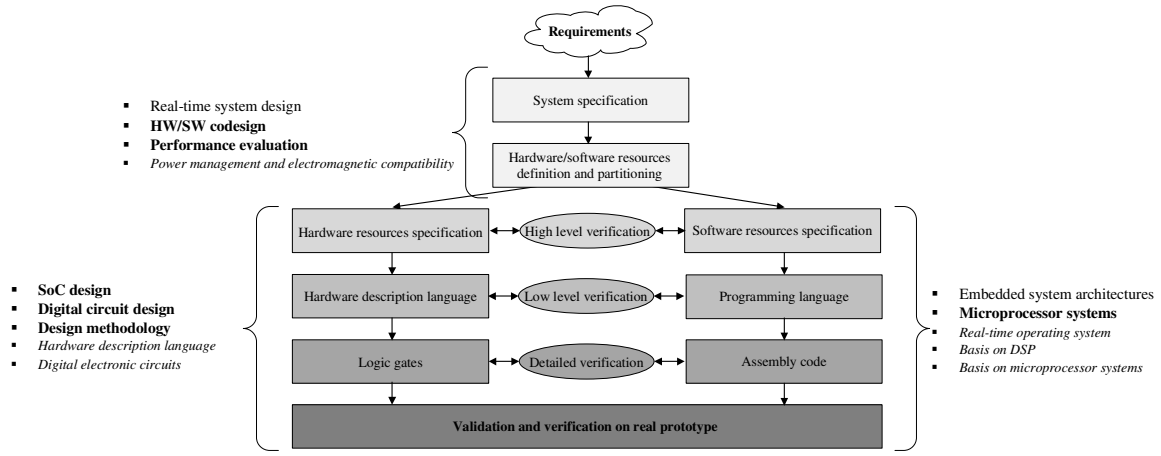


Figure 1: Lessons taught in the field of embedded system design. The titles in bold are modules which I assume the responsibility, the titles in italics are modules which I episodically participated.

B.2 Responsibilities and services to the community

I participate in the organisation of the ETN department of Polytech Nantes. I was thus able to take on the following responsibilities:

- 2005-2006: Responsible for international internships for the ETN department. This responsibility consisted of supporting students in their search for an international internship.

- 2006-2012: Responsible for the 3rd year of training within the ETN department. This responsibility involved setting up timetables, organizing semester juries and monitoring students.
- since 2015: Responsible for monitoring end-of-study internships for the ETN department. This responsibility consists of supporting students in their search for internships, the distribution of internship offers, the assignment of supervisors, the intervention of industrial partners presenting the electronics professions, interaction with the 'Company relationship' service of Polytech Nantes.

I was in charge of the organization and animation of two improvement councils of the ETN department. The first council took place in April 2017 and focused on the theme of follow-up and evaluation of internships. It allowed an assessment of the systems put in place within the ETN department in the monitoring of internships during the three years of training. The second council took place in October 2019 and focused on the theme of entrepreneurship. It allowed an assessment of the various actions implemented within the ETN department and to identify the changes required to raise awareness among students in the department about business creation.

From the 2017-2018 school year, I ensured the implementation of a new procedure for monitoring and evaluating end-of-study internships based on skills. This procedure was initially applied to the ETN department and then extended to Polytech Nantes. At Polytech Nantes, I have been part of the working group on internships since 2018. Since 2021, I have participated in the working group on the creation of a FabLab at Polytech Nantes.

I participated in the implementation of two partnership agreements between companies (ReflexCES and STMicroelectronics) and the Polytech Nantes school. Since 2017-2018, I have been organizing with the company MicroChip, a half-day visit to the Nantes site of this company for final year students of the ETN department. This visit is an opportunity for a presentation of the different activities of the company and potential positions. The constructive collaborations with these companies have benefited the students of the ETN department (technical projects, end-of-study internships, hiring) and the school (numerous donations of materials used in teaching and research, conferences by industrial speakers).

I participated in the seminar organized in November 2017 by the RFI Ouest Industries Créatives, on the theme "Creative methods used in educational transformation". The objective was to work on the redesign of teaching modules by adopting creative methods.

Every year, I participate in the school's various recruitment and promotion actions: open days, student fairs, recruitment interviews, presentations at IUTs (institutions of higher education providing technician level training). As such, in 2016 and 2017, I made various films for the presentation of the ETN department, final year options and student projects. These films are currently available on the school's website [2].

I was a member from 2005 to 2007 of the commission of specialists 61st joint section University of Nantes / Ecole Centrale de Nantes.

I participated in recruitment committees for the University of Nantes (in 2007 for position labelled 1169 and in 2010 for position labelled 1040), the University of South Brittany (in 2014 for position labelled 0939), INSA Rennes (in 2009 for position labelled 0169).

I initiated two partnership agreements between Polytech Nantes and the following foreign institutions: the University of Engineering and Architecture of Friborg in Switzerland (33 Polytech Nantes students on internship from 2007 to 2014), the University of Queensland in Australia (15 Polytech Nantes students on internship from 2015 to 2018).

I was president of one baccalaureate jury in 2008.

[2] https://polytech.univ-nantes.fr/les-formations/cycle-ingenieur/ingenieur-electronique-et-technologies-numeriques-2022795.kjsp?RH=1336143536804

C Summary of research activities

C.1 Addressed area of research

The topic addressed in the context of my doctoral thesis was the optimization and implementation of digital communication algorithms on heterogeneous platforms, combining programmable signal processing processors and dedicated hardware resources. This experience was an opportunity to discover the many facets and research issues associated with the design of embedded system architectures. This area of research evolves according to the requirements in applications and possibilities of execution platforms. Thus, since the early 2000s, the constant increase in the dissemination of embedded systems, described in particular in [1], has led to the emergence and concretization of concepts such as the Internet of Things, data science and embedded artificial intelligence. This development requires mastering the complexity of the hardware-software architectures of embedded systems in order to allow the design of devices operating under numerous constraints (functionalities, cost, size, reliability, execution time, power and energy consumption, etc.). The field of research in the design of embedded system architectures contributes to this in particular by proposing approaches, methods, models and tools used by the various participants in the design process.

Since my integration at the University of Nantes in 2004, my research activity has gradually focused on system-level design and on the evaluation of the performance of hardware and software architectures. The objective of this work is to promote the modelling and analysis of architectures in order to allow optimized definition of resources under time, power consumption and cost constraints. The positioning of the work carried out in this context is illustrated in Figure 2.

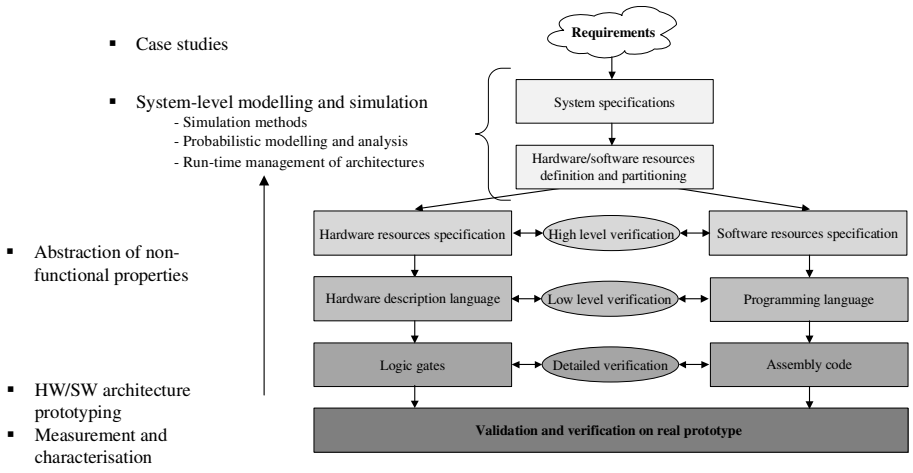


Figure 2: Position of research activities in the field of hardware/software architectures design of embedded systems.

The objective of this work relates to the definition of methods favouring the creation of performance models used for the definition of hardware and software resources of architectures. The originality of this work lies in the combination of the aspects covered: simulation, formal modelling, static analysis and online management, implementations and measurements on real targets. Also, this work was inspired by case studies from different fields of application (mobile radiocommunication, automotive, image processing). This work and the contributions made are detailed in the following chapters of this document. The supervision activities and collaborations set up are listed below.

C.2 Supervision activities

• PhD thesis co-supervision

QUENTIN DARIOL
Probabilistic symbolic simulation for embedded artificial intelligence multi-core systems — *Sep. 2020-xxx xxxx*
Funding: grant from WISE-International (research consortium in Pays de la Loire region).
Supervision rate: 50%
PhD thesis supervision: Sébastien Pillement (University of Nantes)

HAI-DANG VU
Fast and accurate performance models for probabilistic timing analysis of SDFGs on MPSoCs — *Oct. 2017-Mar. 2021*
Funding: grant from French Ministry of Higher Eduction (MESRI).
Supervision rate: 60%
PhD thesis supervision: Sébastien Pillement (University of Nantes)

SIMEI YANG
Evaluation and design of a run-time manager for ultra-low power multi-processor systems on chip — *Nov. 2016-June 2020*
Funding: China Scolarship Council (CSC)
Supervision rate: 40%
PhD thesis supervision: Sébastien Pillement (University of Nantes)

TAKIEDDINE MAJDOUB
Transaction-level modelling methods for simulation efficiency improvement for performance prediction of automotive architectures — *Oct. 2009-Oct. 2012*
Funding: grant from ANR (French agency for research), CIFAER project
Supervision rate: 60%
Supervision rate: Fabienne Nouvel (INSA Rennes)

ANTHONY BARRETEAU
Transaction-level modelling methods for resource definition of future mobile radiocommunication systems — *Oct. 2007-Dec. 2010*
Funding: grant from French Ministry of Higher Eduction (MESRI).
Supervision rate: 60%
Supervision rate: Jean-François Diouris (University of Nantes)

- **Supervision of Master thesis**

DHARMENDER SINGH
A dynamic correction method for fast yet accurate simulation of multiprocessor systems — *Feb. 2018-Jun. 2018*

JIATONG LI
Extraction of stochastic models for computation and communication time on a multicore Zynq platform — *Feb. 2017-Jun. 2017*

NADIA GHAZALI
Execution trace analysis methods for performance modeling of embedded system architectures — *Feb. 2015-Jun. 2015*

CHEN XI
Development of hardware-in-the-loop interfaces for ESL tools — *Feb. 2009-Jun. 2009*

MARIA CHEIK WAFA
Transaction level modelling of a FlexRay communication network — *Feb. 2008-Jun. 2008*

ANTHONY BARRETEAU
Automated generation of test environment from high level models *Feb. 2007-Jun. 2007*

ROMAIN GUIGNARD
Study and modelling of a SoC architecture in CoFluent Studio *Feb. 2006-Jun. 2006*

• Participation in juries and thesis monitoring committees

I participated in two thesis juries as an examiner:

- M. Pelcat (INSA Rennes, 2010), *Rapid prototyping and code generation for multi-core DSPs with application to the physical layer of 3GPP LTE base stations*
- G. Roquier (INSA Rennes, 2008), *Study of data flow models for multiprocessor software synthesis*

I participated in three thesis juries as a guest member:

- M. Balluet (Université de Rennes 1, 2021), *Creation of an image analysis system for an automated microscope*
- S. Cotard (Université de Nantes, 2013), *Contribution to the robustness of multicore real-time systems for the application domain*
- L. Dorie (Université de Nantes, 2007), *Performance models for the codesign of reconfigurable systems: application to the software radio*

Finally, I participated in the thesis follow-up committees of Mr. Balluet and Mr. Chagneau.

C.3 Contracts and research projects

• *Coordinator for University of Nantes of the RFI WISE-International pSSim4AI project with OFFIS institute, Germany* *2020-2023*
Object: This project deals with the evaluation and optimisation of performance of neural networks on multicore platforms.
Partner: OFFIS (Oldenburg, Germany)

• *Coordinator for University of Nantes of Hubert Curien Partnership PETA-MC with OFFIS institute, Germany* *2019-2020*
Object: This project deals with the development and evaluation of probabilistic models and analysis methods for timing and energy prediction of multicore systems.
Partner: OFFIS (Oldenburg, Germany)

• *Coordinator for University of Nantes of the RFI WISE PETA-MC international project* *2019-2020*
Object: This project deals with the development and evaluation of probabilistic models and analysis methods for timing and energy prediction of multicore systems.
Partner: OFFIS (Oldenburg, Germany)

• *Coordinator of an industrial grant with Intel company* *2016-2017*
Object: Development of a C-code generation tool from the environment Intel CoFluent Studio and experimentation with the Intel Galileo2 platform
Partner: Intel (Nantes)

• *Invited researcher at University of Queensland, Australia* *2012-2013*
Object: Research project about the improvement of simulation efficiency for early performance evaluation of multiprocessor systems.
Partners : Adam Postula, Neil Bergmann (School of Information Technology and Electrical Engineering, University of Queensland)

• *Coordinator for University of Nantes of the ANR CIFAER project* *2007-2011*
Object: Design of innovative communication infrastructures for the automotive domain. Study and evaluation of communication infrastructures performance.
Partners: Atmel, See4sys, IRISA, IETR

• *Coordinator for University of Nantes of a GDR ISIS Young researcher project* *2007*
Object: Optimisation of scheduling strategies for image and signal applications.
Partners: IETR (Image team, INSA Rennes)

More generally, I set up strong exchanges with the team of Dr Kim Grüttner of the OFFIS institute (Oldenburg, Germany), specialized in the design of embedded systems. These exchanges have materialized through various projects by stays for myself in Germany in 2016, 2018 and 2019, stays for doctoral students and researchers in France and Germany in 2019 and 2020, internships for engineering students in Germany (four internships carried out from 2016 to 2019). In 2020, we started a thesis co-funded by OFFIS and RFI WISE.

C.4 Publications

Over the period 2004-2021, since my installation at the University of Nantes, my scientific publications are distributed as follows:

- 7 publications in international journals
- 24 publications in international conferences
- 5 publications in national conferences
- 6 technical reports and project deliverables

The publications made as part of my doctoral thesis from 2000 to 2003 are divided into: 2 publications in international journals, 1 publication in a national journal, 3 publications in international conferences.

The full list of these publications is given in the appendix to this document.

C.5 Member of boards and committees

From 2008 to 2011, I participated in the council of the IREENA laboratory as an elected member.

From 2012 to 2015, I participated in the Polytech Nantes research council as an elected member.

C.6 Animation of research

Proofreading of articles for international journals: Eurasip Journal on embedded systems, Integration the VLSI journal, Eurasip Journal on advances in signal processing, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Elsevier Microprocessors and Microsystems.

Proofreading of articles for the Gretsi and DASIP conferences.

Participation in the program committee of the SDR-WinnComm conference in 2009 and 2010.
Participation in the organizing committee of the DASIP conference in 2007.

Member of the ISIS and SOC2 research groups.
Organizer of the inter-GDR ISIS SOC2 day 'Modeling and design of mobile radiocommunication systems' in 2010. Invited talk at CEA in 2015, GDR SoC in 2007, GDR ISIS in 2006, in National Microwave days (JNM) in 2005.

Participation in the science day organized at the University of Nantes in 2019.

Scientific referent of the 'Embedded and secure communicating systems' axis within the new ASIC team of the IETR, from January 2022. Member of the working group 'Scientific Animation' of the IETR laboratory.

Chapter 1

Introduction

This chapter introduces and positions my research activities carried out since my installation at the University of Nantes in 2004. This work falls within the field of the design of hardware and software architectures of embedded systems. It focuses more particularly on design activities at the so-called system level, that is to say located at the earliest stage of the design process, where the choices of architectures are initially made. The challenges associated with this field of research come from the need to master the complexity of designing architectures. This complexity is induced by the constant increase in the density and diversity of the hardware and software resources that constitute architectures. Therefore, the objective of this work is to improve design practices by proposing methods used for defining under constraints and verifying the non-functional requirements of the architectures (in our work, time, power consumption and cost). This chapter first presents the context of the research work as well as the addressed issues. The nature of the contributions made is then presented. Finally, the organization of this document is explained.

1.1 Context

1.1.1 Evolutions and trends in the design of hardware-software architectures of embedded systems

A simple observation of the objects manipulated by everyone in our everyday life underlines the strong dissemination of electronic devices. These devices are embedded in equipment belonging to numerous and varied fields of application. The notion of embedded system designates those devices for which the internal architecture consists of an association of hardware and software resources. The design of these architectures strongly depends on the constraints specific to the field of application: constraints on the cost of the solution, execution time, electrical power consumption, robustness, security, dependability. Gradually, the evolution of manufacturing technologies [2] has led to hardware resources that are ever denser in number of transistors and therefore capable of supporting many uses. These hardware resources then make it possible to support the execution of software whose complexity is measured in terms of millions of lines of code that can be executed. This technological evolution is accompanied by a necessary evolution of design practices. This allows the complexity of the designed systems to be controlled. These practices are essential in order to guarantee the development of hardware and software architectures under specific constraints and this according to constrained development time and costs.

In 1983, Daniel Gajski and Robert Kuhn proposed in [3] to organize electronic circuit design practices according to three complementary points of view: the physical point of view, the behavioural point of view and the structural. It was proposed to consider these points of view according to three levels of detail according to which the hardware resources of a circuit can be described: the transistor level, the logic gate level, and the architectural level (also called register transfer level). The design process of hardware resources can thus be presented as a transformation of a behavioural representation (the what) into a structural representation (the how) which, associated with a progressive refinement from the architectural level to the transistor level, leads to an actual realization in a given manufacturing technology. This proposal underlines the need to organize

the design based on structuring principles that can guide the designer in his work.

During the 1990s, integrated circuits made it possible to support complete systems on a chip, combining processor cores, local memories and dedicated hardware resources. To appropriately design these systems-on-chips (SoCs) and meet the related constraints, the selection of hardware and software resources must be carefully optimized. Different approaches, such as those presented in [4], emerged in order to guide the designer in the phases preceding the partitioning between hardware and software and the joint optimization of resources (*hardware-software codesign*) . Various proposals were made in order to establish levels of abstraction higher than the architectural level [5], [6] in order to capture the specificities of hardware and software resources at the earliest stage of the design process. Also, given the preponderant influence of interactions between resource categories, the importance of distinguishing computation mechanisms from communication mechanisms as soon as possible was highlighted [7], [8]. Added to these trends is the need to promote the reuse of existing resources (*Intellectual Property*, IP), at the different levels of abstraction in the design process. The concept of platform-oriented design (*platform-based design*) emerged [9] in order to have generic and easily adaptable execution supports according to application needs.

During the 2000s, systems-on-chip integrated several heterogeneous processors whose characteristics differ according to the specificities of the software to be executed. These years saw the emergence of so-called system-level design languages (SLD) including SystemC, standardized in 2012 [10], SystemVerilog [11] and SpecC [12]. These languages are adapted to the description of the characteristics of hardware and software resources. They notably support the separation of communication and calculation mechanisms as well as the possibility of modelling and simulating an architecture according to different levels of abstraction. Thus, these languages were used to carry out various design activities (resource sizing, software modelling and validation using virtual prototypes, integration of hardware-software resources) and were adopted by industry [13].

The evolution observed during the 2010s tends towards the integration of multi-core processors and an ever-increasing number of hardware resources within systems on chips. As underlined in [14] and [15], this constant increase in the complexity of systems justifies consolidating the methods used at the system level in order to favour the dimensioning under constraints of hardware-software architectures. According to these tendencies, system-level design therefore plays an essential role in the design process due to the fact that any modification of the initial choices generates additional costs that are all the higher as they are detected late.

1.1.2 Principles of electronic system-level design

As mentioned previously, electronic system-level design (ESL) aims to establish a hardware-software resource architecture that meets the different requirements of the system under study. Figure 1.1 illustrates some of the principles previously mentioned through the Y diagram used in 2009 by Gajski in [16][1]. This figure positions in particular the design activities at the system level. These activities assume the use of models capturing the different aspects of the studied architecture and making it possible to estimate its particular properties and to check compliance with the constraints. As illustrated in Figure 1.1, modelling a hardware-software architecture supposes to consider the following different aspects[2]:

- **Functional view**: it is described independently of the targeted technologies, focusing on the expected functionalities for the studied system. This view is represented according to two complementary approaches: structural and behavioural. This representation is generally based on the use of a computational model (MoC, *model of computation*) adapted to the characteristics of the functionalities to be described [18].
- **Executive view**: it displays the components capable of realizing the elements of the functional view. These components are conventionally classified into computation resources, communication resources and storage resources. Subsequently, we will also use the term platform to refer to this view.
- **Physical view**: designates the result of the deployment, the allocation, of the elements described in the functional view onto the components of the executive view.

[1] In this simplified representation of the Y diagram, only four levels of abstraction are identified. In more detailed versions, five levels are exhibited, highlighting the functional analysis of the systems to be designed. [2] The terms used to designate these views frequently differ. At this stage, we use the terms used in [17].

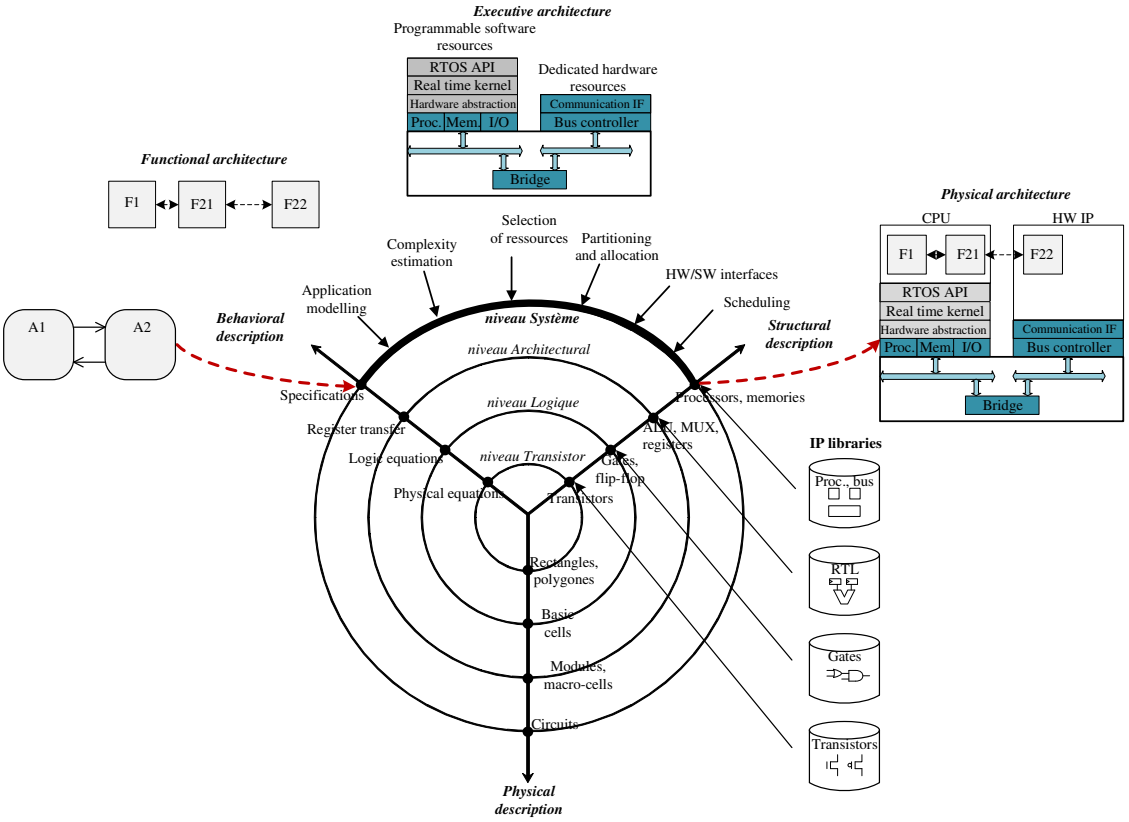


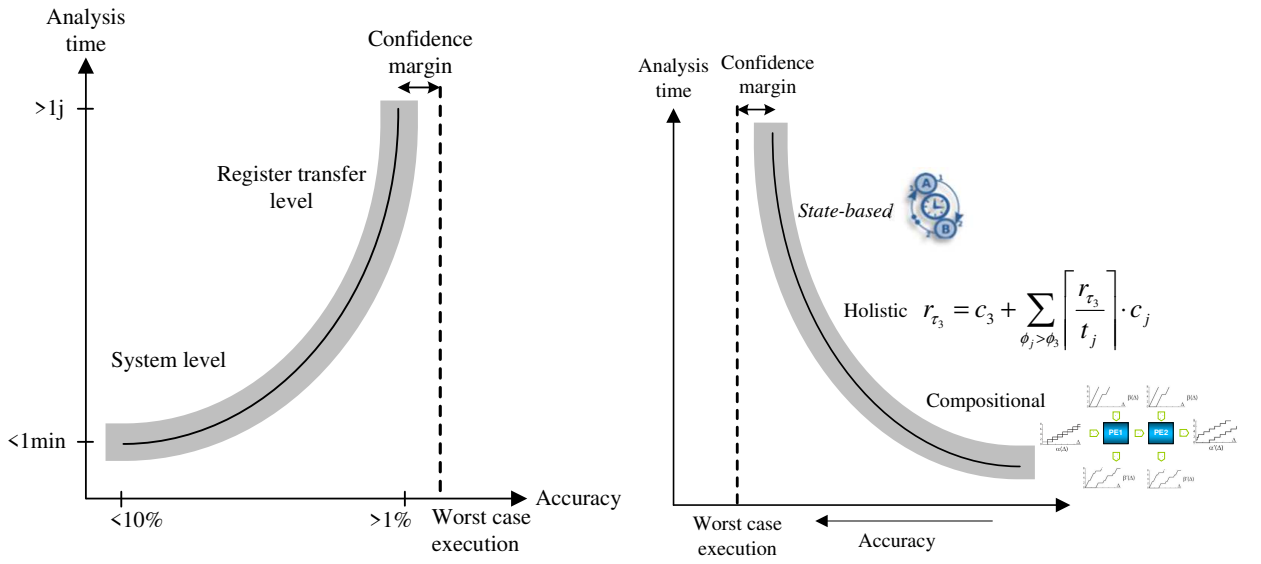
Figure 1.1: The Y diagram with detailed at the system level, inspired from [16]. The main activities led at the system level are identified.

As indicated in Figure 1.1, complementary activities are associated with the establishment of these views, in order to favour the selection of resources (through the estimation of the complexity of the elements of the functional architecture) and to establish policies for arbitration and scheduling of shared resources.

The obtained physical architecture corresponds to a description of the hardware and software resources that will have to be implemented in given hardware description and programming languages. To evaluate the relevance of the architecture choices, the physical architecture is used beforehand in order to evaluate the performances delivered: we then speak of a *performance model*. The term *performance* here designates any property that can be estimated (*e.g.*, end-to-end latency, input or output throughput, energy consumed). This representation is used in order to evaluate the relevance of the architecture choices and to carry out an appropriate dimensioning. As we will detail later, this representation requires prior estimations made in order to characterize each element of the model (*e.g.*, execution time of a function of the functional view on a computation resource of the executive view). These estimates come from knowledge on the elements of the model (resulting from measurements on target or from results of detailed simulations). The analysis of these models therefore makes it possible to quantify the performances achieved. By considering different configurations of physical architectures, it is therefore possible to explore a given design space in order to identify an optimized architectural solution that best satisfies the constraints considered.

Different approaches can be adopted in order to analyze the properties of the performance models thus created. As illustrated in Figure 1.2, the efficiency of these analysis methods can be judged considering the criteria of accuracy and speed of analysis. As presented in [19], it is common to classify these approaches into two categories:

- **Simulation-based approaches**: they consider a performance model that can be executed and thus allow the analysis of the evolution of the architecture model according to different operating scenarios. However, they suppose an intensive evaluation in order to be able to cover many possible operating situations, without guaranteeing an identification of the worst execution cases. Part (a) of Figure 1.2 illustrates the effectiveness of simulation-based performance evaluation methods according to the level



(a) Simulation-based approaches

(b) Formal approaches

Figure 1.2: Illustration of the compromise between accuracy and analysis time for different performance evaluation methods, (a) simulation-based approaches, (b) formal approaches.

of abstraction considered. It is commonly established that an accuracy of the order of 10 % can be achieved with such models and this for a reduction of a factor greater than 1000 in simulation time compared to models accurate to the cycle of clock near [16]. Examples of academic approaches based on system-level simulation are presented in [20]. As illustrated in Figure 1.2, simulation traditionally leads to underestimating the properties with regard to the worst case of operation. This is especially due to the difficulty in establishing the required stimuli allowing to identify the boundary situations.

- **The so-called formal approaches**: these consider an expression of the properties of the architecture studied according to a precise mathematical formalism. Two categories of formal approaches can be identified: those based on analytical expressions of the properties to be evaluated (*analytical approaches*) and those based on the exploration of the space of possible operating states (*state-based approaches*). Such approaches are notably presented in [21], [22]. Notably, some of these approaches allow a more exhaustive exploration of the possible operating situations in order to allow the evaluation of the worst execution cases. Part (b) of the figure 1.2 illustrates the relative position of these different evaluation approaches according to the criteria of accuracy and analysis effort. Analytical approaches (distinguished between holistic and compositional in the figure 1.2) traditionally lead to over-estimates with respect to the worst operating case. So-called *state-based* approaches turn out to be limited in terms of the complexity of the systems that can be considered given the explosion of the space of states to be analyzed.

In [R1-1], we detailed and compared different approaches recently used for performance evaluation of multiprocessor architectures of embedded systems. As discussed in the next section, the nature of the platforms considered and the way their basic components are associated significantly influence the effectiveness of these approaches.

1.1.3 Issues and addressed problems

The functional and non-functional requirements applied to embedded systems lead to an increase in the complexity of the platforms used. This complexity is reflected both in the micro-architecture of the processors and in the overall architecture of the platforms. Current trends correspond in particular to an increase in the execution parallelism offered. Thus, these platforms are increasingly multiprocessor and multicore in nature in order to deliver high computing performance and optimized energy consumption. The organization of these platforms then consists of a set of processors, of identical or distinct natures, interconnected through

networks or communication buses and having different levels of memory hierarchy [23], [24]. In this context, the complexity of design lies in the number and diversity of resources to be apprehended but also in the nature of the interactions between resources. The influence of possible interactions between computing resources can be evaluated according to the so-called compositionality criterion (*compositionality*) [25]. This criterion expresses the degree to which the behavior of a given system can be induced from the behaviors of each of its constituents. This property depends on the influence of shared resources and therefore on the level of interdependence between components. Figure 1.3, inspired by the classification proposed in [26], positions the possible organizations of a multiprocessor system according to the degree of compositionality and the influences of the shared resources. According to this classification, a so-called fully compositional organization (*fully timing compositional*) presents limited interactions allowing to deduce the global properties of the architecture taking into account the properties of each elementary component. Conversely, an organization that does not present the property of compositionality will bring into play phenomena that do not allow such a deduction, further complicating the prediction of non-functional properties.

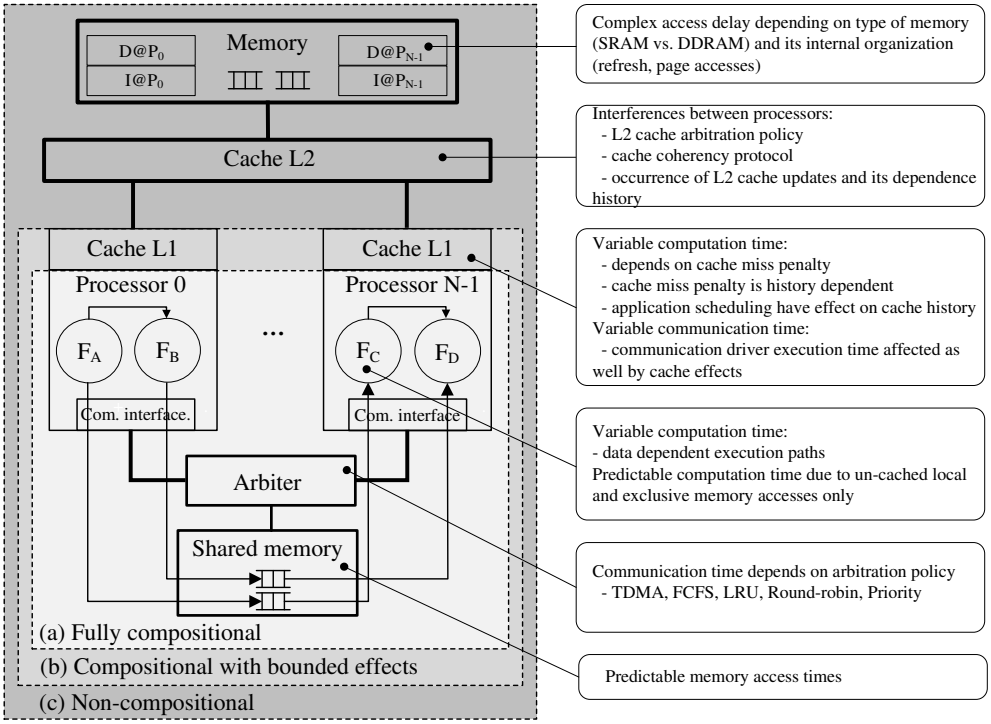


Figure 1.3: Different levels of complexity in shared resources of multiprocessor systems.

The complexity of the architectures to be considered leads to a difficult coverage of the operating cases which can lead to the non-detection of problematic situations and therefore a non-respect of certain constraints. Late detection in the design process of such defects can lead to additional costs for projects or even serious failures in the case of systems in operation. Thus, the quality of the system-level models used for the prediction of the non-functional properties of the architectures proves to be essential in order to allow an analysis and an effective dimensioning. There are two observations that motivate my work:

- the modeling effort at the system level must be controlled while allowing the creation of models presenting a satisfactory compromise in terms of speed of analysis and precision.
- The effort of analysis (of simulation in the case of the work presented below) must be mastered to allow the study of ever more complex architectures within reasonable timeframes.

These two points motivate the contributions made in order to favor the analysis and the optimization of the performances of the architectures during the design of the systems. In addition to these two axes, the dynamic management of resources (that is to say during operation) also represents an interesting approach in order to promote the optimization of the performance of architectures. In the field of multiprocessor architectures, such an approach leads to adapting the use of hardware resources in order to optimize the overall performance

under constraints of execution time or power consumption. Thus, through certain co-supervised theses, I was also able to take an interest in this field both in order to propose original management strategies and also in order to promote the modelling and simulation of such management. The contributions developed in the context of my research therefore fall within these three axes.

1.2 Contributions to the research field

1.2.1 Development of activities

My research activities were successively carried out within the MCSE team (modeling and design of electronic systems) of Professor Jean-Paul Calvez and then within the Communication Systems team of the IETR. The MCSE team was at the origin of the CoFluent Studio modeling and simulation environment, now owned by the Intel company [27]. Initially, my activities were positioned towards this environment, and it served as a means of demonstrating and validating the contributions made. Subsequently, the scope of my work was extended by considering models directly described in open programming languages such as SystemC, in particular within the collaboration developed with the OFFIS institute in Germany. The nature of the case studies covered in my work has also evolved. Initially, in the following of my PhD thesis work, my attention focused on the study and optimization of the physical layer of communication systems for platforms combining programmable processor and hardware accelerators. Subsequently, I became more generally interested in the study of data flow-oriented and distributed applications within multiprocessor systems. Figure 1.4 illustrates the temporal progress of the work carried out, through the co-supervised theses, the projects and the main themes addressed. It highlights in particular the topics covered in the framework of the co-supervised theses in connection with the research themes.

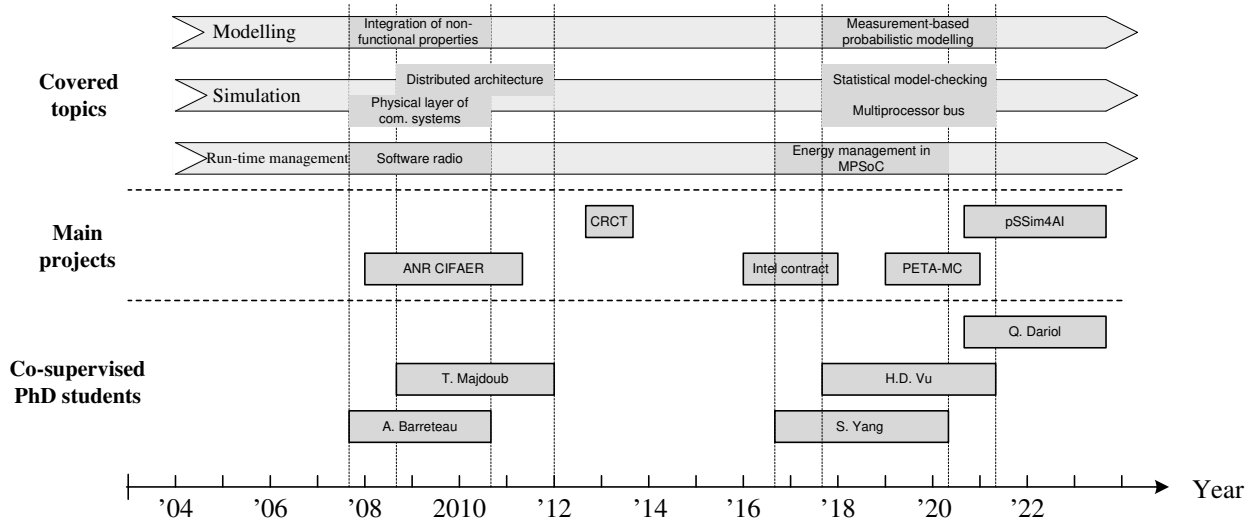


Figure 1.4: Position of the covered research topics, main projects and co-supervised PhD thesis.

As mentioned previously, the research work carried out was organized around three complementary axes: the creation of performance models, the improvement of simulation efficiency and the study of online resource management. Each co-supervised thesis was an opportunity to approach these axes in an original way.

1.2.2 Contributions to improving system-level simulation efficiency

Given the increasing complexity of embedded system architectures, improving simulation efficiency is an important issue. A first contribution focused on the definition of a simulation approach and aimed at reducing the required analysis times while preserving the accuracy of the estimates obtained. This approach is hybrid in nature: it combines simulation and formal models. This approach aims to reduce the number of calls to the simulation engine by disregarding the many processes generated by traditional simulation approaches at the system level. The principles associated with this method are detailed in Chapter 2 of this document.

Different case studies made it possible to quantify the potential contribution of the proposed simulation approach. These case studies focus on different hardware-software architectures of embedded systems. Two case studies of significant complexity are presented in Chapter 3 of this document. The first presented case-study corresponds to the analysis of a distributed architecture of embedded controllers for the automotive field. The second presented case-study focuses on the analysis of a multiprocessor architecture built around the ARM AXI4-Lite bus standard.

Secondly, I was interested in the adoption of statistical model-checking simulation (SMC) methods for the analysis of the temporal properties of multiprocessor architectures. The interest of such methods is to control the coverage of the simulations carried out on performance models. The adoption of these methods and the associated modelling flow are presented in Chapter 5.

1.2.3 Contributions to the creation of system-level models of hardware-software architectures

First, a generic execution model has been proposed to serve as a basis for modelling and simulation at the system level. The objective of this model is to favour the creation of simulable models used for performance evaluation of hardware-software architectures. The principle of this model is presented in Chapter 2 of this document in connection with the proposed simulation approach.

Different case studies have validated the proposed generic model. A first case study is presented in Chapter 3 and concerns the modelling of communication resources for multiprocessor architectures. A second case study concerns the modelling of computing resources for systems in the field of mobile radiocommunication. This case study is presented in Chapter 4 of this document.

Secondly, I was interested in the contribution of probabilistic models and their creation according to an approach based on measurement (*measurement-based probabilistic timing analysis*) within the framework of the study of multiprocessor systems. The purpose of this approach is to obtain reliable estimates used to correctly calibrate performance models. The approach developed is presented in Chapter 5. The accuracy and speed of the models created are evaluated on case studies in the field of image processing.

1.2.4 Contributions to online management of hardware-software architectures

A first line of work focused on the modeling of systems for which the nature of the functionalities supported, and therefore the workloads induced, evolve during operation. A first contribution focused on extending the modelling and simulation possibilities of such architectures within the CoFluent Studio environment. This contribution was then used in the so-called software radio context in order to allow the performance analysis of a mobile radio terminal according to different use cases. This work is presented in Chapter 4.

Secondly, still in connection with the study of systems whose supported functionalities evolve during operation, different allocation and scheduling strategies have been proposed in order to optimize the energy efficiency of multiprocessor platforms. These strategies were applied within homogeneous and heterogeneous sets of processors. This work was evaluated by the study of different data flow oriented applications in the multimedia domain. Also, a simulation approach has been proposed in order to allow the simulation of the online management of computing resources. The proposed strategies and the simulation approach are presented in Chapter 4 of this document.

1.3 Organization of the document

The following four chapters detail the contributions made. Chapter 2 presents the principles associated with the hybrid modelling and simulation approach proposed in order to favour the creation of hardware-software architecture models with increased simulation speed and a preserved level of precision. Chapter 3 deals with the study of hardware-software architectures for which the modelling of communication resources is essential. The contribution of the approach presented previously is thus illustrated and the gains in terms of simulation speed are evaluated. Chapter 4 presents the work carried out for the modelling of computation and memory loads as well as the online management of the computing resources of multiprocessor architectures. Chapter

5 presents the adoption of probabilistic models and methods for the analysis of the temporal properties of MPSoC architectures. Finally, in the appendix to this document, the list of publications and references used in this document are given.

1.4 Publications related to Chapter 1

Research report

[R1-1] R. Stemmer, H.-D. Vu, M. Fakih, K. Grüttner, S. Le Nours, and S. Pillement, "Feasibility Study of Probabilistic Timing Analysis Methods for SDF Applications on Multi-Core Processors," IETR ; OFFIS, Research Report, Mar. 2019. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02071362.

Chapter 2

Contributions to system-level modelling and simulation

2.1 Introduction

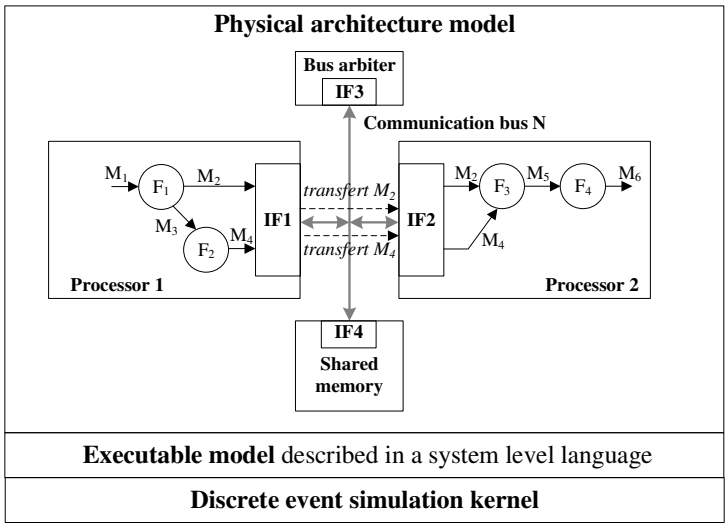
This chapter presents the work carried out in the field of system level design in order to improve the efficiency of the models used for the evaluation of the performances of the hardware-software architectures of embedded systems. In this field, many approaches have been proposed to promote the creation of simulable models of such architectures [20]. These models are used to estimate and compare the performance of candidate architectures early in the design process; they thus make it possible to explore the design space. In the case of simulation-based approaches, the models are executed according to the principles of discrete-event simulation in order to account for parallelism and synchronizations within the studied architectures [28]. In this context, the role of the simulation engine resides in the advancement of the simulation time and the management of the various processes constituting the models created. The proposal made in this work aims to limit the number of calls to the discrete-event simulation engine while seeking to preserve the accuracy of the models considered. The work carried out led to the definition of a hybrid approach, combining simulation and calculations, and leading to significant gains in terms of simulation speed while preserving the accuracy of the estimates. This chapter specifies the nature of the contribution made and the methods and models proposed. We also discuss some results illustrating the potential and limitations of this approach.

This work finds its origin in some of the proposals of the thesis of Mr. Anthony Barreteau carried out from 2007 to 2010 and during preliminary discussions with Professor Jean-Paul Calvez. The research leave conducted in 2012-2013 was an opportunity to develop and concretize these ideas. Since then, this approach has been extended and tested through various case studies and also as part of Mr. Dharmender Singh's Master's internship in 2018.

2.2 Addressed issue

The principles related to the modeling of a physical architecture at the system level have been mentioned in chapter 1. The functional view of an architecture is organized in the form of elements subsequently called functions or actors, interacting through coupling relationships. Function behavior typically corresponds to a sequence of computational and communication primitives. Control statements (*e.g.*, finite iteration, infinite iteration, alternate) are used to establish the sequence of these primitives. At this stage, we are not making any assumptions about the nature of the communication mechanisms used, as these may be of different natures (synchronization without data exchange, shared variable, message queue, appointment). Part (a) of the 2.1 figure gives an example of the organization of a physical architecture represented from the functional and executive views.

The functional structure represented comprises four functions, noted from F_1 to F_4, interacting through coupling relations, M_1 to M_6. These elements are allocated on the resources of an executive platform



(a)

```
Process_F1 :
while(1) {
   ReadToken(M1);
   Execute(token);
   WriteToken(M2);
   Execute(token);
   WriteToken(M3);
}
```

```
Process_F2 :
while(1) {
   ReadToken(M3);
   Execute(token);
   WriteToken(M4);
}
```

```
Process_F3 :
while(1) {
   ReadToken(M2);
   Execute(token);
   ReadToken(M4);
   Execute(token);
   WriteToken(M5);
}
```

```
Process_F4 :
while(1) {
   ReadToken(M5);
   if (token.DataSize>Threshold)
      Execute(token);
   else
      //NOP
   WriteToken(M6);
}
```

(b)

Figure 2.1: (a) Example of a system-level model with three architecture viewpoints: functional, executive and physical, physique, (b) description of functions behaviours with elementary communication and computation statements. The execution semantic related to each statement needs to be defined such as illustrated in Figure 2.7.

made of two computation resources (here, programmable processors) connected through communication resources (interfaces, bus and shared memory). It is important to understand that such models do not necessarily integrate the functionalities provided by the application: they express the workloads induced by the application (*workload model*). Part (b) of figure 2.1 illustrates the behavioral description of functions using a computation primitive (*Execute*) and two primitives for writing and reading through message queues (*WriteToken*, *ReadToken*). Such a description serves as the basis for the creation of executable models described in a programming language supporting the mechanisms expressed within the architecture model (*e.g.*, SystemC, SpecC). Among the approaches in the academic field that implement these principles, we can cite SCE [29], Sesame [30], SystemCoDesigner [31] as well as those presented in quoteKangas-06,Kreku-08,Arpinen-09. Among the industrial approaches, we can cite the Intel CoFluent Studio [27], Space Codesign [32] and Visualsim from Mirabilis Design [33] environments. These approaches differ in particular according to the model of computation considered to describe the applications, the programming language used to describe the executable models or the possibility of automatically generating lower-level descriptions.

During the execution of such a model, each primitive involves a workload on the communication and computational resources of the platform (*e.g.*, number of operations to be performed, size of data to be swap). The computation or communication times associated with these workloads must therefore be established. Their values typically come from analyzes carried out elsewhere, through measurements on real prototypes, detailed simulations or static analyzes of the codes associated with each function, as illustrated in [34]. The evolution over time of an architecture model thus formed reflects the workloads successively induced by the application, and this taking into account the limited capacities of the resources of the platform. Figure 2.2 illustrates such an evolution for the example of figure 2.1.

Platform resource usage is observed as the application runs. The specific simulation instants, symbolized by vertical arrows, correspond to the start and end times of the execution of the computation and communication primitives of the application. They correspond to the instants of call to the simulation kernel in charge of the execution of the processes that constitute the executable model and the advancement of the simulation time t_s. Thereafter, the specific simulation instants of the studied model will be noted in the form $x_i(k)$. By way of example, $x_{WrM2S}(k)$ designates the k-th instant at which the write operation within the relation M_2 begins. More specifically, we note u the instants corresponding to the inputs of the studied system and y the instants corresponding to the outputs. Thus, in the figure 2.2, $u(k)$ designates the k-th instant at which the function F_1 receives data through the relation M_1 and $y(k)$ the k-th instant at which F_4 produces data through the relation noted M_6.

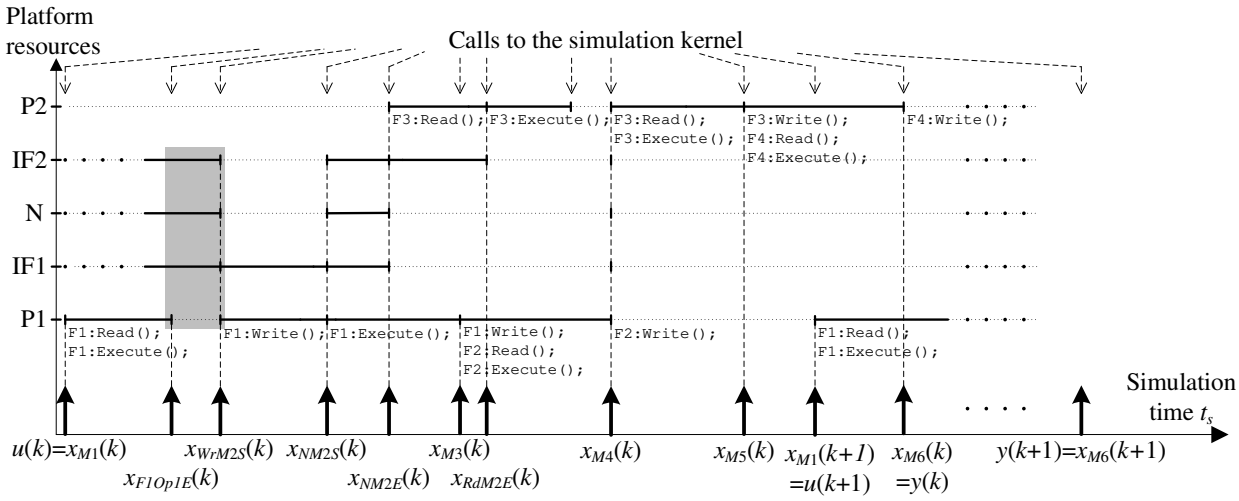


Figure 2.2: Discrete event simulation of the architecture model in the figure 2.1. In this example, the instants associated with the communication of M_2 via the interfaces IF1 and IF2 and the communication bus N are detailed. The instants associated with the communications of the other relations are not detailed for the sake of readability. The shaded part corresponds to a situation of contention on the interface IF1.

Such an execution makes it possible in particular to analyze the impact of the resources of the platform on the execution of the application, this taking into account the platform arbitration strategies. Additional delays are thus introduced into the execution of the elements of the application due to concurrent access to shared resources. The situation illustrated in the figure 2.2 corresponds to the situation where the interface IF1 is busy carrying out a transfer on the communication bus and cannot immediately receive the data carried by the relation M_2.

The construction of accurate performance models first requires the establishment of reliable abstractions, quantifying the duration associated with the communication and computation primitives used to describe the studied architecture. The influence of shared resources must then be described at the considered level of abstraction. For this type of model, the simulation duration is essentially linked to the number of calls to the simulation engine, the operations described being of limited complexity. The fact of having to consider ever more components within an architecture (functions, relations, elements of a platform) implies controlling the simulation effort of the created models and this while delivering an acceptable level of precision.

2.3 Principles of the proposed system-level approach

The proposed approach aims to limit the number of calls to the simulation engine while preserving the level of accuracy achieved by traditional approaches. To do this, this approach uses the properties expressed within the performance models created and which establish, in particular, the dependencies between components of the application as well as the strategies for arbitrating the resources of the platform. Based on this knowledge, the proposed approach consists in predicting, during the simulation, the future instants of execution of the model without requiring calls to the simulation engine. Figure 2.3 illustrates this simulation principle by comparing it with the approach previously illustrated.

Part (a) of the figure 2.3 illustrates the different simulation instants and the different calls to the simulation engine occurring during an execution of the model illustrated in the figure 2.2. The instants $u(k)$ and $y(k)$ correspond respectively to the instants when the data carried by the relations M_1 and M_6 are available at the input and at the output of the model. Each instant $x_i(k)$ corresponds to a simulation instant of the model. As an example, the instants denoted $x_{NM_4S}(k)$ and $x_{NM_4E}(k)$ correspond to the instants of start and end of the use of communication bus N during transfer of data associated with relation M_4. Part (b) of the figure illustrates the execution of a model created according to the proposed approach, for which time $y(k)$ is predicted at time $u(k)$. This prediction corresponds to a calculation symbolized by the action `Compute_y()`, relating to the set of instants $x_i(k)$. This calculation is based on the knowledge of the temporal dependencies expressed

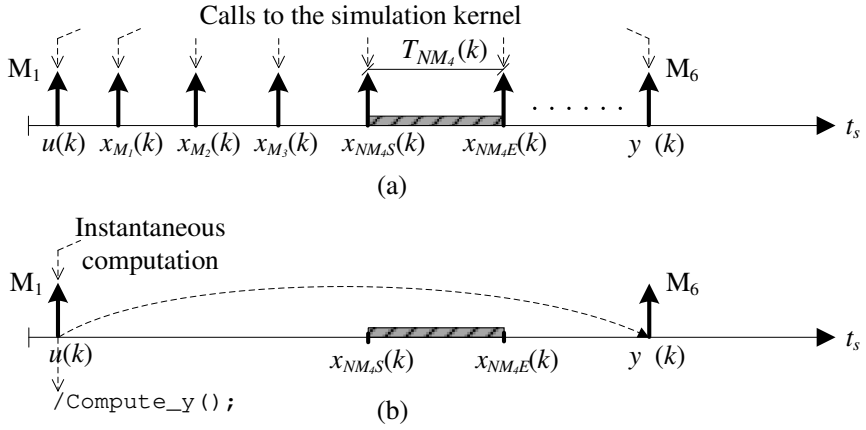


Figure 2.3: (a) Execution of a model with classical simulation approach, (b) execution of a model with instantaneous computation of future simulation instants.

within the initial architecture model. At this stage, the expression of these dependencies can be made by a set of state equations expressed in the form:

$$\begin{cases} X(k) &= f(X(k-1), u(k)) \\ y(k) &= g(X(k), u(k)) \end{cases} \tag{2.1}$$

The state vector X designates here the set of instants x_i involved in the calculation of $y(k)$. f and g reflect the dependencies between the instants y, the current and previous values of the state vector as well as the instants u. As mentioned above, these equations are established taking into account the dependencies between the elements of the architecture model. The proposed construction for these state equations is discussed in the next section.

An essential point of this approach concerns the fact that, during the simulation, certain situations can impact the prediction made and lead to degrading the accuracy achieved. This phenomenon occurs in particular in the case of the use of shared resources of the execution platform. Figure 2.4 extends the example of figure 2.3 and illustrates this phenomenon.

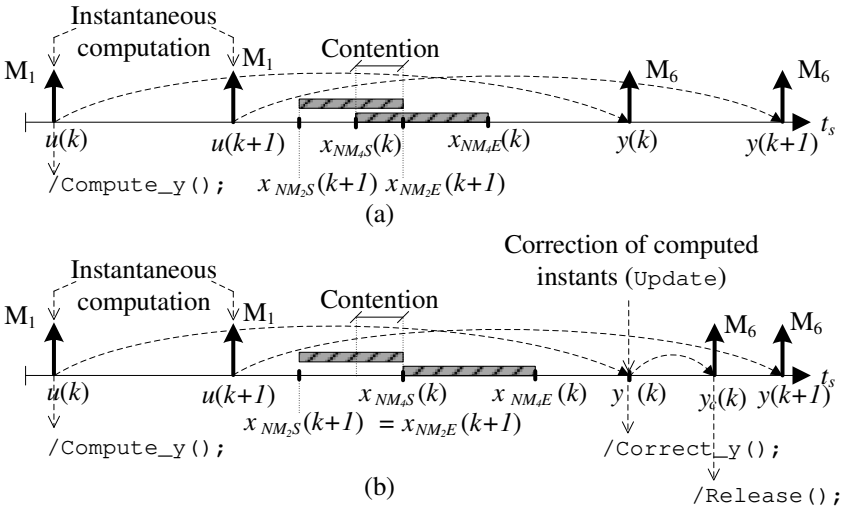


Figure 2.4: (a) Situation with erroneous prediction of instants when a shared resource is used, (b) principle of the correction method of resource usage instants.

Part (a) of the figure 2.4 illustrates the situation where the use of the shared resource occurs before the previously predicted instant $x_{NM_4S}(k)$. In the situation presented, M_2 uses bus N before M_4. Consequently, the

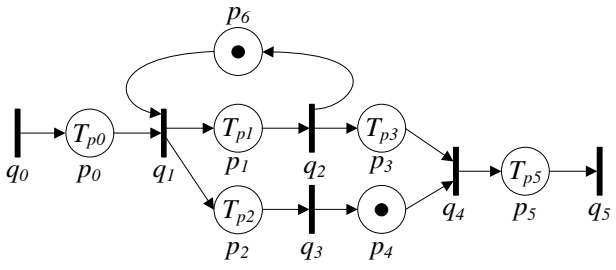
contention phase on the communication bus N is not correctly established and the model thus constructed delivers degraded precision. The approach proposed to correct such phenomena is illustrated in part (b) of the figure 2.4. It consists in identifying the situation of contention taking into account the value of the state vector. In doing so, it is possible to retroactively correct the prediction made and thus to take account of the effect of the contentions on the predicted instants. On the given example, the identification and the correction are carried out at instant $y(k)$. Instant $y_c(k)$ then corresponds to the newly established exit instant taking into account the contention phase that has occurred. This principle was also used in the context of Mr. Singh's Master for the case of multiple applications allocated on the same resource platform [J2-1].

In this field of study, many approaches have been proposed in order to limit the number of simulation events while seeking to maintain the accuracy of the models created. A first example concerns the two coding styles proposed in the SystemC TLM2.0 [10] standard: the so-called *approximately-timed* (TLM-AT) style and the so-called *loosely-timed* style (TLM-LT). The TLM-AT approach leads to the description of numerous synchronization points between processes, involving numerous interventions of the simulation engine (*lock-step simulation*). The TLM-LT approach allows processes to run independently of the simulation engine, for a given time interval (*temporal decoupling*). Such an approach is notably used to speed up the simulation of models used for software verification on virtual prototypes but may involve inaccuracies about the influence of shared resources [35], [36]. The principle consisting in predicting the future instants of evolution of a model in order to abstract some of its elements was notably considered in the so-called ROM approach (*Result-Oriented Modeling*) proposed by Schirner and Dömer in [37], [38]. A retroactive adaptation of the predicted instants was also considered in order to be able to correct the possible influence of shared resources. However, the method used to predict the synchronization instants during the simulation was based on a priori knowledge of the durations to be used. Different approaches combining simulable system-level models and calculations have also been proposed in order to offer an optimized compromise between simulation speed and accuracy [39]–[41]. The proposed approach differs from this work in the way simulation and calculations are associated. The prediction of simulation instants is based on a description that express the dependencies between simulation instants. This approach also makes it possible to retroactively correct the calculated instants in the case of conflicts of access to shared resources. This approach was successively described in international conferences [C2-1]–[C2-3] and in international journals [J2-1], [J2-2]. Hereafter, the two principal aspects of this approach are explained: the establishment of the state equations then the executable model used in simulation.

2.4 Creation of state-based models

An essential aspect of the proposed approach supposes to establish the dependencies between simulation instants of the elements of a given architecture, these instants then being calculated during the simulation without using the simulation engine. These equations are established taking into account the knowledge of the physical architecture model. Thus, the application model establishes the dependencies between functions and this taking into account different communication protocols between functions. The platform model establishes the organization of communication and computation resources as well as the arbitration strategies for access to shared resources. We used the formal model of Petri nets in order to rigorously establish the dependencies between simulation instants. More specifically, we used the formalism of timed Petri nets (*timed Petri nets*) to establish the existing dependencies for each of the primitives used within the architectural models studied. This choice lies in the fact that this formalism has a direct correspondence with the (max, plus) algebra. This algebra represents the appropriate framework for expressing in the form of state equations the dependencies within a discrete event model [42]. Timed Petri nets correspond to an extension of Petri nets for which the time constraints are given by a single value. In our case, these values represent minimum occupation durations of the places by the tokens. Such a network is defined by the tuple $\mathcal{T} = (\mathcal{P}, \mathcal{Q}, {}^\bullet(.), (.)^\bullet, \mu_0, T, \rho)$, for which $\mathcal{P}$ denotes a nonempty finite set of places, $\mathcal{Q}$ denotes a nonempty finite set of transitions, ${}^\bullet(.) \in (\mathbb{N}^{|\mathcal{P}|})^{|\mathcal{Q}|}$ denotes the upstream incidence function, $(.)^\bullet \in (\mathbb{N}^{|\ mathcalP|})^{|\mathcal{Q}|}$ is the downstream incidence function, $\mu_0 \in \mathbb{N}^{|\mathcal{P}|}$ is the marking initial of the network, $T \in \mathbb{R}^{|\mathcal{P}|}$ is the set of durations on places. A transition q is active if each upstream place contains at least one token. Basically, triggering a transition results in removing a token from each place upstream and adding a token to each place downstream of the transition. Additional weights can

be carried by the arcs, indicating the number of tokens needed to trigger a transition. Occupation duration in a place is the time spent in a place before contributing to the activation of downstream transitions. Part (a) of the figure 2.5 illustrates this formalism for a simple network formed of seven places and six transitions.



$$
\begin{aligned}
x_1(k) &= max(T_{p0}(k) + x_0(k), x_2(k-1)) \\
x_2(k) &= T_{p1}(k) + x_1(k) \\
x_3(k) &= T_{p2}(k) + x_1(k) \\
x_4(k) &= max(T_{p3}(k) + x_2(k), x_3(k-1)) \\
x_5(k) &= T_{p5}(k) + x_4(k)
\end{aligned}
$$

(a) (b)

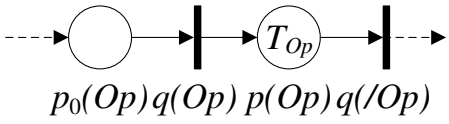
Figure 2.5: (a) Notation of timed Petri nets used to describe the dependencies between simulation instants, (b) expression of the transition instants in the illustrated network.

We note $x_j(k)\ j = 1,\ldots,|\mathcal{Q}|, k \geq 0$, the instant at which the transition q_j is activated for the k-th time. Taking into account the construction rules of such a network, the relations between instants of transition can be expressed according to two operators: addition and maximization. The addition expresses a shift in time taking into account the time spent in a place, the maximization reflects the effect of a wait, of synchronization. Part (b) of the figure gives the instants of the transitions of the illustrated network.

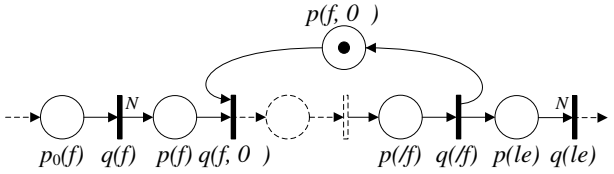
We adopted this formalism in order to rigorously express the existing dependencies within an architecture model as illustrated on the example of the figure 2.1. For a given architecture model, the expression of the dependencies is established by successively considering the elements of the application model then the constraints introduced by the allocation on the resources of a platform. Such an approach was notably presented in [43] for the case of the EFFBD notation (*enhanced function flow block diagram*). In our case, we tried to remain independent of a specific notation by considering the mechanisms encountered in many modeling approaches. First, a set of elementary patterns must be established taking into account the meaning given to each element of the application model. Each pattern can be described by an elementary Petri net $\mathcal{T}^n$. The figure 2.6 illustrates some patterns established to describe the dependencies associated with the calculation mechanisms and the control instructions encountered for the description of the application model.

The network associated with the complete application model is obtained by connecting each elementary pattern n required taking into account the description of the behavior of each function and the organization between functions. The figure 2.7 illustrates the patterns considered in the case of communications between functions according to the rendezvous and FIFO protocols of infinite capacity. The network associated with the application model is then completed taking into account the considered allocation constraints and the characteristics of the elements of the platform. The execution times of the computation and communication primitives are taken into account in the form of an occupation duration in places. The limited capacities of platform resources are taken into account in the form of additional places and arcs on the network. On the example of the figure 2.8, part (a) shows the dependencies between transition times for the case of the communication of M_2 through the interfaces IF1 and IF2 and the communication bus N. In this example, the places noted $p(IF1)$ and $p(IF2)$ indicate the fact that each interface IF1 and IF2 cannot simultaneously manage the transfer of data on the bus and receive or deliver new data. Part (b) of the figure 2.8 illustrates the construction made in the case of the function F_4 described in the figure 2.1. This construction is obtained by combining the elementary patterns illustrated in the figures 2.6 and 2.7. In [J2-2], a complete example is presented to illustrate the construction process implemented.

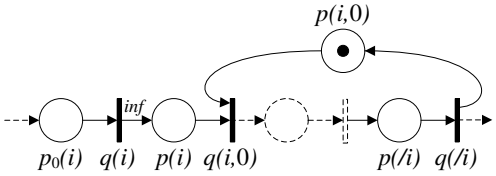
The establishment of such a Petri net makes it possible to rigorously establish the dependencies within an architecture model. In order to carry out the computations required during the simulation, these dependencies are formulated in the form of a directed graph, denoted $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. $\mathcal{V}$ designates the set of nodes of $\mathcal{G}$, formed by the transition instants $x_i(k)$ of the network. $\mathcal{E}$ denotes the set of arcs formed by pairs $(x_i(k), x_j(k'))$, where



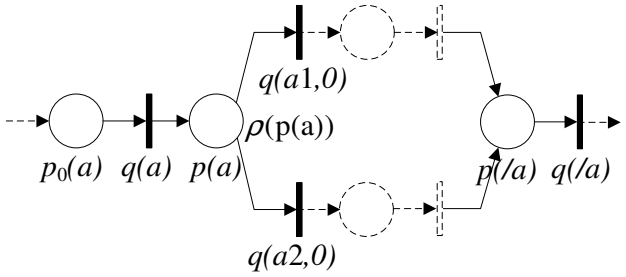
(a) Pattern related to the computation statement, $q(Op)$ and $q(/Op)$ denote the start and end transitions. T_{Op} denotes the computation duration.



(b) Pattern related to the finite iteration statement, $q(f,0)$ and $q(/f)$ denote the start and end transitions of one iteration, $q(le)$ denotes the end of all the iterations.



(c) Pattern related to the infinite iteration statement, $q(i,0)$ and $q(/i)$ denote the start and end transitions of one iteration.



(d) Pattern related to the alternative instruction, $q(a)$ and $q(/a)$ denote the start and end transitions of the alternative, ρ is the used to designate the selected transition.

Figure 2.6: Examples of elementary patterns used to describe the dependencies between start and end instants of computation and communication statements.

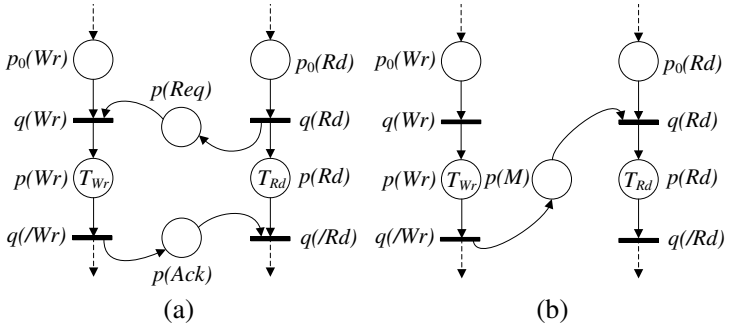


Figure 2.7: Examples of patterns related to communication statements: (a) rendez-vous protocole, (b) infinite capacity FIFO protocol. T_{Wr} and T_{Rd} denote the write and read durations.

$x_i(k)$ denotes the source node and $x_j(k')$ the destination node. These arcs are established when a dependency exists between times $x_i(k)$ and $x_j(k')$. The weight associated with the arc $(x_i(k), x_j(k'))$ corresponds to the duration between the instants $x_i(k)$ and $x_j(k')$. In the proposed approach, this graph will be used in simulation in order to carry out the calculations previously expressed within the state equations (2.1). The traversal of the graph reflects the execution of the Petri net given the evolution rules used. A detailed example of such a graph is presented in [J2-2].

2.5 Generic execution model

The form taken by the executable model created according to the proposed simulation approach is discussed here. This is the simulated model that evolves according to the principles illustrated in the figures 2.3 and 2.4. This model has an equivalent behavior from the point of view of its inputs and outputs to an executable model created using a traditional approach but with a limited number of required processes. By applying this principle to the example in figure 2.1, the executable model created must therefore exhibit equivalent behavior from the point of view of the $M1$ input and from the point of view of the $M6$ output. Part (a) of the figure 2.9 illustrates the organization and behavior of the execution model considered. The instants of synchronization between functions are themselves expressed within the executable model according to the

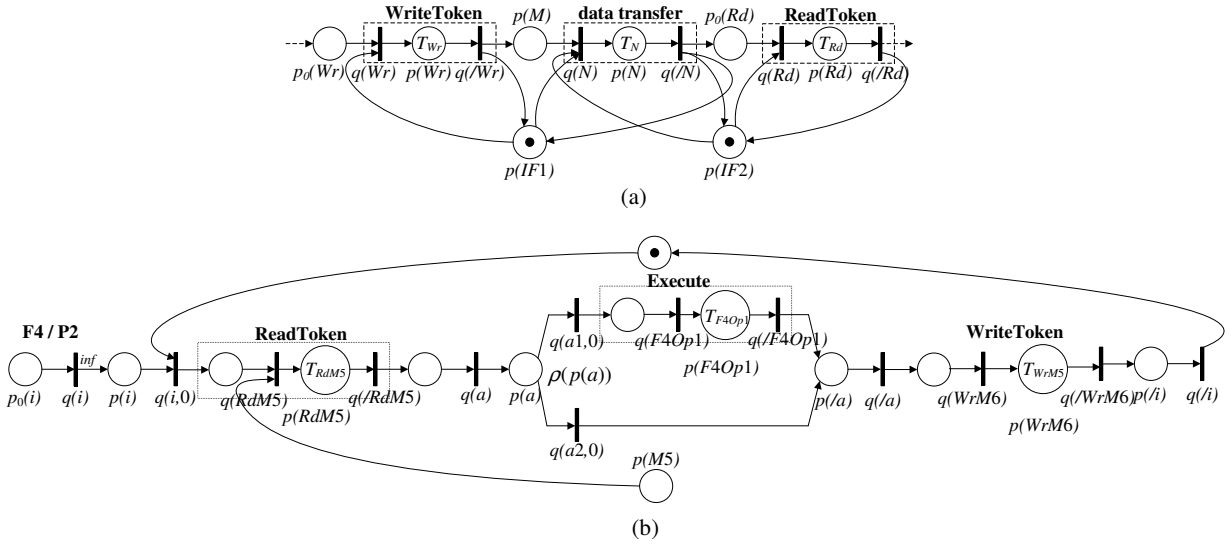
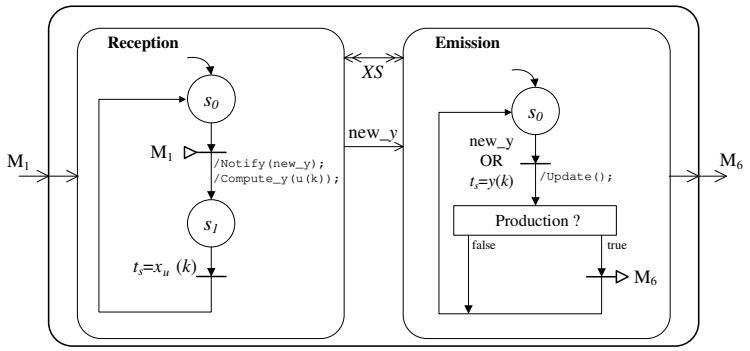


Figure 2.8: Examples of built Petri net models to describe the dependencies between start and end instants of communication and computation statements.

approach previously presented.



(a)

```
1: procedure UPDATE(Def XS XS)
2:   if new_y then //New output instant set
3:    y(k)=ReadFirst_y(XS);
4:    Production=false;
5:   else //Simulation time has reached y(k)
6:    if CorrectionNeeded(XS) then
7:      Correct_y(XS);
8:      Production=false;
9:    else
10:     Production=true;
11:     Release(XS);
12:     y(k)=ReadFirst_y(XS);
13:   end if
14: end if
15: end procedure
```

(b)

Figure 2.9: (a) Structural and behavioral description of the proposed generic execution model, (b) management procedure of the computed instants.

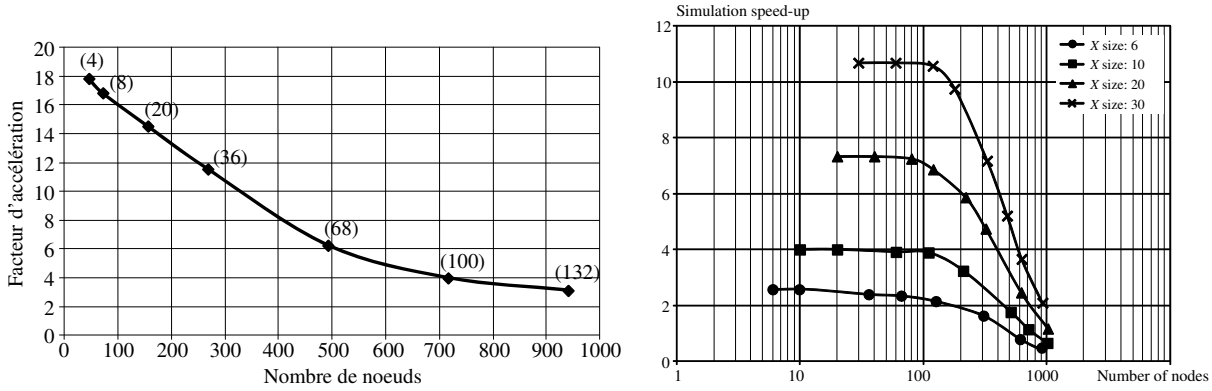
According to the behaviour thus described, a new input data can be consumed when the instant $x_u(k)$ is reached. The action noted `Compute_y` corresponds to the calculation of the instants $X(k)$ and $y(k)$ taking into account the instant of reception $u(k)$. XS designates the set of instants X calculated and not reached during the simulation. The process noted Emission is activated according to two conditions: each time a new output time is calculated (new_y) or when the simulation time has reached the calculated time ($t_s = y(k)$). The presented model is said generic as it can be applied for different state equations and therefore different architecture models (the case presented here is with one input and one output).

Part (b) of the figure 2.9 presents the sequence of operations performed by the procedure denoted `Update`. This sequence of operations and the behavior of the executable model are illustrated in part (b) of the figure 2.4. When a new output time has been computed, the next time (line 3) is set to $y(k)$ and no output is performed. When the simulation time t_s has reached the value $y(k)$, XS is analyzed to determine if a contention has occurred (action `CorrectionNeeded` line 6). In the example of figure 2.4, the action `CorrectionNeeded` corresponds to the comparison of $x_{NM_1E}(k')$ with $x_{NM_4S}(k)$ and $x_{NM_4E}(k)$. The operation `Correct_y` (line 7) designates the action which performs the correction of the instants $X(k)$ from the updated instants. In the example of the figure 2.4, $x_{NM_4S}(k)$ is updated with the value $x_{NM_1E}(k')$ and the times depending on it are also updated. $y_c(k)$ corresponds to the corrected value of $y(k)$ taking into

account the contention to access resource N. According to the behavior presented, the instants stored within XS can therefore be corrected retroactively at the instant $t_s = y(k)$. When the simulation time has reached the value $y(k)$ and no correction is necessary, the instants stored within XS can be deleted (line 11).

2.6 Implementation and preliminary results

The approach presented was implemented to quantify the gains obtained in terms of simulation duration. The objective was also to estimate the potential limits of the approach given the complexity of the calculations to be made during the simulation. Two modelling and simulation environments were considered. The first is the Intel CoFluent Studio [27] modeling and simulation environment. This environment allows the capture of physical architecture models by combining functional and executive views. Once the graphical description made, the models are generated in SystemC in order to be simulated and analyzed. The models proposed by this environment are very rich and make it possible to approach numerous mechanisms of the functioning of hardware-software architectures. We have therefore established different patterns required for the construction of state-based models according to the construction approach previously presented. A description of the generic execution model in the formalism specific to this environment has been developed. It was thus possible to compare different models created with or without the proposed approach and thus to estimate the precision obtained and the simulation times. The figure 2.10 presents two sets of results obtained in order to illustrate the potential gains of the proposed approach as well as the influence of the complexity of the approach.



(a) Experiment results from [J2-2]

(b) Experiment results from [C2-1]

Figure 2.10: Influence of the complexity of the calculation method on the achieved simulation speed-up.

Part (a) of the figure, taken from [J2-2], illustrates the effect of applying the proposed method on a variable number of functions composing a physical architecture model similar to the one illustrated in figure 2.1. The number of functions considered is indicated in parentheses. In doing so, the gains observed in terms of simulation speed change by a factor of 3 to 18, while preserving the accuracy of the models. We observe the significant influence of the complexity of the calculation method on the accelerations obtained. Thus, for a set of 100 abstract functions by applying the proposed method, the gain obtained is a factor of 4 for a graph $\mathcal{G}$ composed of more than 700 nodes. Part (b) of the figure, taken from [C2-1], also illustrates this phenomenon by keeping a constant number of abstracted functions but increasing the size of the graph required to establish an equivalent accuracy.

In [J2-1], we quantified the contribution of the correction method mentioned in the section 2.3. To do this, the simultaneous execution of two applications is considered on a platform having a shared communication bus. A first application is described with a variable number of functions depending on the configurations, periodic input data and randomly generated calculation and communication times. The second application is described in order to cause contention on the communication bus and this according to a controllable contention rate. A first come first served (*first come first served*, FCFS) bus arbitration is considered. These models were created within the Intel CoFluent Studio environment. The table 2.1 presents the results obtained for different configurations studied. The accuracy of the models was evaluated by comparing the instants of synchronization between the functions of the applications with an execution of the models not using the

proposed approach. The application of the proposed correction method allows a complete correction of the errors introduced by the second concurrently executed application. The acceleration factors for the simulation of the two concurrent functions and of the single reference application were also estimated.

Configurations	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**
Number of functions in the reference application (A)	4	4	8	8	16	16	32	32
Size of the state vector X^A	24	24	59	59	120	120	248	248
Occupation rate of the bus (%)	24	28	11.2	13.5	6.4	8.5	1.1	1.7
Error rate with correction (%)	0	0	0	0	0	0	0	0
Error rate without correction (%)	3.1	11.4	1.5	4.4	2.1	2.16	0.35	0.96
System model simulation speed-up	3.8	2.2	3.7	2.2	2.2	1.57	3	1.57
Reference application model simulation speed-up	14.95		24.4		13.65		4.52	

Table 2.1: Tested configurations and associated simulation results [J2-1].

The second implementation of the proposed approach corresponds to a SystemC development of the generic execution model. This development was used in the thesis of Mr. Hai Dang Vu and will be presented in chapter 3. More generally, we have been able to put this approach into practice through various case studies carried out as part of co-supervised theses. The table 2.2 summarizes the case studies discussed as well as the gains obtained in terms of simulation speed. These case studies will be presented in the indicated chapters. In particular, the approach presented was used to improve the efficiency of existing models (study of the HomePlugAV communication interfaces and the LTE mobile radio receiver presented respectively in chapters 3 and **??**). It was also used to promote the creation of effective models based on a detailed understanding of the mechanisms observed on a real target (study of the AXI4-Lite multiprocessor bus presented in chapter 3).

Case study	**Related PhD thesis**	**Simulation speed-up**
LTE radiomobile receiver	A. Barreteau (chapter 4)	4
AXI4-Lite multiprocessor bus	H.D. Vu (chapter 3)	Between 2.5 and 1700
HomePlugAV communication	T. Majdoub (chapter 3)	200
H263 decoder	D. Singh [J2-1]	Between 2 and 14

Table 2.2: Summary of the case studies using the proposed modelling and simulation approach. These case studies are presented in the indicated chapters. The indicated speed-up factors are obtained without degrading the accuracy of the predictions.

2.7 Conclusion

The approach presented in this chapter combines simulation methods and formal models in order to optimize the efficiency of simulable models used for the performance evaluation of hardware and software architectures of embedded systems. The creation of simulable models using the proposed approach turns out to be systematic for the creation of state model and executable model. A first prototype of a software tool was developed using Acceleo technology for the manipulation and transformation of models [44]. This prototype tool aimed to validate the principle of automated creation of simulable models using the proposed approach, and this from models captured in the Intel CoFluent Studio environment. In parallel with the continuation of this development, it would be beneficial to approach the reduction of the complexity of the formal model created, this in order to be able to consider systems comprising a high number of components. Finally, it would be interesting of establishing suitable patterns in order to describe the main strategies of arbitration of shared resources existing within multiprocessor architectures.

From an organizational point of view, the approach presented could be developed in particular within the framework of the research stay carried out in 2012-2013. It was also possible to apply this work to various case studies carried out within the framework of co-supervised theses. These applications are illustrated in the following chapters 3 and 4.

2.8 Supervision

Master thesis

- Dharmender Singh, *A dynamic correction method for fast yet accurate simulation of multiprocessor systems*, University of Nantes, June 2018.

2.9 Publications related to chapter 2

International journals with program committee

[J2-1] S. Le Nours and D. SINGH, "A Generic Executable Model for Fast Yet Accurate Contention Simulation in Multiprocessor Systems," *IEEE Embedded Systems Letters*, vol. 12, no. 4, pp. 117–120, 2020. DOI: 10.1109/LES.2020.2966801. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02436682.

[J2-2] S. Le Nours and A. Postula, "A Hybrid Simulation Approach for Fast and Accurate Timing Analysis of Multi-Processor Platforms Considering Communication Resources Conflicts," *Journal of Signal Processing Systems*, vol. 90, no. 12, pp. 1667–1685, Dec. 2018. DOI: 10.1007/s11265-017-1315-x. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01643250.

International conferences with program committee

[C2-1] S. Le Nours, A. Postula, and N. Bergmann, "A dynamic computation method for fast and accurate performance evaluation of multi-core architectures," in *Design, Automation & Test in Europe*, Dresde, Germany, Mar. 2014, DATE 2014. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00906661.

[C2-2] S. Le Nours, "Timing correction technique for fast and accurate state-based performance models," in *Forum on specification & Design Languages*, ser. FDL 2015, Barcelone, Spain, Sep. 2015, paper#54. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01178416.

[C2-3] S. Le Nours, A. Barreteau, and O. Pasquier, "A state-based modeling approach for fast performance evaluation of embedded system architectures," in *IEEE International Symposium on Rapid System Prototyping*, Karlsruhe, Germany, May 2011, p. 2011. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00795240.

Chapter 3

Communication modelling of multiprocessor architectures

3.1 Introduction

This chapter presents the work carried out around two case studies considered in the context of two co-supervised theses. These works have in common to address the design of architectures involving several processors interconnected through communication buses of different natures. In both cases, the work focused on the prediction of temporal properties (typically, the execution latency of applications) with the objective to optimize the studied architectures according to parameters such as the number of processors or the memory capacity required. The creation of simulable models was then necessary and the improvement of the efficiency of these models was an essential point in order to favour the dimensioning of the architectures. Thus, in collaboration with the two co-supervised students, we were able to evaluate the application of the simulation approach presented in Chapter 2 and compare its contribution to models created according to other approaches. Moreover, we considered the implementation of this approach according to two development environments: Intel Cofluent Studio and SystemC.

For the two discussed case studies, it was essential to correctly describe the communication mechanisms between processors. As mentioned in Chapter 1, the analysis of the temporal properties of embedded systems can be carried out according to different levels of detail. Figure 3.1 illustrates the different levels of abstraction addressed in the context of these studies given the description of the communication mechanisms.

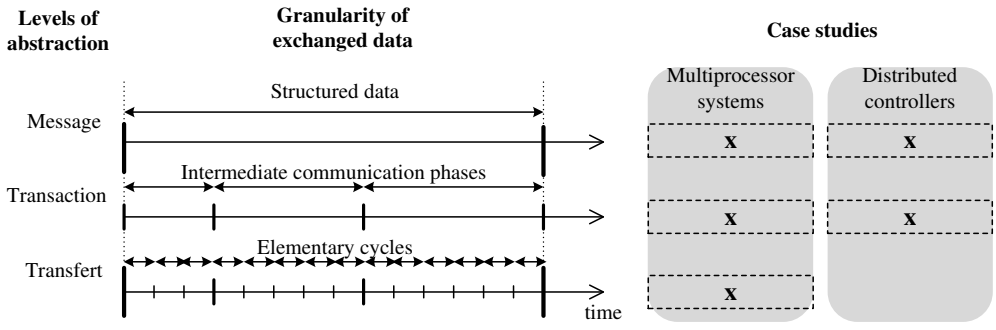


Figure 3.1: Illustration of levels of abstraction in communication description for the two case studies.

As illustrated, the *message* level considers the exchange of structured data through atomic primitives (*e.g.*, *WriteTokens* for writing a set of tokens). The communication time is not zero and, within the framework of our studies, it will be calculated taking into account the simulation approach presented in Chapter 2. The *transaction* level describes the successive exchanges of refined data by identifying certain instants representative of the communications. The arbitration of exchanges is explicitly described. The *transfer* level considers elementary data exchanges according to a defined protocol and for which the time scale is the clock

cycle (*cycle-accurate).* A similar classification was adopted by G. Schirner et al. in [45]. They present a quantitative evaluation of the effect of these different levels of abstraction on the accuracy of performance models and their simulation durations. This evaluation was made by considering different communication protocols: the ARM AHB system-on-chip multiprocessor bus protocol, the CAN fieldbus protocol and the Motorola ColdFire multiprocessor bus protocol.

In the first part of this chapter, the case of an architecture envisaged for the automotive field is presented. This architecture was studied as part of the ANR CIFAER project and the thesis of Mr. Takieddine Majdoub [46] conducted from 2009 to 2012. This thesis approached the dimensioning of a distributed architecture with several electronic controllers communicating through a protocol adapted to power line carrier transmission. In the scope of this work, we were able to evaluate the contribution of the proposed simulation method on this case study. This study was conducted before we considered the different principles mentioned in Chapter 2.

In the second part of this chapter, we present an architecture integrated on a chip with several processors communicating by shared memory through a multi-master communication bus. This case study was addressed as part of Mr. Hai-Dang Vu's thesis [47] conducted from 2017 to 2021. This thesis was oriented towards the application of probabilistic models and methods for the study of multiprocessor systems. It also offered the opportunity to optimize the communication bus model used for this study. We were thus able to compare the models created with measurements on a real target, making possible an advanced comparison of the precision of the built models. Only aspects related to the modelling of communication buses with the proposed method are presented in this chapter. The other facets of the work carried out within the framework of Mr. Hai-Dang Vu's thesis will be presented in Chapter 5.

3.2 Improvement of simulation efficiency for distributed architecture analysis

The work presented in this section focuses on the study of distributed architectures for the automotive field, work begun in 2008 through the Master of Miss Maria Cheik Wafa [C3-1] and continued as part of the thesis of Mr Takieddine Majdoub [46]. It represents a first application of the principles mentioned in Chapter 2 to the case of communication protocols adapted to networks of on-board computers.

3.2.1 Characteristics of the studied systems

This case study was addressed in order to evaluate the benefits of data transmission based on power line carrier (PLC) for high-speed communications within motor vehicles. The studied system, represented in Figure 3.2, corresponds to a distributed architecture of several on-board computers and supporting a video stream application applied to the automotive field.

The application consists of managing a video stream, captured by one of the controllers and then transmitted to other controllers through the HomePlugAV communication protocol. This protocol is adapted to PLC transmissions. It supports the transfer of data with IP and Ethernet formats with throughput of up to 150 Mbit/s. Each controller (*Electronic Control Unit*, ECU) essentially consists of a microprocessor, performing the application tasks and managing data flows according to the RTP-UDP and IP protocols, and a HomePlugAV communication interface. The purpose of the study carried out was to estimate the capacities required of the memories within each communication interface.

As illustrated in the lower part of Figure 3.2, the HomePlugAV protocol supports two communication media access techniques: TDMA (*Time Division Multiple Access*) type access and type CSMA/CA (*Collision Sense Multiple Access/Collision Avoidance*). Considering the needs of the application, only the TDMA access technique was considered in this case study. A detailed presentation of this protocol and the operation of the communication interfaces is given in [46]. In this architecture, the role of the HomePlugAV communication interfaces is to classify the IP packets according to their source and their destination, and to ensure the transmissions are correclty performed. The data exchanged between ECUs correspond to sets of bytes of constant size designated by the term *Physical Block* (PB). These PBs are transmitted in the time intervals allocated for this purpose for each ECU. They are kept within the interfaces until they are positively acknowledged.

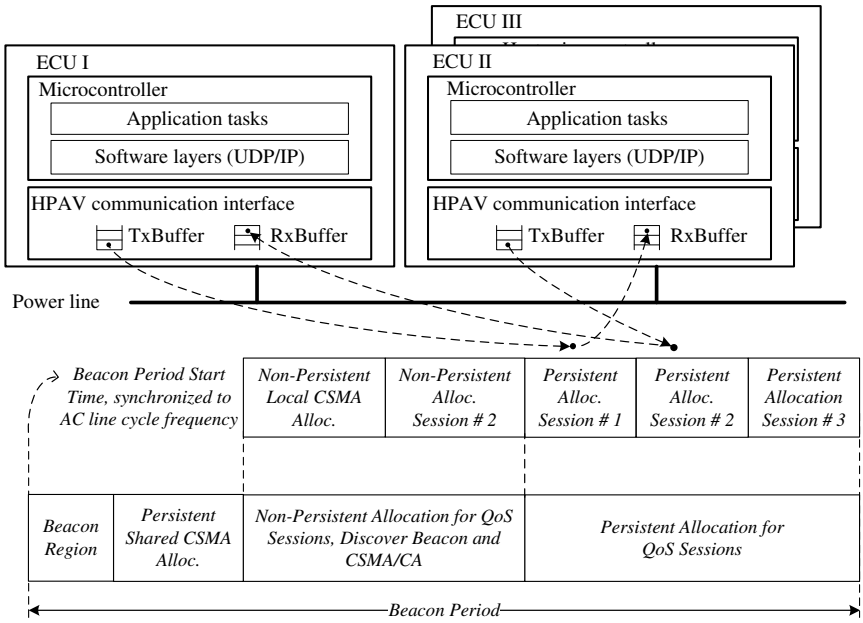


Figure 3.2: Illustration of the distributed architecture studied in the scope of the CIFAER project [J3-1].

In order to illustrate the operation of this protocol, Figure 3.3 represents the typical use of the memory of the transmitting HomePlugAV interface for the transmission of PBs to a receiver.

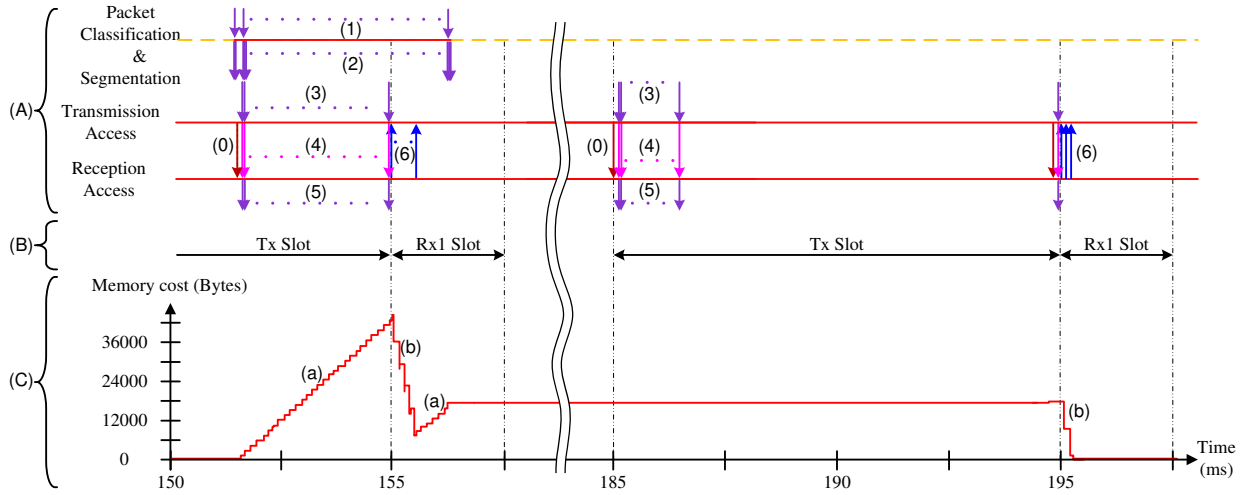


Figure 3.3: Illustration of exchanged between two HomePlugAV interfaces and evolution of the interface memory usage during the transmission [J3-1].

Part (A) represents the various exchanges carried out between the transmitting and receiving interfaces. Part (B) indicates the different time intervals allocated for each element. Part (C) represents the evolution over time of the occupation of the memory within the sender interface. The interval (1) designates the IP packets successively received by the sender interface and intended for the receiver. The interval (2) corresponds to the data successively stored within the interface, inducing the increase in the memory occupation (a). When a PB is formed and the transmitter time slot occurs, the start of a frame is notified (0) then the stored PBs are successively read within the interface (3) and transmitted over the communications medium (4). The reception of the transmitted data corresponds to the interval (5). Acknowledgments are delivered by the receiver when the time interval allocated to it occurs (6). The acknowledged PBs are then deleted from the sender interface, the memory occupation then being reduced (b). The observation given in Figure 3.3 comes from the simulation of the model presented in [46]. This model considered the organization presented in Figure 3.2 with three ECUs. This model was used to predict the memory occupation of the different interfaces

as well as the end-to-end latency during the transmission of video sequences between on-board computers. The influence of parameters such as the error rate on the PBs transmitted or the size of the IP packets used was also evaluated. The main results obtained being are presented in [J3-1].

3.2.2 Application of the simulation method

The model initially created exhibits the different phases of transmission of PBs as well as the acknowledgments made in the time intervals allocated for each ECU. This level of detail, considered here as the transaction level, allows an analysis of the memory occupation within the interfaces during the different communication phases. Such observations can be used in order to appropriately size these resources taking into account the constraints and the parameters of the complete system. However, such a representation leads to a high number of communications between model elements, significantly impacting the simulation duration required. The proposed simulation method was applied in order to reduce the number of exchanges between the interfaces while preserving the prediction made of the use of the memory resources. For this case study, the reduction in the number of exchanges required is obtained by applying the following observations:

- each transmitted message corresponds now to a variable set of PBs resulting from the reception of IP frames. The duration between two successive transmissions is thus calculated during the simulation taking into account the number of transmitted PBs.
- The instants of transmission are also calculated taking into account the TDMA intervals previously established.
- The supposed transmission instants of each PB are indicated within each message, in order to allow a detailed observation of the occupation of memory resources for the message level model.
- The acknowledgment instants of PBs are also calculated locally within the elements of the message level model in order to overcome the exchanges previously displayed in the transaction level model.

Figure 3.4 illustrates the application of the simulation method in the case of the scenario previously illustrated in Figure 3.3. Part (A) of Figure 3.3 illustrates successively the formed IP packets. Part (B) presents the

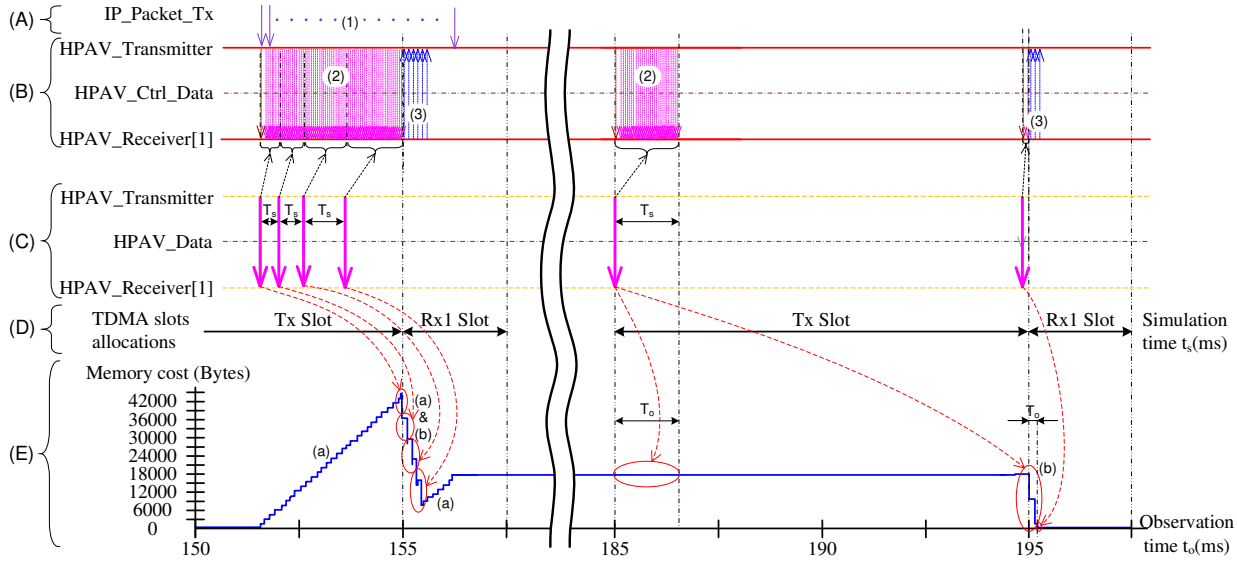


Figure 3.4: Evolution over the simulation time of transaction level and message level models [J3-1].

exchanges of PBs carried out between the sender and the two receivers by considering a transaction level model. Part (C) presents the exchanges performed for the message level model created using the proposed simulation method. The durations denoted T_s are calculated during the simulation, corresponding to the durations between two successive sendings of messages. Part (D) presents the TDMA allocations for the transmitter and the two receivers. Part (E) presents the evolution of the amount of memory allocated in order to store the PBs in the buffer of the transmitting interface.

Number of transmitted images	Transaction level	Message level	Speed-up factor
150	249.4 s	1.38 s	187.8
1500	2498.6 s	12 s	208
4500	7459.3 s	35.9 s	208

Table 3.1: Observed simulation duration (in seconds) for execution of transaction and message level models. The models were executed on a Intel Core2 Dual (2.66 GHz) running under Windows7 [C3-2].

The message level model therefore considers a grouping of PBs within a single data exchange. The number of PBs is variable and depends on the quantity of data present in memory. Also, we used the possibility of predicting the instants of acknowledgment of the PBs, these instants depending on the organization of the access intervals to the communication medium. According to this observation, the instants of releasing the memory can be established at the instant of the production of the PBs. An important point lies in the fact that the observation of the evolution of memory occupation is no longer carried out solely according to the simulation time t_s but taking into account a local variable called observation time t_o. This variable is introduced because the observations are no longer linked solely to the flow of simulation time but are made taking into account the calculations carried out locally within the elements of the model. The models developed were presented in an international journal article [J3-1] and in international conferences [C3-2], [C3-3]. This work was also presented in the reports of the CIFAER project [R3-1], [R3-2].

3.2.3 Obtained results

Experiments were conducted using the Intel CoFluent Studio modeling and simulation environment. The previously mentioned transaction and message level models were described within this environment. These models are created first of all through a graphical capture <ith a notation specific to the environment, supplemented with codes in the C++ language. These sets are then automatically translated into SystemC to be simulated and analyzed. The simulations were carried out in order to study the occupation of storage resources during the transmission of several sets of images, from 150 images to 4500 images, between three computers. Table 3.1 gives the simulation durations observed for the simulation of the different models. For the different situations evaluated, the gains obtained in terms of simulation time lead to a reduced analysis time, offering the possibility for architects to be able to explore different design alternatives more quickly.

3.3 Improvement of simulation efficiency for multiprocessor systems-on-chip analysis

The case study presented here was cosnidered within the framework of the thesis of Mr. Hai-Dang Vu and within the framework of the collaboration with the German institute OFFIS.

3.3.1 Characteristics of the studied systems

The systems studied here correspond to data flow oriented applications distributed on multiprocessor platforms, as illustrated in Figure 3.5. The applications are described on the basis of the *Synchronous Data Flow* (SDF) model of computation, introduced by E. Lee in [48]. They are organized into a set of actors exchanging tokens through communication channels. Subsequently, each exchanged token corresponds to a pixel composing the image processed by the studied application. In the example of Figure 3.5, the illustrated application comprises four actors and five communication channels noted from C_0 to C_4. The communication channels correspond to buffers of FIFO type with limited capacity and for which read and write accesses are blocking. An actor can only begin its computation phase once the reading of the various data from the input channels has been completed. Once the computation phase has been executed, an actor writes the necessary data to its output channels.

The studied systems are based on the use of a platform organized as a *tile-based platform* where each tile corresponds to a programmable processor with its private instruction and data memories as well as a

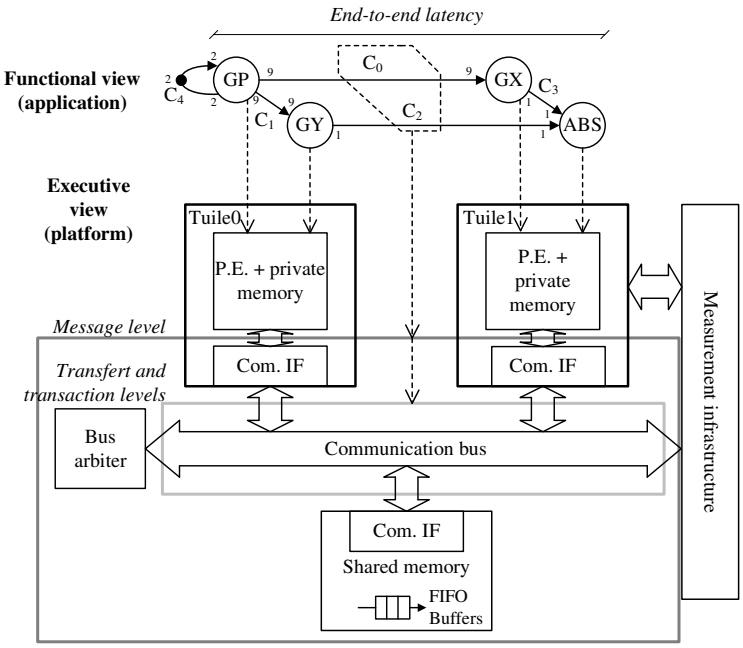


Figure 3.5: Illustration of the studied multiprocessor systems. They are made of an application described following the SDF model of computation and a platform with multiple tiles connected through a communication bus and a shared memory.

communication interface. Such a platform has a variable number of tiles, one or more shared memories and a shared communication bus. Shared memories are used to temporarily store the data exchanged between the actors of the application. In the example in Figure 3.5, the five channels are exchanged through the communication bus and shared memory. It should be noted that each tile can execute its program without interfering with the other tiles as long as this program does not explicitly access the shared resources. The interest of such a system lies in the fact that it can be easily extended in number of tiles.

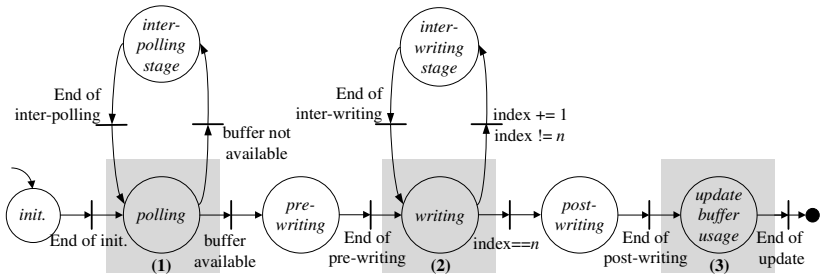
As part of Mr. Hai-Dang Vu's thesis, we were particularly interested in the modelling of communication between tiles in order to describe the influence on the temporal properties of the system (here, the end-to-end latency of the application allocated on different tiles). At the application level, communications between actors are carried out through two functions: *WriteTokens* leads to writing a set of tokens within the shared memory, *ReadTokens* leads to reading a set of tokens within the shared memory. The example of writing *n* tokens is shown in Figure 3.6.

```
void WriteTokens(channel_t *channel,
            token_t inputtokens[])
{
  // poll for empty buffer
  while(*(channel->full));                    (1)

  // write token into channel buffer
  for(int i=0;i<channel->prate;i++)
  {                                           (2)
    Channel->tokens[i]=inputtokens[i];
  }

  //flag buffer as full
  *(channel->full)=1;                         (3)
}
```

(a)



(b)

Figure 3.6: (a) Coding example of the procedure *WriteTokens* for writing *n* tokens, (b) state-transition diagram describing the sequence of writing *n* tokens into the shared memory. The shaded rectangles highlight the states during which the shared resources (communication bus, memories) are accessed.

Part (b) of Figure 3.6 represents the sequence of states observed during the writing of *n* tokens within a shared memory. The states denoted by *polling*, *writing* and *update buffer usage* correspond to situations where processors access shared resources and for which contention may occur. Within the framework of the

experiments carried out, the arbitration of the communication bus is done according to the *First Come First Served* protocol. The states denoted *initialization*, *inter-polling stage*, *pre-writing*, *inter-writing stage* and *post-writing* correspond to situations where the execution of the program of each processor does not interfere with any other resource. At the beginning of the writing of n tokens, the polling phase aims to check the availability of the targeted buffer. The *inter-polling* phase reflects a minimum duration between two successive polling phases. Once the buffer is ready, a copy of the data of a token is written (state *writing*) and at the end of the writing of the data the state of the buffer is updated in order to notify the possibility to read the written data (corresponding to the elementary state *update buffer usage*). The procedure for reading n tokens is carried out according to a similar sequence of states. Each state has its own duration expressed in number of clock cycles. These elementary durations will be established experimentally by measurements on a real target via the measurement infrastructure mentioned in Figure 3.5. This infrastructure is also used to allow the computation time of each actor to be measured as well as the execution latency of the application on the platform.

Ultimately, the modelling and simulation of such a system should make it possible to estimate the execution latency of the application for different design alternatives, in particular taking into account the number of tiles used and the distribution of the actors of the application.

3.3.2 Application of the simulation method

The analysis of the temporal properties of the studied systems requires taking into account the effect of communication mechanisms and the penalties induced during simultaneous accesses to shared resources. These mechanisms have been successively described at the transfer, transaction and message levels of abstraction. Figure 3.7 considers the case of two simultaneous communications, a write and a read of n tokens, through the bus and shared memory.

Part (a) of the figure corresponds to a transfer level model built in order to reproduce the operation illustrated in Figure 3.6. At this level of detail, the phase of scanning the state of the buffer as well as the successive exchanges token by token are expressed. Each of the elementary phases is exhibited through a specific duration[1]. The situations noted (1), (2) and (3) correspond to the different situations of access to shared resources previously highlighted in Figure 3.6. The situation noted (1) corresponds to conflicts due to two simultaneous scanning phases. The situation noted (2) corresponds to the situation where an elementary write phase is delayed due to an elementary read in progress. Finally, the situation noted (3) corresponds to the situation where the phase of updating the state of the channel C_1 is delayed due to the updating of the channel C_0.

At this level of detail, two phases involve numerous calls to the simulation engine: the scanning phase and the successive writes or reads for each token. A message-level model using the simulation principles introduced in Chapter 2 has been built in order to limit the number of calls required while preserving precision on the use of resources, and therefore the effect of contentions on access to shared resources[2]. Using the proposed approach, the communications are described by considering as atomic the transfer of n tokens, the duration of the communications then being computed during the simulation. For this level of abstraction, bus arbitration is no longer explicitly expressed, it will be taken into account in the computation performed. Part (b) of Figure 3.7 illustrates the application of the proposed simulation method for a message-level model of communications. At this level of abstraction, only three significant instants are exhibited: the instants of the start and end of communication of the tokens (x^w_{ComC} and x^w_{ComE}), and the instant when the buffer is available (x^w_{ComS}).

The application of the proposed simulation method leads to the calculation of the communications durations, noted T_W and T_R Figure 3.7. The calculations of the durations of the *WriteTokens* and *ReadTokens* functions are performed at instants $x^w_{ComS}(k)$ and $x^r_{ComS}(k')$ and are denoted `ComputeCommTime`. It is

[1] The simulation instants noted with an upper index w are associated with the writing process, those noted with an upper index r are associated with the process of reading. Each instant noted $x^w_j(k)$ (respectively, $x^r_j(k)$) corresponds to the k-th instant of transition between two elementary phases of the token writing process (respectively, of reading tokens). [2] A transaction level model was also considered in order to replace the polling phase by waiting for an event notifying the change of state of the buffer. The different writings and successive readings were always exhibited.

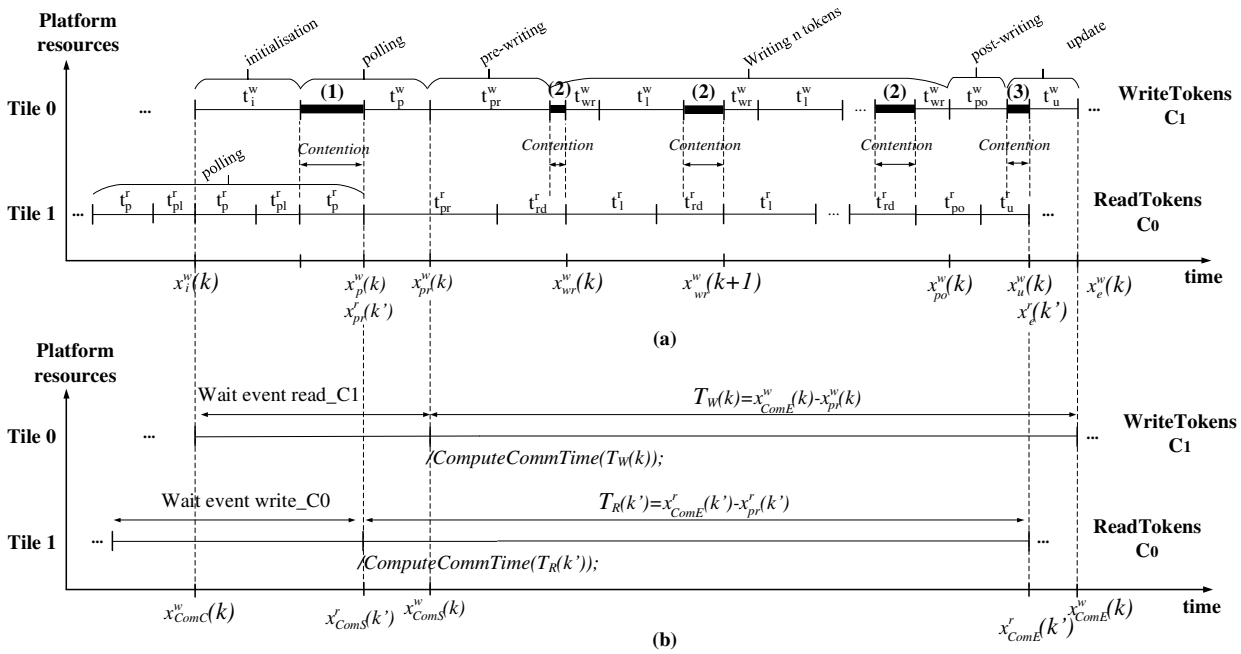


Figure 3.7: Execution of models describing the writing of n tokens on channel C_1 and the reading of n tokens from channel C_0 in the example of Figure 3.5. (a) At the transfer level, the basic delays and penalties caused by contentions are highlighted. (b) At the message level, the communication times (denoted T_W and T_R) are calculated during the simulation.

therefore necessary to establish the durations of the communication phases even in the case where situations of conflict of access to shared resources arise. According to the proposed approach, the expression of communication durations is based on the knowledge of the arbitration protocol used as well as on the mechanisms of exchange between actors (FIFO with limited capacity in the studied systems). This knowledge makes it possible to establish the dependencies between instants of communication. Figure 3.8 illustrates the modelling made in the formalism of timed Petri nets of two simultaneous communications via the bus and the shared memory studied. In Figure 3.8, the place denoted p_{shared} describes the limited availability of

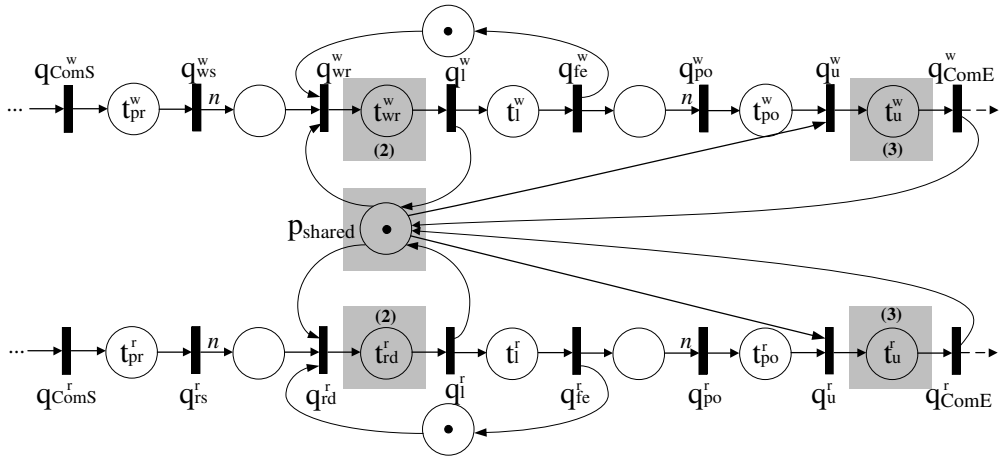


Figure 3.8: Timed Petri net for writing n tokens on channel C_1 and reading n tokens from channel C_0. Communication is considered at transaction level with the FCFS arbitration policy. The shaded rectangles correspond to the situation during which the shared resources (bus, memory) are accessed.

shared resources. Based on this model, transition times can be expressed using the addition and maximization operators. The complete state model, extended to several tiles, is presented in [47] and [C3-4]. This model is

used in order to calculate during simulation the durations of writing n tokens and this taking into account the number of communications observed.

The description in the SystemC language of the message level communication model is illustrated in Figure 3.9. Figure 3.9 represents the message-level model organization of a channel communication between

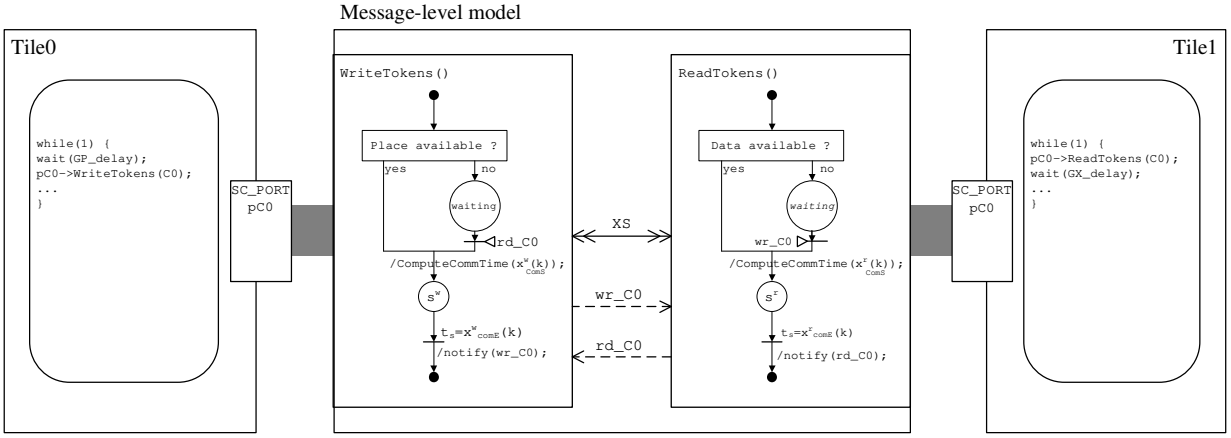


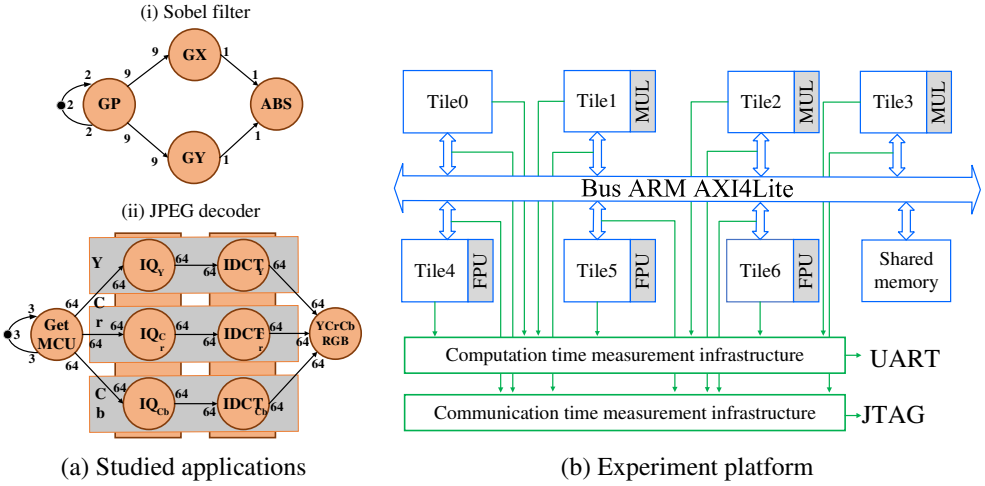
Figure 3.9: Illustration of the organization of the message level communication model described in the SystemC language. This model uses the calculation method presented in Chapter 2.

two tiles. The communication infrastructure is then described using the generic execution model introduced in Chapter 2. This is transcribed in the form of a hierarchical channel for which the interfaces comprise the methods of access to the communication resources. The behaviour of the methods describes the access conditions for the exchange of finite capacity FIFO type channels as well as the required durations. The `ComputeCommTime` establish during the simulation the durations of the writes and reads of n successive tokens based on the state model established from the network illustrated in the figure 3.8. In the case of multiple channels exchanged between tiles, the instants in XS are shared among the interfaces related to each channel. The calculated instants can thus be updated depending on the observed communication situations. This description is detailed in [47].

3.3.3 Obtained results

The evaluation of the developed models was done through two distinct applications studied according to different allocations on a platform based on seven tiles. The considered applications correspond to a Sobel filter and a JPEG decoder. The modeling according to the SDF formalism of these applications is given in part (a) of Figure 3.10. The multiprocessor platform studied is described in part (b) of this figure. Seven tiles are connected to a shared memory through an AMBA AXI4-Lite [49] type communication bus. Each tile corresponds to a Xilinx MicroBlaze core associated with private instruction and data memories. The measurement infrastructure put in place makes it possible to record the computation times of each actor allocated to a tile as well as the communication times between actors. This set-up was implemented on a Xilinx ZC702 prototyping board based on a Xilinx Zynq-7000 FPGA. Different allocations of the applications studied on the experimentation platform were considered. Each implementation made it possible to record the execution time of each application and could thus be confronted with the model created elsewhere. Also, preliminary measurements on the target made it possible to establish the durations of the elementary communication phases, these quantities then being used within the transfer, transaction and message level models.

In Table 3.2, the average latency values obtained for different allocations of applications on the considered platform are presented. For example, the experiments denoted *Sobel2* and *Jpeg3* correspond respectively to Sobel filter and JPEG decoder applications allocated on 2 tiles and 3 tiles. The *Measurement* column presents the average execution times for 1 000 000 of iterations of the applications implemented on the experimentation platform. The next two columns give the mean values predicted by the transfer-level and message-level models previously presented. The results of the transfer level model show an over-estimation of 3.57 %



(a) Studied applications (b) Experiment platform

Figure 3.10: (a) Modeling of the studied applications: Sobel filter and JPEG decoder. (b) Experiment platform formed by seven tiles, a shared memory and a communication bus.

Experiment	Measurement	Transfer level	Message level
Sobel1	3690	3821.6 (3.57 %)	3756.6 (1.8 %)
Sobel2	2902.5	2960.6 (2 %)	2936.6 (1.17 %)
Sobel4	3097.4	2859.6 (-7.7 %)	3153.6 (1.81 %)
Jpeg1	2385860	2387757.6 (0.08 %)	2387489.6 (0.07 %)
Jpeg3	940836	940174.7 (-0.07 %)	941445.3 (0.07 %)
Jpeg7	941059	925969.7 (-1.61 %)	941518.1 (0.05 %)

Table 3.2: Comparison of average execution durations obtained by measurements on a real taget and by simulation of transfer and message level models. The values are expressed in clock cycles.

for the configuration *Sobel1* and 2 % for the configuration *Sobel2* while the configuration *Sobel4* presents an underestimation of 7.7 %. In the case of the JPEG decoder, the errors observed turn out to be very low due to the limited influence of the communication on the total execution time of the application. In all the configurations studied, the proposed message level model presents a high level of accuracy. Thus, in the case of the *Sobel4* configuration, for which communications represent an important part, the observed error is 1.81 %. In [47], we also compared the distribution of measured execution times and estimated from the proposed message level model. The shapes of the observed distributions remain identical between measured values and estimated values, regardless of the considered allocations.

Table 3.3 presents the durations measured for the execution of 1 000 000 of application iterations on the experiment platform. Simulation times for 1 000 000 iterations are given for transfer and message level models. In the case of the JPEG application, the simulation time varies from 30 s for the configuration *Jpeg1* to 1 h 55 min for the *Jpeg7* configuration. In the case of the message level model, the simulation time remains less than 10 s for the two applications studied. The acceleration factor of the message level model compared to the transfer level model is 602.25 for the *Jpeg3* configuration and 1731.25 for the *Jpeg7 configuration* .

Also, the interest of such performance models is illustrated here by the comparison between the *Jpeg3* and *Jpeg7* configurations. Thus, the gain obtained in terms of execution time does not turn out to be significant by deploying the JPEG decoder application on 7 tiles, compared to the solution on 3 tiles. Moreover, this estimation turns out to be much faster than considering executions on a real target, fully justifying the usefulness of the message-level models created.

3.4 Conclusion

The work presented here allowed the application of the proposed simulation method on realistic case studies. It was thus possible to confirm the potential gains in terms of simulation time as well as to verify the achieved

Expérience	Mesure	Modèle transfert	Modèle message	Accélération
Sobel1	0:07:23	0:00:08	0:00:03	2.67x
Sobel2	0:07:03	0:00:11	0:00:03	3.67x
Sobel4	0:07:13	0:00:13	0:00:02	6.5x
Jpeg1	13:14:31	0:00:30	0:00:06	5x
Jpeg3	5:12:58	0:40:09	0:00:04	602.25x
Jpeg7	5:13:02	1:55:25	0:00:04	1731.25x
Intel® Xeon® Broadwell-EP CPU E5-2630 v4 (2.20 GHz) https://ccipl.univ-nantes.fr. Each simulation is split into 20 processes, each process is allocated to a processor.				

Table 3.3: Execution durations on real target and simulation durations (in HH:MM:SS) for 1 000 000 of application iterations according to different possible allocations.

levels of precision. The case study addressed with Mr. Majdoub made it possible to quantify the potential contribution of the imagined simulation approach. This was addressed before proposing a generalization of the approach and its principles. In the case of the work carried out with Mr. Hai-Dang Vu, the model developed was supplemented by annotations from measurements on a real target, resulting in a high-precision model. This approach should be applied to advanced bus and network-on-chip communication mechanisms typical of high data flow architectures. One thinks in particular of the use of DMA for systematic transfers by data packets between memory zones. For the two presented case studies the models were developed manually and progressively. Eventually, it would be appropriate to propose an automation of the creation of such models in order to promote their use in the modeling and analysis process.

From an organizational point of view, Mr. Majdoub's thesis addressed the dimensioning of a distributed architecture for a new type of protocol applied to the automotive field. This case study was therefore used to experiment with the proposed modelling approach. As part of this thesis and the associated CIFAER project, we were unable to compare the results of the models developed with concrete implementations, which was a limit of the work then carried out. The proposed optimization of the simulation of the considered case study was a contribution of this thesis. As part of Mr. Hai-Dang Vu's thesis, the use of the simulation method was an opportunity that emerged during the work carried out. This study was processed in a few weeks as the principle of the model associated with the ARM AXI4-Lite bus quickly emerged from our discussions on the operation of this bus. This study was thus able to enrich Mr. Vu's thesis work as well as the collaboration carried out elsewhere with the OFFIS institute.

3.5 Supervision

PhD thesis

- Hai-Dang Vu, *Fast and accurate performance models for probabilistic timing analysis of SDFGs on MPSoCs*, University of Nantes, March 2021.
- Takieddine Majdoub, *Transaction-level modelling methods for simulation efficiency improvement for performance prediction of automotive architectures*, University of Nantes, October 2012.

Master thesis

- Maria Cheik Wafa, *Transaction level modelling of a FlexRay communication network*, Juin 2008.

3.6 Publications related to Chapter 3

International journals with program committee

[J3-1] T. Majdoub, S. Le Nours, O. Pasquier, and F. Nouvel, "Performance evaluation of an automotive distributed architecture based on a high speed power line communication protocol using a transaction level modeling

approach," *Journal of Real-Time Image Processing*, ISSN 1861–8200, 2013. DOI: 10.1007/s11554-013-0323-8. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00777350.

International conferences with program committee

[C3-1] M. Cheik Wafa, S. Le Nours, O. Pasquier, and J. Calvez, "Transaction level modeling of a flexray communication network," in *Proc. on specification and Design Languages (FDL'09)*, 2009.

[C3-2] T. Majdoub, S. Le Nours, O. Pasquier, and F. Nouvel, "Application of temporal decoupling to the creation of efficient performance models of automotive architectures," in *DASIP'2012*, Karlsruhe, Germany, Oct. 2012, pp. -. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00718414.

[C3-3] ——, "Transaction level modeling of a networked embedded system based on a power line communication protocol," in *14th Euromicro Conference on Digital System Design*, Oulu, Finland, Aug. 2011, p. 2011. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00795261.

[C3-4] H.-D. Vu, S. Le Nours, S. Pillement, R. Stemmer, and K. Grüttner, "A Fast Yet Accurate Message-level Communication Bus Model for Timing Prediction of SDFGs on MPSoC," in *Asia and South Pacific Design Automation Conference_ASP-DAC 2021 (Virtual Conference)*, ser. ASP-DAC 2021, Tokyo, Japan, Jan. 2021, p. 1183. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02938566.

Research report

[R3-1] S. Le Nours, O. Pasquier, and T. Majdoub, "Modeling of the reconfigurable architecture and the communication network," IETR, Deliverable of the ANR CIFAER project, 2012.

[R3-2] ——, "Summary of the developed techniques for the modelling of the reconfigurable system," IETR, Deliverable of the ANR CIFAER project, 2012.

Chapter 4

Modelling and management of computation resources in multiprocessor systems

4.1 Introduction

This chapter presents the contributions made within the framework of two co-supervised theses. These works have in common to address the dimensioning under time and energy constraints of the computing resources of multiprocessor architectures processing large data streams (physical layers of radiocommunication systems, audio and video applications). First, the contributions made aim to favour the modelling, evaluation and performance optimization of the studied multiprocessor architectures. Also, this work addresses the definition of online management strategies for computation resources under functional (quality of service) and non-functional (timing and energy) constraints.

The first part of this chapter presents the work carried out as part of Mr. Anthony Barreteau's thesis on the study and optimization under timing constraints of mobile radiocommunication system architectures. Firstly, this work led to proposals for integration of non-functional properties within hardware-software architecture models. This work was carried out from 2007 to 2010 and the proposals made formed the beginnings of the approach previously presented in chapter 2. Secondly, this work also addressed system-level modelling of the dynamic management of computation resources. This work was preceded by Master theses by Mr. Romain Guignard in 2006 and Anthony Barreteau in 2007 on high-level modeling of SoC architectures.

The second part of this chapter presents the work carried out within the framework of Ms. Simei Yang's thesis from 2016 to 2020, related to the proposal of online management methods for computation resources of multiprocessor systems. The purpose of this thesis was to optimize the energy consumed by the computation resources of such architectures. The contributions focused on the definition of scheduling and allocation strategies for multiprocessor and multicore platforms. As part of this thesis, we were also able to contribute to the definition of a method for simulating the online management of computation resources.

4.2 Modelling and sizing of mobile radiocommunication system architectures

The work carried out within the framework of Mr. Barreteau's thesis [50] aimed to apply the principles of system-level modelling for the study and dimensioning of mobile radiocommunication system architectures. Considering the characteristics of such architectures, it was a question of identifying possible limits to the existing modelling flows. Two contributions was made: the first concentrates on the description and simulation of performance models and the second considers the online management of architectures under quality of service constraints.

4.2.1 Modelling and simulation method for non-functional properties

Mobile radiocommunication systems require high computing capacities in order to satisfy the timing constraints associated with the communication protocols supported. These computations are notably due by the various functions of the physical layer of the communication systems. These functions operate successively between the analog/digital conversion stage and the preparation of the data for the higher level communciation protocol layers. One of the objectives of Mr. Barreteau's thesis was to promote the evaluation of computing and memory capacity requirements for such architectures.

As mentioned in Chapter 1, the creation of workload models of an application is done by first describing the structure of the application and the behaviour associated with each of its elements. In a second step, the allocation of the elements of the application on the resources of a given platform is described. The application is organized as a set of functions and coupling relationships between functions. The behaviour of each function expresses the sequence of computation and communication actions performed. A communication or computation load can be associated with each action and reflects, for example, the number of arithmetic operations performed or the memory size required. The evolution of the model thus formed can be analyzed with regard to the degree of potential parallelism of the application and the capacity offered by the communication and computation resources of the execution platform. Figure 4.1 illustrates such a model on the case of the example introduced previously in Chapter 2.

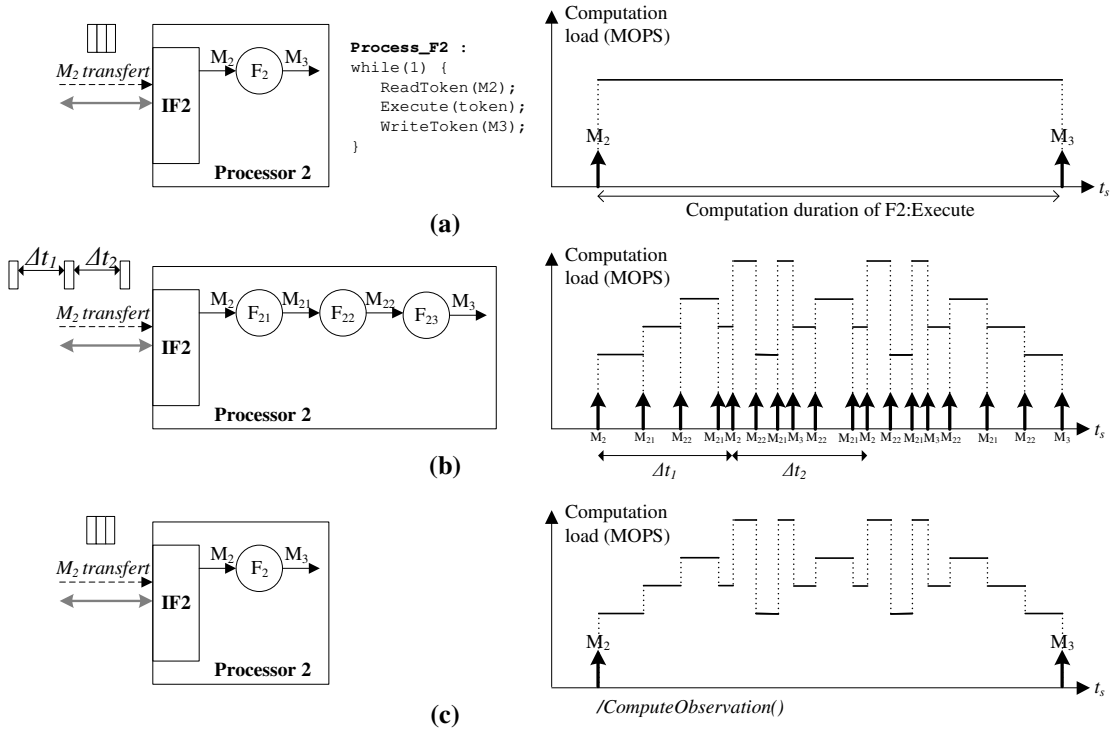


Figure 4.1: Modelling and evolution of the computation load according to different approaches. Δt_1 and Δt_2 denote the interval of time between two successive data to be processed. Situation (b) corresponds to a pipeline execution of the architecture.

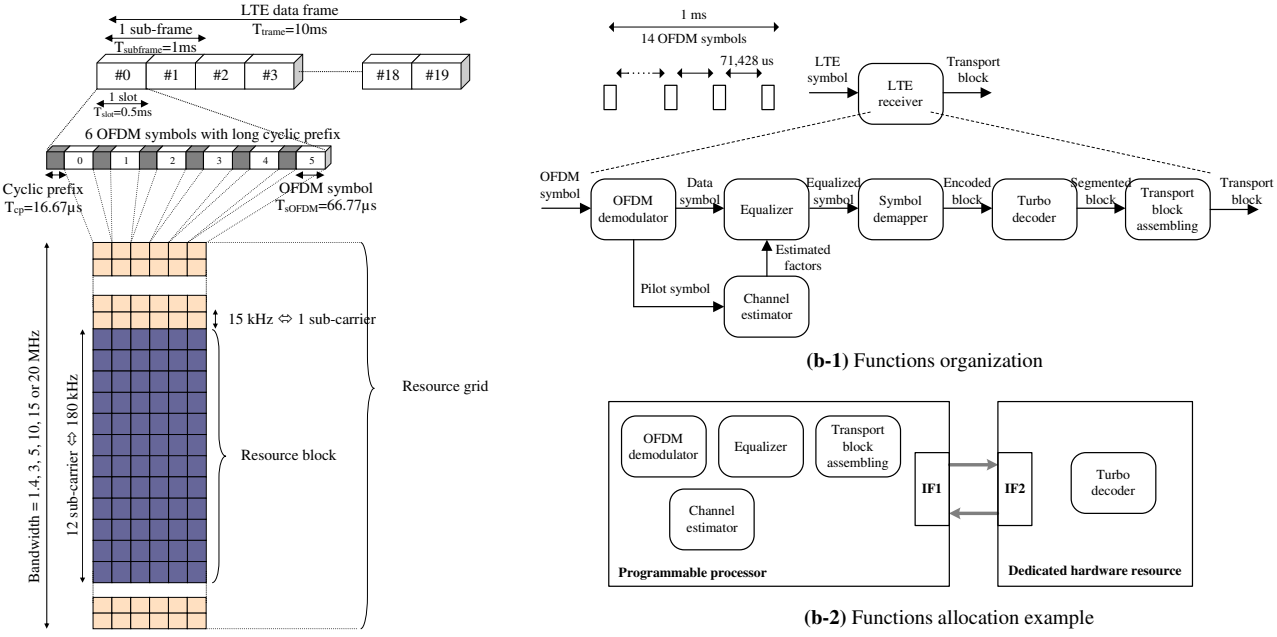
Part (a) of Figure 4.1 illustrates the modelling of a function executed on a processor with a given behaviour. The evolution associated with this model shows here the computation load induced (expressed in Mega operations per second (MOPS)) during the execution of the function F_2. At this level of detail, this load is described according to a unique value between the instant of reception of data via relation M_2 and the instant of production of data via M_3. Typically, this value may depend on the content of the data carried by the relation M_2. Part (b) of the figure illustrates a refined modelling, showing elementary functions composing F_2. The grain of the operated data is also refined. Each elementary function induces its own computation load, leading to a more precise evaluation of the occupation of the processor. At this level of detail, the number of exchanges generated between functions also turns out to be greater. The proposal made as part of

Mr. Barreteau's thesis aimed to limit the number of elements (functions and relationships) simulated within the performance models while trying to preserve the precision of the observations made for finer levels of detail. Part (c) of the figure illustrates this proposition. In the case observed, a level of precision equivalent to case (b) is reached for a limited number of functions and exchanges between functions. The principle then proposed consisted in establishing a decoupling between the simulation time noted t_s and the observation time manipulated locally within the simulated function. The observations made locally are then calculated at the times when the function is triggered (reception instant of M_2 in the case of Figure 4.1).

This proposal uses the principle of instantaneous calculation presented in Chapter 2. As mentioned previously, this approach makes it possible to reduce the number of calls required to the simulation engine by locally calculating some of the instants when computation or communication resources of the platform are used. As part of M. Barreteau's thesis, we considered this method in order to limit the number of functions used to predict the computing capacity of the studied systems. Originally, this reduction was done by combining within the simulable model the loads induced by all the elementary functions considered [C4-1], [C4-2]. It was then latter that we considered the use of (max, plus) algebra and the formalism of Petri nets in order to systematize the definition of the calculations carried out during the simulation and to establish a generic model of execution used for the simulation of the models. In the context of Mr. Barreteau's thesis, this proposal was notably applied to the following case study inspired by the field of mobile radiocommunications.

Study of the physical layer of the LTE protocol

This case study addressed the sizing of the computing and memory resources required to support the processing associated with the physical layer of the LTE (*Long Term Evolution*) [51] communication standard. The radio access technology associated with this standard, called E-UTRA (*Evolved Universal Terrestrial Radio Access*), has been defined in order to support transmission rates of up to 100 Mbit/s for the downlink. Part (a) of the figure 4.2 illustrates the organization of a reference LTE radio frame.



(a) Organization of a LTE data frame

(b-1) Functions organization

(b-2) Functions allocation example

(b) Structural organization of functions and resources of a LTE receiver

Figure 4.2: (a) Organisation of an LTE data frame in the time and frequency domains, (b-1) organization of the functions of the physical layer in reception, (b-2) possible allocation of functions on two types of computing resources.

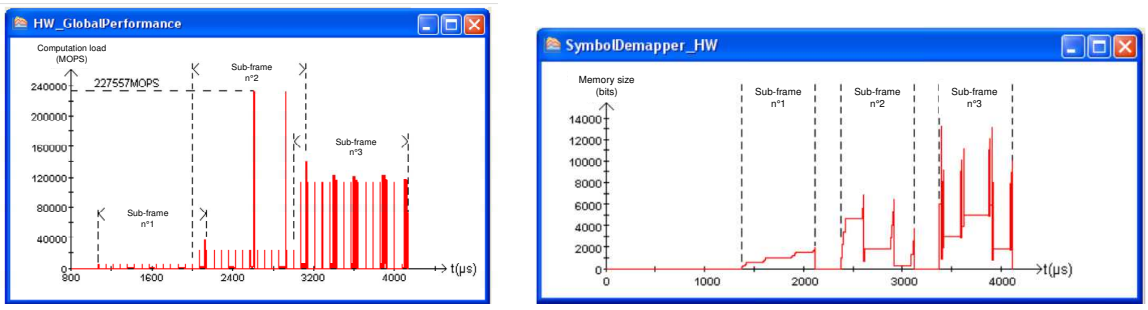
This reference frame is organized in the form of ten sub-frames, themselves divided into two time slots of 0.5 ms. Each elementary interval is made up of six or seven OFDM symbols (*Orthogonal Frequency Division Multiplex*) spaced by long or short cyclic prefixes. The OFDM modulation technique makes it possible to allocate one or more blocks of resources to each receiver according to the quantity of information to be transmitted and the timing constraints to be respected in order to ensure a given level of quality of service.

A resource block designates the smallest unit used to send data to a mobile terminal, corresponding to the association of twelve consecutive sub-carriers. Thus, in order to be able to adapt the quantity of data required for each receiver, the number of allocated resource blocks, the number of occupied sub-carriers as well as the type of modulation used can change from one subframe to another. For this case study, only a configuration with a single antenna for transmission and reception was considered. Part (b) of Figure 4.2 represents the organization of the main functions of the physical layer executed upon receipt of an LTE subframe in order to form a block of data used by the upper layer of communication. The main time constraint imposed was to process fourteen OFDM symbols in 1 ms. The role and behaviour of each function is detailed in [50]. Also, different expressions were established in order to formulate the computation load induced by each elementary function taking into account the different possible configurations of the received sub-frames.

Such a receiver was approached according to the two levels of abstraction illustrated in part (b-1) of Figure 4.2:

- a first level displays the six functions operating on each OFDM symbol received,
- a second level corresponds to the only reception function operating on each OFDM symbol received, the global calculation load induced by all the elementary functions being calculated locally. This calculation is carried out according to the dependencies between functions and according to the allocation considered on the platform studied.

The contribution of the proposed approach therefore consisted in the possibility of analyzing the workload induced for different architectures, while keeping a limited number of functions and communications within the model created. These different models were entered into the Intel CoFluent Studio environment in order to verify the validity of the models and the gain in terms of simulation speed. The models created made it possible to estimate the computing and memory capacities required for two possible architectures: a first architecture using dedicated hardware resources for the execution of all the functions and a second architecture associating programmable processor and dedicated accelerator for the single channel decoding function. The figure 4.3 illustrates the observations obtained for the evolution of the computation load induced by the functions of the physical layer as well as the memory required for the first architecture.



(a) Usage of computation resources.

(b) Usage of memory resources.

Figure 4.3: Observations of the occupation of computation and memory resources for an architecture based on a dedicated hardware resource [50], [J4-1].

The presented observations relate to the reception of three distinct sub-frames with varying number of resource blocks and type of demodulation used. Such observations make it possible to estimate the occupation of the calculation resources as data frames are received. These observations were obtained with the same level of precision for the models created according to the two levels of abstraction previously mentioned. The gain obtained in terms of simulation speed was of the order of 4 between the two models created within the Intel CoFluent Studio environment. Subsequently, we used the approach presented in Chapter 2 to describe the load model associated with the different functions of the LTE receiver. The graph then used in simulation comprised 84 nodes corresponding to the different synchronization instants calculated locally at the reception of each OFDM symbol. This case study was therefore initiated as part of Mr. Barreteau's thesis and was then continued in order to illustrate the validity of the extended approach.

4.2.2 Modelling of dynamic functions

One characteristic of the studied radiocommunication systems was their ability to support a wide variety of applications and to use different communication standards and protocols. These various functionalities could be activated during the operation of the system and this taking into account the needs of the user and the available communication infrastructures. The modeling and simulation of dynamically activated functions, *i.e.* during operation, supposed to extend the capacities of the software environment used within the framework of the thesis M. Barreteau. Classically, any function identified within a performance model is created at the start of the model simulation. Once the simulation has started, this function can be executed as soon as its activation conditions are met (*e.g.*, availability of the data to be processed or the computing resources). We therefore proposed the case of so-called dynamic functions whose creation could be controlled during the simulation and no longer only at the start of the simulation. A dynamic function as we have defined it evolves according to three states:

- the *Stopped* state designates the initial state of the function when the simulation starts. This state corresponds to the case of a function not created or not allocated on a computation resource.
- The *Started* state represents the state where the feature is created and activated. The function is then allocated and can thus be executed.
- The *Suspended* state represents an intermediate state where the function is created but can no longer be executed.

The state control of a dynamic function is carried out from a specific relation indicating the state required for the function. This adaptation of the behaviour of the functions was integrated within the CoFluent Studio environment and could thus be used within the framework of M. Barreteau's thesis. This adaptation corresponded to the addition of a process specific to each function in order to manage their current state during the simulation. The case study presented below used this new modelling and simulation capability.

Study of a multi-standard mobile terminal

This case study focused on the modeling of a mobile radio terminal able to support several communication protocols and to autonomously select the protocol adapted to the operating situation. We considered the case of a mobile terminal implementing the E-UTRA and WiFi access technologies in order to support voice communication, Internet browsing and streaming video playback applications. The system thus imagined had to be capable of autonomously selecting the communication interfaces to be used given a changing network environment, offering variable communication capacities (urban or rural environment). The objective of the study related to the analysis of the evolution of the computation loads induced by the various communication interfaces and this according to various possible operating scenarios.

Figure 4.4 illustrates the adopted modelling of the communication interfaces. The selected level of granularity corresponds to the reception of data frames of duration equal to 10 ms for E-UTRA technology (considering a bit rate of 384kbit/s) and 347 us for WiFi (for a considered speed of 54Mbit/s). The behaviour of the E-UTRA reception and WiFi reception functions is described in such a way as to represent the processing related to the MAC and RLC (*Radio Link Control*) protocol layers of the E-UTRA and LLC (*Logical Link Control*) from WiFi. These protocol layers ensure in reception the demultiplexing of the different data streams contained in a radio frame. Each stream is associated with an application used by the user. They also carry out the reconstitution of the data blocks associated with each application which have been previously segmented to be transmitted over the radio interface.

In the system studied, the communication interfaces were activated according to the needs of the applications used. These functions were modelled as dynamic functions in order to describe the influence of their activations and deactivations during the operation of the system. The relation noted *RAT control* was then used to activate these functions taking into account the quality of service required by the applications used.

The model of the studied system was described and simulated within the Intel CoFluent Studio environment. Figure 4.5 illustrates a simulated operating scenario and the influence on the latency of the considered applications. The model created is simulated according to a test environment for which the three types of

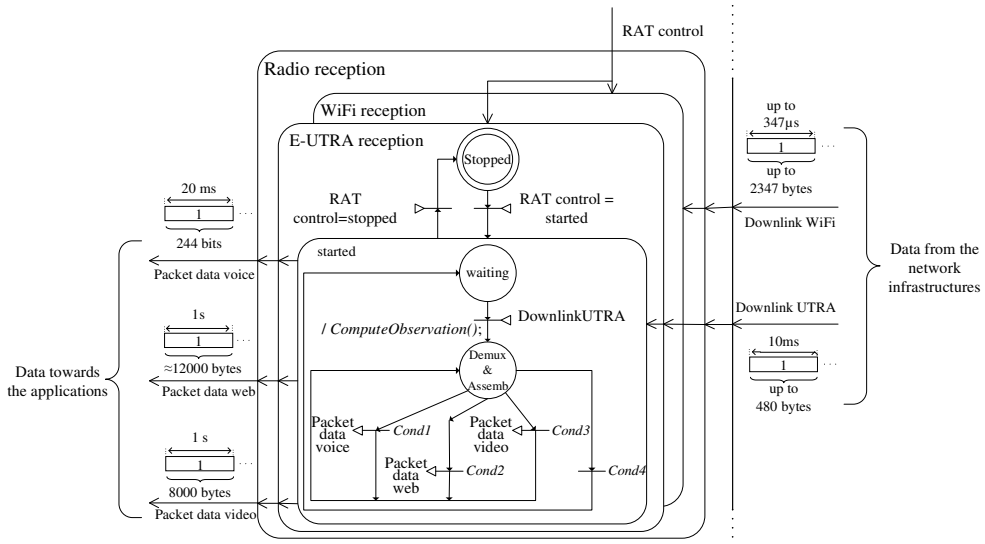


Figure 4.4: Description of the behaviour of the communication interfaces according to the dynamic function principle [J4-1].

applications are successively activated. The available networks are also evolving by offering different possible speeds (E-UTRA at 384 kbps then 130 kbps, WiFi unavailable then available at 1500 kbps). The scenario represented in Figure 4.5 illustrates the transition of video data transmission from the E-UTRA protocol to the WiFi protocol. The latency associated with streaming video increases given the reduced throughput offered by E-UTRA technology. The selection of the WiFi technology is then carried out autonomously and leads to reducing the latency of this application.

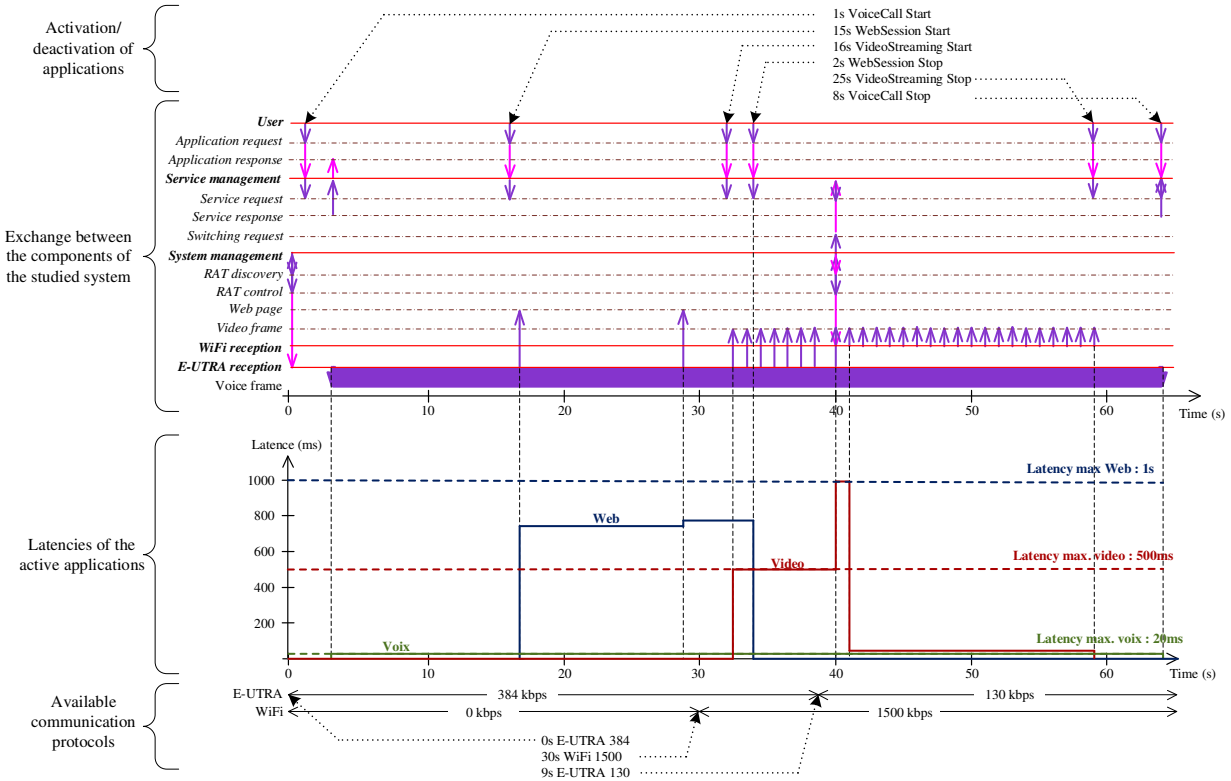
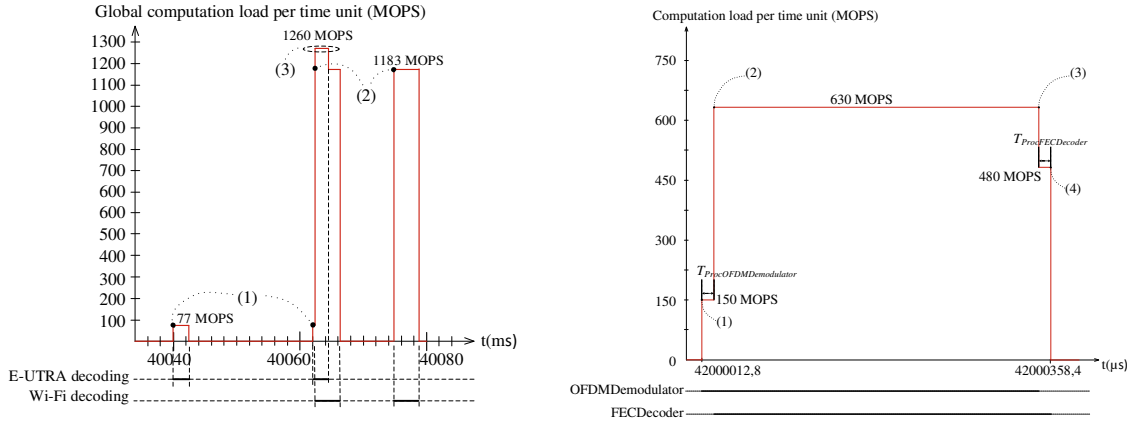


Figure 4.5: Evolution of the usage of resources according to an operating scenario that modifies the applications executed as well as the network available [J4-2].

The proposed method of integrating non-functional properties was also used to complete the description of the behaviour of communication interfaces. The computation loads associated with the different functions of the physical layers of the two communication protocols were thus integrated into the description of the interfaces. As shown in Figure 4.4, the action `ComputeObservation` is executed upon receipt of the data in order to describe the use of the computing resources associated with the functions of the physical layer of each interface. It was thus possible to take into account the computation load induced by the different functions making up the physical layer of the standards considered. The characteristics of such a system and the details of its modeling are presented in [50], [J4-1]. Figure 4.6 illustrates two results obtained for the observation of the use of resources according to the operating scenario considered.



(a) Evolution over time of the computation load (in MOPS) for the E-UTRA and WiFi interfaces (channel decoding function).

(b) Usage of computation resources for the WiFi reception process.

Figure 4.6: Observation of the computation loads induced by the physical layer functions of the E-UTRA and WiFi protocols [J4-2].

Part (a) of Figure 4.6 represents the evolution according to the considered operating scenario of the total computation capacity required for the E-UTRA and WiFi interfaces. The instants noted (1) and (2) correspond to the start times of the channel decoding function on the received data. The interval noted (3) corresponds to the time interval for which the computing capacity is maximum, corresponding to the moment when the two interfaces are simultaneously active. Part (b) of Figure 4.6 represents the evolution of the computing capacity required for the WiFi interface for the processing of a WiFi data frame. This frame is made up of 27 OFDM symbols, the demodulation and channel decoding functions being applied successively to these symbols. In part (b) of Figure 4.6, instants (1) and (2) designate the times at which these functions start during the processing of the first OFDM symbol, instants (3) and (4) designate the instants of end of processing of the last OFDM symbol.

The two aspects addressed in the context of Mr. Barreteau's thesis could thus be put into practice within a professional modeling and simulation environment. The contributions made it possible to approach the dimensioning of the resources of mobile radiocommunication systems for advanced operating scenarios. Subsequently, Ms. Yang's thesis was the opportunity to deepen the study of online management of multiprocessor systems.

4.3 Management strategies of multiprocessor architectures

Ms Simei Yang's thesis focused on the optimization under timing and energy constraints of multiprocessor and multicore architectures (MPCSoC, *Multi Processor and Core System on Chip*). The application framework corresponded to the case of multiple data flow applications that could be executed simultaneously according to different possible operating scenarios. The main objective of the thesis focused on the definition of online management strategies for multiprocessor platforms in order to jointly optimize the allocation and scheduling of tasks on computing resources as well as the operating frequency of the processors.

4.3.1 Related work and contribution of our work

The gradual increase in the number of processor cores is observed in many fields of application of embedded systems. Such multiprocessor platforms deliver significant execution parallelism while allowing to limit the required operating frequencies of the cores. These platforms favour the simultaneous execution of multiple applications, these applications being themselves described according to a set of tasks in order to take advantage of the offered parallelism. The process that establish the allocation and scheduling of tasks and communications between tasks on the platform's available resources[1] is conducted considering criteria such as performance and power consumption. The establishment of an optimized mapping can be carried out during design phase (static definition) or during operation (dynamic definition). Many works exist in order to propose static approaches for optimizing the mapping of data flow applications on multiprocessor platforms [52], [53]. Within the framework of this study, we were interested in the case of systems for which the workload can evolve during operation according to the applications simultaneously executed. For such systems, the static definition of optimized mappings lead to consider a design space that is far too vast given the possible combinations of applications. Consequently, the dynamic definition approaches and therefore online management of resources represent interesting solutions in order to adapt the use of resources during operation and thus optimize the performance and the overall energy consumption of the system. Among the various dynamic definition approaches [54], we considered a hybrid approach associating static preparation of certain possible mappings for each application and definition during operation of an optimized solution for all the supported applications.

The studied systems consider a set of applications app_i executed according to different possible operating situations u_m. Each situation u_m is characterized by the set $\{app_1, \ldots, app_I\}$ of executed applications. In the study conducted, the considered applications are inspired by the multimedia field, corresponding to the periodic processing of large data streams. These periods represent constraints on the execution times of the applications. For each situation u_m, a hyper-period is defined as the least common multiple among the periods of the applications of u_m. The SDF formalism is used to describe each application, even if this choice was not a limitation as to the contributions made. In the developed hybrid approach, different possible mappings are considered for each supported application. Figure 4.7 illustrates the notation used to describe the applications studied and the associated mappings. As shown, these mappings represent distinct solutions in terms of

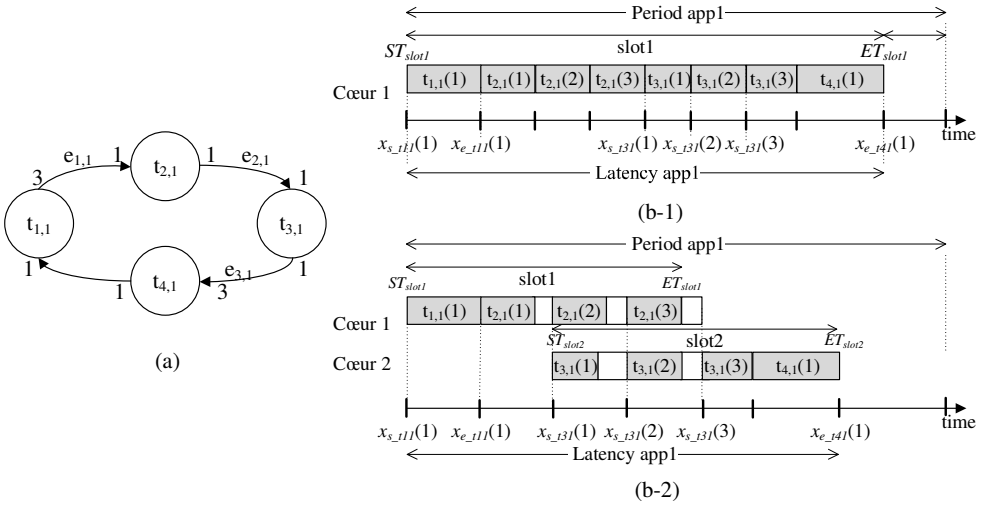


Figure 4.7: (a) Example of an application described according to the SDF model of computation, (b) Computation resources usage for two distinct mappings associated with the application app_1.

latency and platform resource usage. Each considered mapping is characterized by a periodic pattern formed by the instants $x_i(k)$ when the resources of the platform are used. This pattern corresponds to an execution trace established for each mapping over a given period. In Figure 4.7, the notion of *slot* designates a time interval during which a computing resource is used by the same task. The instants ST and ET designate the

[1] This process and the resulting result will be referred to by the term *mapping*.

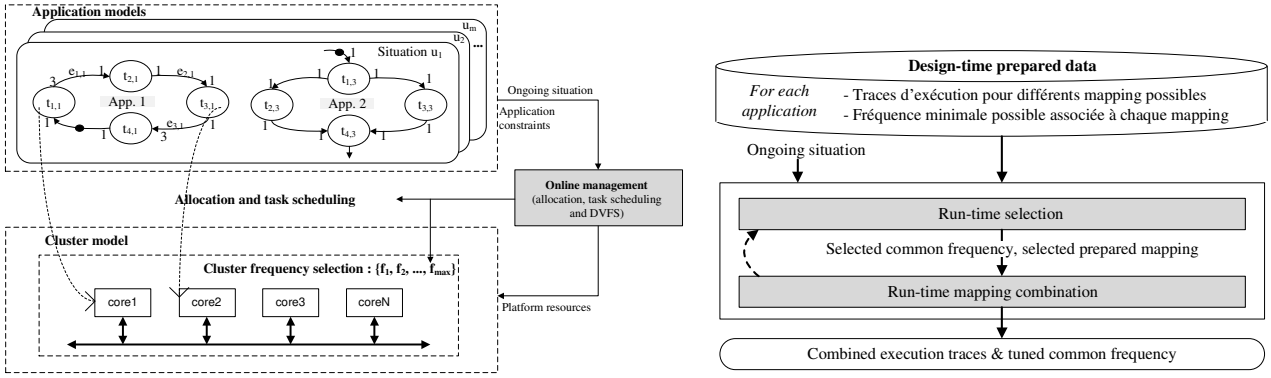
start and end times of the interval.

These sets will be referred to as *clusters*. This type of organization makes it possible to allocate distinct operating frequencies for each cluster [55]. The proposed dynamic management strategies aimed to minimize the average power consumed over each hyper-period by optimizing the mapping and the operating frequency of the clusters considered. These strategies were proposed at two levels: at the local level within a cluster made up of several cores, and at the global level on the scale of several clusters.

4.3.2 Proposed online management strategies

Local management of computation resources

This sub-section concerns the intra-cluster optimization of the computation resources usage in order to minimize the average power consumed for different applications executed simultaneously. The position of the online manager is shown in figure 4.8.



(a) Online management of resources. **(b)** Proposed management strategies.

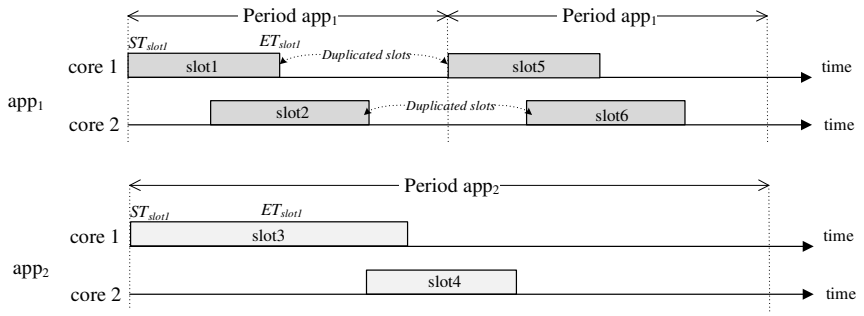
Figure 4.8: (a) Position of the online manager in relation to the applications running within the same cluster, (b) Process for selecting the frequency of the cluster operation and defining task allocations.

The object of the studied manager is to establish the operating frequency of the cluster as well as an optimized combination of the execution traces of each of the active applications. In doing so, the timing constraints specific to each application as well as the number of available computing resources must be respected. The proposed strategy assumes a set of information established statically during design:

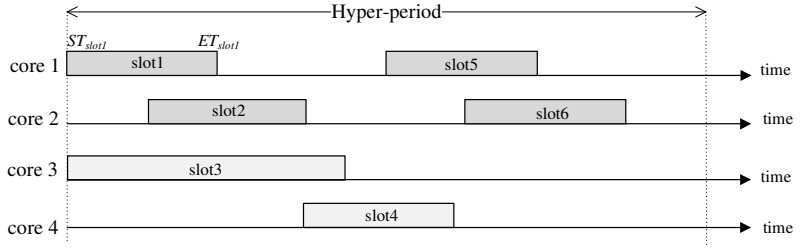
- for each application supported by the cluster, a limited number of possible mappings are established for different numbers of cores. The considered mappings reflect different occupations of computation resources and thus distinct latencies for each application.
- A specific operating frequency is associated to each considered mapping, corresponding to the smallest operating frequency of the cluster for which the timing constraint associated with the application is respected.

This information is used at the start of each new operating situation u_m in order to establish a scheduling and an optimized allocation for all the active applications. To do this, the iterative process illustrated in part (b) of Figure 4.8 is proposed. The first step consists in selecting among the prepared frequencies the smallest value allowing the active applications to respect their own constraints. This selection leads to selecting a first set of execution traces used in order to establish the occupation of the resources for the active applications. An original method was proposed to optimize the allocation and scheduling of active applications on the computing resources of the cluster [C4-3]. This method uses the prior knowledge of the time intervals during which the computation resources are used for each mapping. Figure 4.9 illustrates this method in comparison with two methods from the literature in the field.

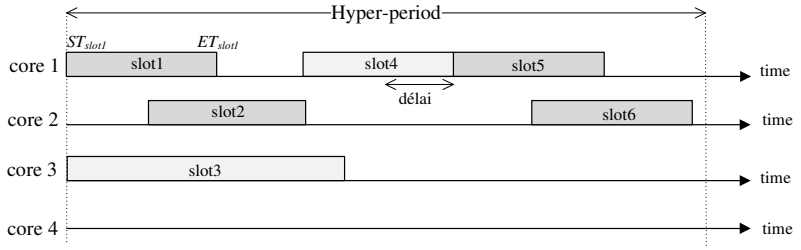
Part (a) of the figure illustrates the time intervals during which two applications denoted app_1 and app_2 use the computing resources. In the considered example, $Period_{app_1} = 2 \times Period_{app_2}$ and the intervals



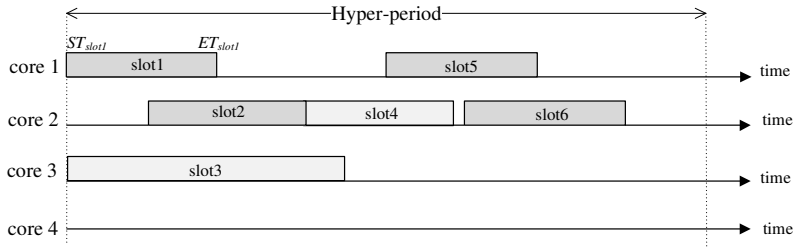
(a) Considered mappings for the applications app_1 et app_2



(b) Combination of execution traces with the FCFS method



(c) Combination of execution traces with the LASP method



(d) Combination of execution traces with the GAPVC method

Figure 4.9: Different methods of allocating timeslots based on prepared execution traces: (a) prepared execution traces for two active applications, (b) combining traces according to the FCFS method, (c) combining traces according to the LASP method, (d) combining traces according to the proposed GAPVC method.

slot5 and slot6 correspond to the duplication of the intervals slot1 and slot2. Part (b) represents a possible combination of time intervals considering the so-called FCFS (*First Come First Served*) approach used in [56]. This approach considers the allocation of a single application per core, avoiding the possible sharing of computing resources. Part (c) represents the combination of time slots according to the LASP approach (*Longest Available Slot Packing*) presented in [57]. This approach successively allocates the tasks taking into account the availability of the cores. In the case illustrated, this approach results in allocating the interval slot4 on core 1 and delaying the execution for the interval slot5. Thus, it has been observed that the LASP approach can lead to increasing the latency of applications over a hyper-period, ultimately leading to higher operating frequencies. The method proposed by Miss Yang makes it possible to preserve the allocation of periodic executions within a hyper period while seeking to limit the number of resources used. This method called GAPVC (*Grouped Applications Packing under Varied Constraints*), illustrated in part (d), considers the

availability of cores by taking into account the duplicate intervals within a hyper-period. The objective of this method is then to establish the allocation of the slots taking into account the duplicated intervals and seeking to limit the number of cores used. Thus, this approach leads to limiting the number of cores used compared to the FCFS method and to limiting the performance degradation observed for the LASP method. On part (d) of Figure 4.9 the interval slot4 is allocated on core 2 in order to avoid delaying the execution of slot5 and in order not to use any additional core . The detailed algorithm of this method is given in [58] and [C4-3].

In order to evaluate the effectiveness of the proposed method, nine different applications were considered. These applications were described according to the SDF formalism with different numbers of tasks, relations and transfers between tasks within an execution period. For each application, between 1 and 3 distinct mappings relating to the use of 1, 2 or 4 cores were prepared statically. For the 9 applications studied, 511 possible operating situations were considered. For each operating situation, the latency obtained for each application and the average power consumed for each combination of applications were then compared according to the method used. Table 4.1 from [58] compares the performance obtained for the GAPVC allocation and scheduling method with the FCFS and LASP methods.

Table 4.1: Comparison of multi-core platform online management strategies for 511 different operating situations.

Number of cores	Strategie	Nb of not satisfactory situations			Comparison with the proposed approach	
		Total	Due to not enough resources	Due to timing constraints	Nb of situations with higher average power	Increase in average power
6	FCFS	102	102	0	16	12.1% to 34.6%
	LASP	90	67	23	31	142.8% to 206.3%
	GAPVC	76	76	0	-	-
8	FCFS	9	9	0	32	12.8% to 35.6%
	LASP	34	2	32	40	206.3%
	GAPVC	3	3	0	-	-
10	FCFS	0	0	0	5	13.6% to 36.0%
	LASP	32	0	32	40	206.3%
	GAPVC	0	0	0	-	-

For the three sets of cores considered, the number of situations where the generated combination is not satisfactory is estimated. The gain in terms of average power is estimated according to the methods evaluated. The average power was estimated considering the occupation rate of the processor cores and the obtained operating frequency [58]. Thus, in the case of 10 cores, the GAPVC method leads to a reduction in the average power consumed of the order of 206% compared to the LASP method and between 16.6% and 36% with the FCFS method.

We could not compare these predictions to measurements made on a real target. However, the implementation of the GAPVC algorithm was considered for an Exynos 5422 multiprocessor platform based on an ARM big.LITTLE [58] architecture. The execution time of the algorithm is measured for the 511 possible operating situations considered. The average time obtained is 0.137 ms on the LITTLE part (Cortex-A7 core clocked at 1.4 GHz) and 0.037 ms on the big (core Cortex-A15 clocked at 2.1 GHz). The execution time achieved for this algorithm turn out to be lower than those of the FCFS and LASP algorithms.

Global management of computation resources

We consider here the case of platforms organized in cluster, where each cluster groups together a homogeneous set of processor cores and can operate at a specific frequency. For this type of platform, the cluster used for the execution of an application can vary according to the operating situation u_m considered. The contribution made aimed to reduce the average power consumed across multiple clusters by optimizing the allocation of applications running simultaneously. The proposal made, illustrated in Figure 4.10, relates to the definition of a hierarchical management of resources. At the scale of the global platform, the proposed method aims to establish at the beginning of each new operating situation u_m the allocation of applications on the clusters as well as the operating frequency required for each cluster. At the scale of the cluster, it is a question of

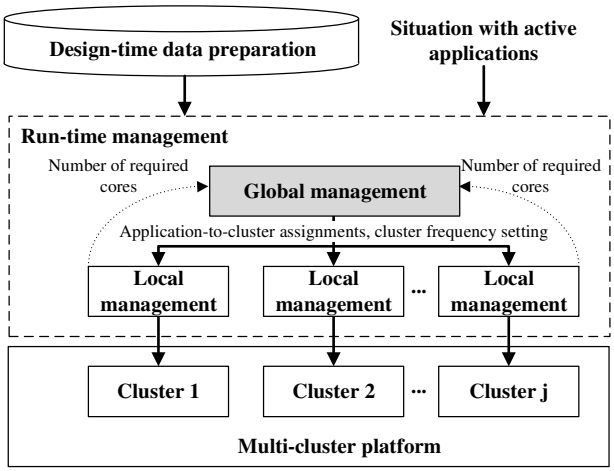


Figure 4.10: Hierarchical organization for managing multiple applications running on a platform consisting of multiple clusters.

establishing the allocation and the scheduling of the tasks of the various applications allocated on the available cores. The management made is based on a set of statically prepared data: for each supported application, a set of possible mappings and, for each mapping, the minimum operating frequency to meet the timing constraints.

The global management approach aims to establish an allocation of applications and a possible frequency of operation of each cluster that minimizes the average power consumed by the entire platform, while taking into account the number of available computing resources and the timing constraints specific to each application. Two allocation strategies were proposed by Miss Yang and are detailed in [58] and [J4-3]. These strategies apply at the start of each new operating situation. The first strategy, named *Neighboring Search Application-to-cluster Assignment* (NSACA) consists in successively allocating the applications according to their minimum operating frequencies, taking into account the number of computing resources available within each cluster. This strategy does not take account of the allocations prior to the new operating situation evaluated and can therefore generate numerous re-allocations of applications. The second strategy, named *Greedy Search Application-to-Cluster Assignment* (GSACA), is proposed in order to control the number of re-allocations authorized from one operating situation to another. This strategy allocates newly active applications on clusters delivering a compatible operating frequency and a sufficient number of computing resources. The re-allocation of already active applications is considered in order to further optimize the overall consumption of the newly formed set. According to this approach, the previously presented scheduling strategy, GAPVC, is used to estimate the number of computing resources employed within a cluster.

The evaluation of these strategies, detailed in [58] and [J4-3], was carried out by considering the dynamic allocation of different applications for multiple possible operating situations. This evaluation was conducted for platforms with different numbers of clusters and different numbers of cores per cluster. Figure 4.11 compares the average consumption obtained for the allocation of ten distinct applications according to 1023 possible operating situations. The comparisons relate to the results obtained for the GSACA strategy relative to different strategies in the domain: an exhaustive strategy considering the evaluation of all possible solutions, the LEF strategy (*Low Energy First*) presented in [55], the LPF (*Low Power First*) strategy presented in [59].

The GSACA method presents performances close to those obtained according to an exhaustive approach for solution search, and this for the various studied platform configurations. On all the configurations considered, the gains observed compared to the LEF and LPF strategies go up to 21.2% and 80.3% respectively. In [J4-3], the influences of the number of application re-allocations and the application scheduling strategy (GAPVC or FCFS) are also evaluated.

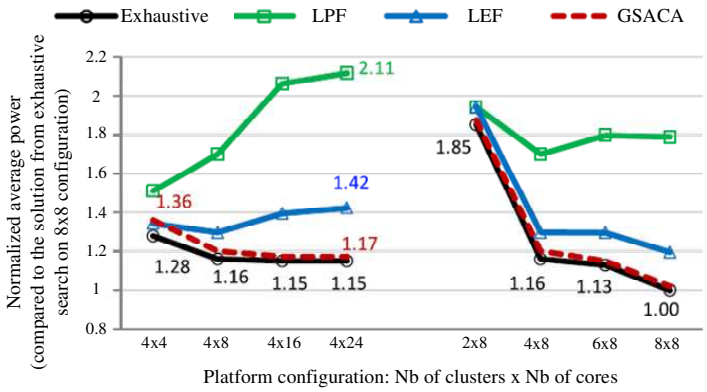


Figure 4.11: Comparison of average power relative to the platform configuration 8×8 [J4-3]. For this comparison, no application re-allocation is necessary.

4.3.3 Online management architecture modelling and simulation

The purpose of this part of the work carried out within the framework of Ms. Yang's thesis focused on the consideration of online management strategies in system-level modelling and simulation process of architectures. The objective was to allow the simulation of applications executed on high-level models of the targeted platforms, taking into account the developed online management strategies. While many system-level modelling and simulation environments exist, few of them allow for the consideration of online resource management and modification of allocations. In [60], the Sesame modeling and simulation environment is extended to take into account the online resource management of a multiprocessor platform. This extension makes it possible to adapt during the simulation the way in which the loads induced by the functions of the application are allocated to the resources of the platform model. In [61], an adaptation of the Intel CoFluent Studio environment is proposed. A specific API is proposed in order to be able to adapt the allocation of functions during the simulation, however this API is not currently available in the current version of this environment.

This work aimed to take into account the possible modification during the simulation of the allocation of functions on the computing resources of the platform model. The experimented proposal is based on the fact that the online management methods studied in the context of Ms. Yang's thesis lead to establishing for each new operating situation an optimized mapping as well as a new operating frequency. Thus, at the start of each new operating situation u_m, a new execution trace is established prior to the execution of the applications on the resources of the platform. In simulation, knowledge of future execution instants allows the manager to successively activate the tasks at the required instants. These instants of activation take into account the evolution during the simulation of the distribution of the functions on the resources of the considered platform. The simulated manager therefore acts in order to notify the availability of the computing resources during the execution of the functions of the application model. Figure 4.12 illustrates this principle for the case of the simulation of two successive situations u_1, considering the simultaneous execution of two applications app_1 and app_2, then u_2, considering the execution only of app_2. At the start of each operating situation u_1 and u_2, the manager establishes an operating frequency and an execution trace, thus establishing the instants of use of the resources of the platform. This knowledge allows the manager to activate the tasks according to the required order and at the newly calculated instants. In Figure 4.12, the instants of start and stop of the functions by the manager are represented. These instants then take account of the resources used.

This approach was implemented within the Intel CoFluent Studio modelling and simulation environment. Figure 4.13 illustrates this implementation by positioning the manager model with an application model. In the implementation performed, the manager model is graphically captured as a function activated at the start of each new operating situation. The action noted `Compute` corresponds to a C++ code establishing the traces associated with each application. During the simulation, the manager controls the state of each function of the applications considered at the calculated instants, controlling thus access to the computing resources of the platform.

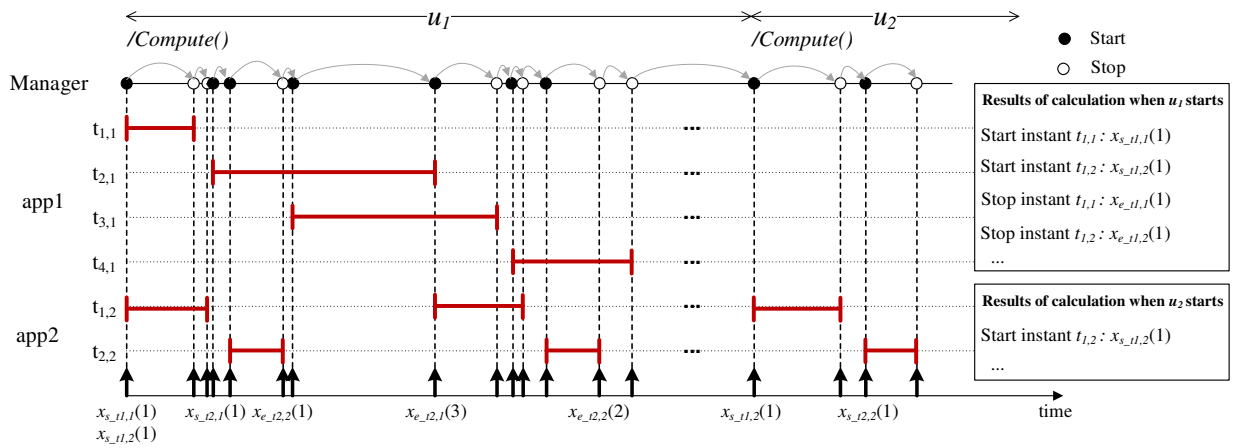


Figure 4.12: Simulation principle using the traces established during each new operating situation.

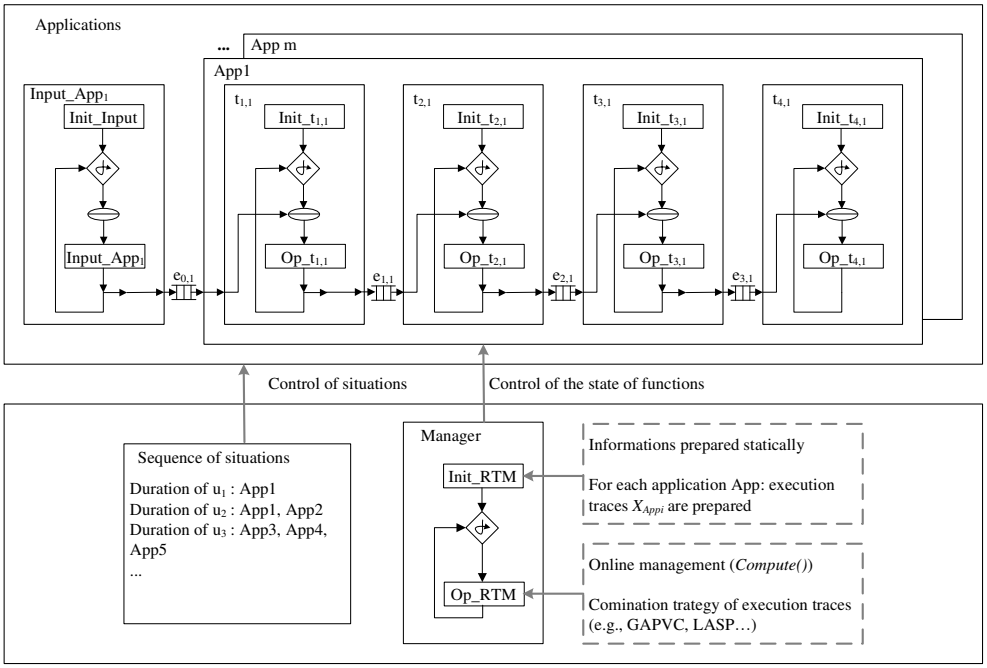


Figure 4.13: A high-level model of an application and associated online manager within the Intel CoFluent Studio environment.

In order to illustrate this implementation, Figure 4.14 illustrates a possible observation of the evolution over time of the dynamic power consumption caused by an application formed of 4 tasks [C4-4]. During the simulation, the manager establishes different operating frequencies and different possible allocations of the tasks. The task execution instants are established and used to activate accordingly the various tasks. The intervals noted (1), (4), (5) and (6) correspond to the durations of execution of the same task but for distinct operating frequencies and allocations.

4.4 Conclusion

Originally, Mr. Barreteau's thesis focused on the study of mobile radiocommunication systems integrating adaptation mechanisms as imagined according to the paradigms of software defined radio and cognitive radio. Initially, the work carried out focused on the integration of non-functional properties with a view to evaluating performance for complex operating scenarios. This work led to the proposal of a simulation method limiting the number of calls to the simulation engine while preserving the precision of the models created.

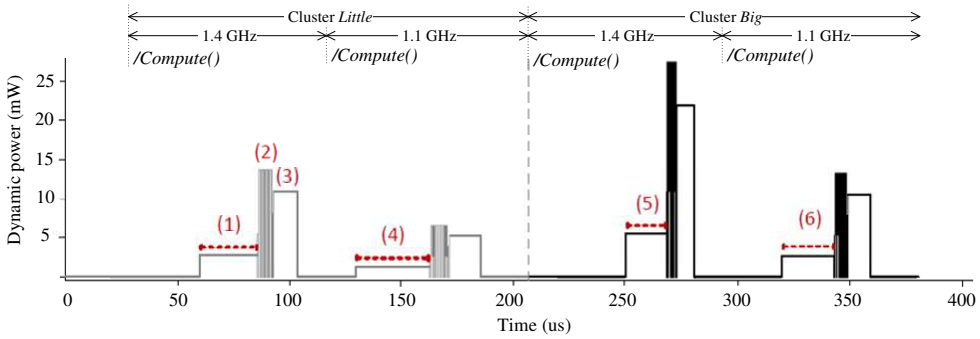


Figure 4.14: Simulation of dynamic power consumption for different operating frequencies and different possible allocations of the considered application [C4-4]. The simulation is conducted within the Intel CoFluent Studio environment using the proposed principle.

This thesis was therefore a first step towards the establishment of a general and systematic approach that was subsequently pursued. In a second step, it was a question of approaching the consideration of dynamic adaptation mechanisms within such systems and this in high-level modelling and design approaches. From an organizational point of view, this thesis corresponded to a study carried out not in connection with the development of a new tool but rather with the idea of using and extending the possibilities of an existing environment (Intel CoFluent Studio). Such positioning can be difficult to value the contributions made. Subsequently, Mr. Barreteau was hired within the company ensuring the development of this environment and still participates in it within the Intel company.

The work carried out within the framework of Ms. Yang's thesis represented an opportunity to address the optimization of the energy consumption of multiprocessor architectures. Ms. Yang made substantial efforts to establish original proposals in this very active field of research. However, this work could not be supplemented by experiments on targets which would have been necessary in order to validate the hypotheses made and the results obtained. This work made it possible to transpose certain principles associated with the modelling and simulation approach developed in our other recent projects. Thus, based on the models developed elsewhere, it would be interesting to continue the work started as part of Miss Yang's thesis in order to optimize the energy consumed, also taking into account the use of communication and memory resources of multiprocessor platforms.

4.5 Suepervision

PhD thesis

- Simei Yang, *Evaluation and design of a run-time manager for ultra-low power multiprocessor systems on chip*, University of Nantes, June 2020.
- Anthony Barreteau, *Transaction-level modelling methods for resource definition of future mobile radiocommunication systems*, University of Nantes, December 2010.

Master thesis

- Anthony Barreteau, *Automated generation of test environment from high level models*, University of Nantes, June 2007.
- Romain Guignard, *Study and modelling of a SoC architecture in CoFluent Studio*, University of Nantes, June 2006.

4.6 Publications related to Chapter 4

International journals with program committee

[J4-1] A. Barreteau, S. Le Nours, and O. Pasquier, "A state-based modeling approach for efficient performance evaluation of embedded system architectures at transaction level," *Journal of Electrical and Computer Engineering*, vol. 2012, Article ID 537327, 16 pages, 2012. DOI: 10.1155/2012/537327. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00664369.

[J4-2] ——, "A case study of simulation and performance evaluation of a sdr baseband architecture," *Journal of Signal Processing Systems*, vol. 73, pp. 267–279, Jun. 2013. DOI: 10.1007/s11265-013-0764-0. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00863981.

[J4-3] S. Yang, S. Le Nours, M. Mendez Real, and S. Pillement, "0–1 ilp-based run-time hierarchical energy optimization for heterogeneous cluster-based multi/many-core systems," *Journal of Systems Architecture*, vol. 116, p. 102 035, 2021, ISSN: 1383-7621. DOI: https://doi.org/10.1016/j.sysarc.2021.102035. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1383762121000370.

International conferences with program committee

[C4-1] S. Le Nours, A. Barreteau, and O. Pasquier, "Modeling technique for simulation time speed-up of performance computation in transaction level models," in *Forum on specification & Design Languages*, Southampton, United Kingdom, Sep. 2010, FDL 2010. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00821173.

[C4-2] A. Barreteau, S. Le Nours, and O. Pasquier, "A simulation-based approach for performance evaluation of SDR baseband architectures," in *2012 Wireless Innovation Forum European Conference on Communications Technologies and Software Defined Radio (SDR'12 - WInnComm - Europe)*, Bruxelles, Belgium, Jun. 2012, SDR'12 - WInnComm –Europe. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00695537.

[C4-3] S. Yang, S. Le Nours, M. Mendez Real, and S. Pillement, "Mapping and Frequency Joint Optimization for Energy Efficient Execution of Multiple Applications on Multicore Systems," in *The Conference on Design and Architectures for Signal and Image Processing*, ser. DASIP 2019, Montreal, Canada, Oct. 2019, SUBMISS14. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02190486.

[C4-4] ——, "System-Level Modeling and Simulation of MPSoC Run-Time Management using Execution Traces Analysis," in *International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS XIX)*, Samos, Greece, Jul. 2019, paper #49. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02114092.

Chapter 5

Probabilistic modelling and simulation of multiprocessor systems

5.1 Introduction

In the context of embedded multiprocessor systems, the modeling of temporal properties is essential in order to be able to predict and optimize the temporal behavior of programs. However, the sharing of resources within multiprocessor platforms significantly influences program execution times and complexifies the modelling and analysis of temporal properties. In this context, the approaches of probabilistic analysis of temporal properties, *probabilistic timing analysis* (PTA), have emerged in recent years in order to favour the analysis of multiprocessor systems [62] . These approaches consider the analysis of execution times according to a set of possible executions in order to establish and analyze the distributions of the obtained values. Thus, the statistical method EVT (*Extreme Value Theory*) is considered in [63] in order to establish a probabilistic analysis of the worst case execution time (pWCET, *probabilistic Worst Case Execution Time*). In [64], the probabilistic model-checking method is adopted in order to analyze the temporal properties of dataflow applications presenting different possible operating scenarios. In [65], a probabilistic extension to the RTC-MPA [66] approach used for best and worst execution time estimation is presented.

In this context, based on the skills developed in modelling and simulation at the system level, my activity focused on the development of two aspects. The first point concerns the establishment of a measurement-based modelling flow for the creation of probabilistic models used for the analysis of the temporal properties of multiprocessor systems. The second point concerns the use of statistical model-checking methods in order to control the simulation of probabilistic models. This set-up aims to establish a complete modelling and analysis flow that we have evaluated, through various experimental achievements on real targets, the accuracy of the results, the speed of analysis and the ability to address complex architectures.

These activities were initially initiated through the Master's internships of Miss Nadia Ghazali and Mr. Jiatong Li [C5-1] then developed within the framework of the thesis of Mr. Hai-Dang Vu [47]. Also, during the thesis of Mr. Hai-Dang Vu, we were able to collaborate with the team of Mr. Kim Grüttner from the German institute OFFIS. This collaboration aimed at establishing and consolidating a common flow of modeling and analysis in order to evaluate the contribution of methods for the probabilistic analysis of multiprocessor systems. The two teams were thus able to share their activities through common case studies and develop complementary contributions. This collaboration has resulted in numerous study trips, in France and Germany, and regular remote meetings. The results of this work have led to various joint publications in an international journal [J5-1] and in international conferences [C3-4], [C5-2], [C5-3].

5.2 Establishment of a measurement-based characterization flow for the creation of probabilistic models

5.2.1 Proposal

This work aims to favour the creation of probabilistic models used for the analysis of the temporal properties of multiprocessor architectures. Two aspects of the functioning of such architectures have been more particularly taken into account: the variation of execution times induced by the data-dependency of programs as well as the effect of shared resources on the execution of programs. The established flow combines a measurement-based characterization approach of the elements of an architecture and the creation of simulable models. The originality of this work compared to the state of the art lies in the possibility to control the level of compositionality of the architectures in order to adopt probabilistic models appropriate to the operating situations observed. In this first part, the case of a fully compositional platform is discussed for which the characterization effort is limited. This choice implies adopting computation and architecture models guaranteeing a strict separation of communication and calculation mechanisms. The case of a non-compositional platform was considered in a second step. The organization of the flow implemented is illustrated in figure 5.1. As indicated, the presented flow comprises three distinct parts.

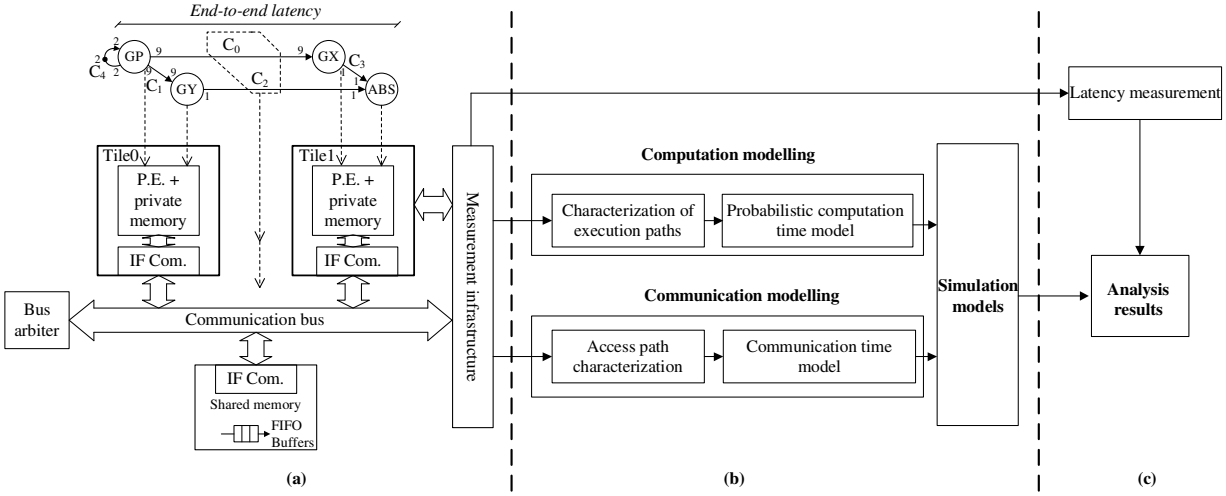


Figure 5.1: Measurement-based characterisation flow for the establishment of probabilistic models for the temporal properties analysis of MPSoC architectures.

Part (a) corresponds to the prototyping phase aiming to implement on a real target a given application on the resources of the studied platform. The established flow is based on a set of assumptions justified by the characteristics of the applications and the platform considered. Applications are described using the *Synchronous Data Flow* (SDF) computation model. This model considers the organization of applications in the form of a set of actors communicating through finite capacity FIFO-type communication channels. The behavior of each actor corresponds to a succession of phases of reading from the input channels, execution of the computation and writing to the output channels. This assumption leads to a strict separation between computation and communication mechanisms.

In order to respect the separation of communication and computation mechanisms, the considered platform is organized on the basis of tiles (*tile-based platform*). Each tile corresponds to a processor core associated with local memory for instructions and data. This organization leads to avoiding any interference between tiles during the execution of computations. Each tile also has a communication interface for access to the shared communication bus. Communications between tiles are made through a shared memory. The allocation of actors on the tiles is established once the programs are compiled and is not modified during execution. The execution of the application is done according to the dependencies established between the actors and taking into account the availability of shared resources. Such an organization is fully compositional:

modifying the allocation of the application and using more tiles does not modify the computation times measured for each actor for a given configuration. The characterization made of the computation times is therefore valid for different organizations of the platform and different allocations of the application.

The measurement infrastructure includes the resources needed to record the computation times of the actors on the tiles as well as the communication times between actors. For the computation time measurement, we used the device developed within the OFFIS institute, presented in [67], in order to measure the duration required between the start and the end of a phase of calculation of an actor. This device was also used to measure the end-to-end latency of complete applications. For the measurement of the communication times, we used a logic analyzer device allowing the observation of the signals of the communication bus. These observations made it possible to record the start and end times of the various phases of communication between tiles.

Part (b) of the flow corresponds to the modelling phase which aim at obtaining a simulable model. This phase uses as input the description of the application and the platform as well as the measurement results related to the execution of an application on the resources of a platform. The upper part of Figure 5.2 details the process leading to the modelling of computation resources.

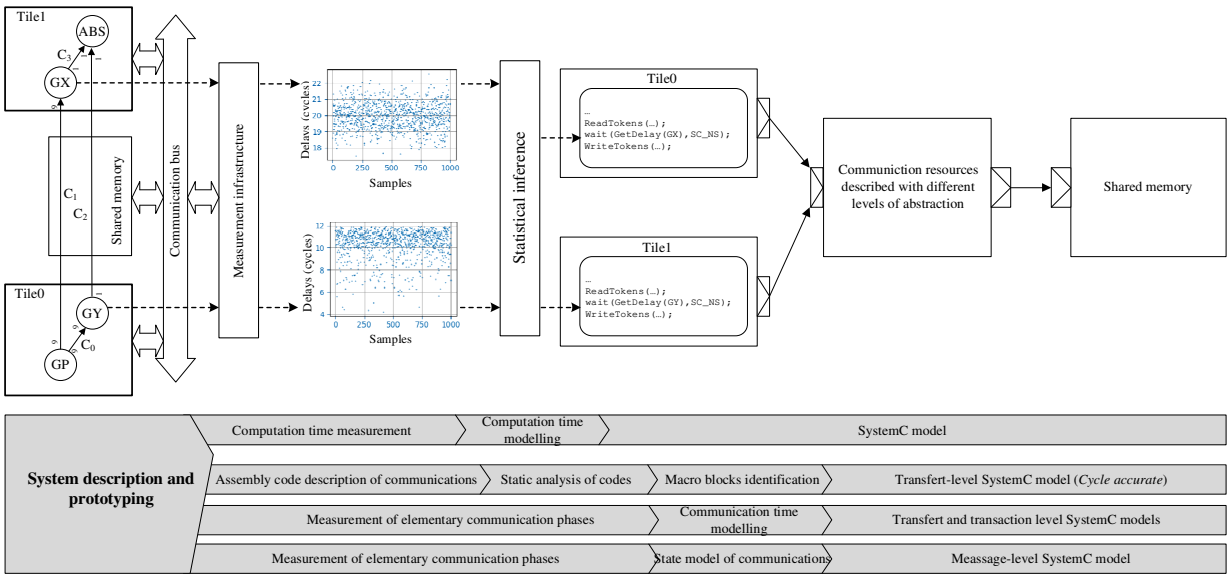


Figure 5.2: Process of characterization and probabilistic modelling of computation and communication resources.

Once the prototyping phase on a real target has been carried out, the studied application is executed on the implemented platform. This execution makes it possible to record the computation durations associated with each actor of the application and this for multiple iterations of the application. Each iteration corresponds to the processing of distinct data, they can therefore make it possible to identify distinct paths within the executed program. The durations thus recorded for each actor are then analyzed in order to establish a representation in the form of a random variable according to a given law of probability. In this chapter, we will present the application of Gaussian laws to describe these variations. The combination of several Gaussian distributions was also considered in [J5-1] using the inference method *Kernel Density Estimation* (KDE) [68]. The right part of Figure 5.2 illustrates the organization of the SystemC model created. The *Tile0* and *Tile1* modules describe the execution of the actors allocated on the computation resources. The behaviour of the actors is described taking into account the durations measured. The function noted `GetDelay` allows to set the computation duration according to the distribution resulting from the measurement data.

The module denoted *Interconnect* models tile accesses to shared memory. The functions *ReadTokens* and *WriteTokens* implement the communication protocol for the transfer of tokens through shared memory. This protocol is the same as the one previously described in Chapter 3, Figure 3.6. The *Shared memory* module models the effect of read and write accesses to memory during communications. The lower part of

Figure 5.2 illustrates three approaches successively considered in our work in order to model communication mechanisms:

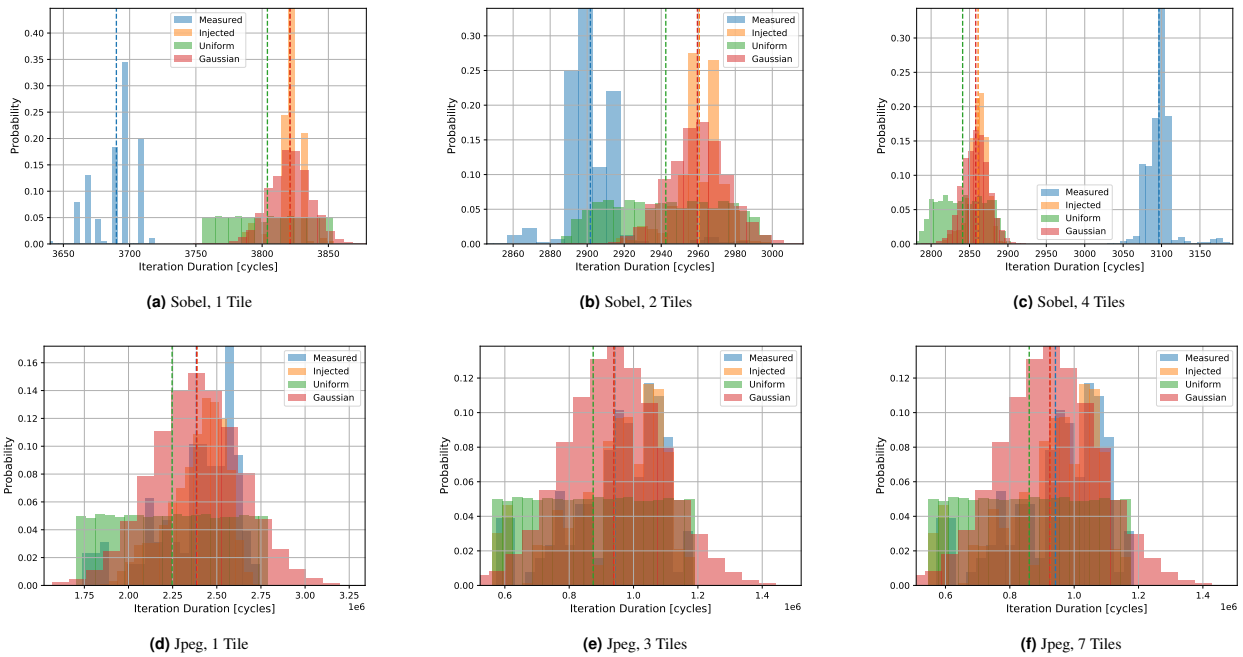
- The first approach, presented in [C5-3], corresponds to a static analysis at the assembler level of the code developed to implement the functions *ReadTokens* and *WriteTokens*. The instructions associated with each phase of the communications are identified and the durations of each phase are thus estimated. These durations were used within a transfer level model.
- The second approach, presented in [C5-2], uses the measurement infrastructure to record the durations of the different elementary phases of communication. These readings were then used in transfer and transaction level models as presented in Chapter 3.
- Finally, the third approach, presented in [C3-4], [J5-1], also uses the measurement infrastructure to record the durations of the different elementary phases of communication and this in connection with the message-level simulable model proposed in Chapter 3.

Part (c) of Figure 5.1 corresponds to the analysis phase of the flow, consisting in carrying out the simulation of the models created. The property analyzed was essentially the end-to-end latency of applications. The simulation results are then compared to the measurements made on the real target. It was then possible for us to quantify the level of precision achieved by the created models as well as to note the required simulation duration. The assessment of the established flow was carried out for different applications and different allocations of applications on the resources of the platform.

Different approaches exist to calibrate performance models of multiprocessor architectures. In [34], three families of approaches are presented. The first consists of establishing an analytical model to quantify the execution time of an algorithm, taking into account parameters such as the number of elementary operations to be performed, the number of memory accesses or the size of the manipulated data. The second approach consists in statically analyzing the code of the algorithm considered in order to extract the possible execution durations. Finally, a third presented approach consists in extracting information directly from executions on a real target. In [69], an approach is presented for the calibration of models created within the framework of the Sesame environment [30]. The elementary computation and communication times come from simulations made using instruction set simulators for the programmable processor part or simulations at the register transfer level for the dedicated hardware resources. The approach presented in [39] associates a simulation at a high level of abstraction of multiprocessor architectures with a stochastic model making it possible to deduce the effect of contention to shared resources on the execution of the software. This stochastic model is established from preliminary tests of reference programs on detailed descriptions of the resources studied. The creation of probabilistic models of multiprocessor systems from measurements has also been adopted by Nouri et al. [70] to extend the BIP modeling environment [71]. This approach consists in executing the entire application on the real target in order to establish a representation of the execution times of each actor composing the application. A detailed presentation of these approaches is given in the research report [R1-1].

5.2.2 Results

The purpose of the experiments carried out was to validate the relevance of the established modelling and simulation flow, both for the construction of computation and communication models. First, the accuracy of the created models was analyzed by comparing the estimates obtained for the end-to-end latency of the applications with the measurements made on a real target. The simulation duration of the created models was also analyzed. The applications and the platform considered for these experiments are the same as those presented in Chapter 3, Section 3.3. These systems were realized on a Xilinx ZC702 prototyping board based on Zynq-7000 FPGA. For these experiments, the models of the studied systems were created according to the process indicated in Figure 5.2. Computation times were measured by running 1 000 000 iterations of each application on a single platform processor. These measurements were then analyzed in order to establish the distribution of execution times for each actor. The different communication models created were considered and a complete comparison of the results was presented in [J5-1]. In this part, we will present the main results obtained for the transfer-level communication models annotated from measurements, as mentioned in Chapter 3.



(a) Sobel, 1 Tile **(b)** Sobel, 2 Tiles **(c)** Sobel, 4 Tiles

(d) Jpeg, 1 Tile **(e)** Jpeg, 3 Tiles **(f)** Jpeg, 7 Tiles

Figure 5.3: Distribution of the measured data (blue) compared to the results otained for models differing according to the modelling of the computation times. Computation times are described by injected data (orange) or a Gaussian distribution (blue). (a), (b) and (c) show the results for the Sobel filter application. (d), (e) and (f) show the results for the JPEG decoder application. Dotted horizontal lines indicate average execution times [47].

Figure 5.3, taken from [47], presents the distribution of end-to-end latency values of the studied systems obtained by measurement and by simulation. The upper part of the figure corresponds to the results obtained for three distinct allocations of the Sobel filter application. The lower part corresponds to the results obtained for three allocations of the JPEG decoder application. The curves presented illustrate the distribution of the values observed for the iteration latencies. These values were obtained by measurement on a real target or by simulation. Two types of models are considered here according to the representation of the computation times. The curves presented in blue correspond to the values obtained by measuring the iteration latencies on a real target. The curves presented in red correspond to the values obtained by simulation of models using a Gaussian law in order to describe the distribution of the computation times of the actors. Finally, the dotted vertical lines indicate the mean values associated with each set of values. For all the studied configurations, the achieved level of accuracy in predicting mean values is acceptable: a difference between simulation and measurement of 7.7% for the configuration *Sobel4* and 1.75% for the configuration *Jpeg7*. As mentioned in chapter 3, we were able to further improve the level of precision by using the message-level communication model. This precision can also be estimated taking into account the shape of the distributions. In [J5-1] the so-called Bhattacharyya distance criterion [72] is used to quantify the similarity between the distributions obtained. According to this criterion, the KDE approach is more efficient for applications with a large number of possible data paths.

With regard to the analysis duration criterion, we compared the simulation durations of the different models with the time required to execute the applications studied on the real target. Thus, the execution of the Sobel filter application, for 1 000 000 iterations, on four tiles takes just over 7 minutes[1]. Running the Jpeg decoder application, for 1 000 000 iterations, on seven tiles takes just over 5 hours. The use of simulable models, at the transfer level, leads to executions of the order of 12 seconds for the Sobel filter on 4 tiles and 2 hours for the Jpeg decoder on 7 tiles [47]. In Chapter 3, we also observed how the message-level model proposed to describe communications allowed a significant reduction in simulation durations (Table 3.3).

The various experiments carried out have demonstrated the effectiveness of the models created. Given the assumptions made, these models can be used to analyze different allocations of the application on the

[1] The results mentioned here are detailed in [47]. They use the simulation infrastructure offered by the intensive computing center of Pays de la Loire [73].

platform. The variability of the execution times of the applications is essentially found within the computation times, taking into account the data dependencies of the algorithms studied. The influence of shared resources was correctly described through the models of communication resources gradually put in place. The creation of effective probabilistic models therefore opens the way to the use of advanced methods to control their simulation. This work is the subject of the next section.

5.3 Adoption of statistical model-checking analysis methods

5.3.1 Proposal

The work mentioned so far dealt with the analysis of the behaviour of systems analyzed through simulation. An intrinsic limit of this analysis approach relates to the limited coverage of operating situations. Our objective here is to allow a more successful simulation analysis, by offering the possibility to the designer to control the simulation effort given a level of confidence in the analysis carried out. To do this, the so-called statistical model-checking approach (*statistical model checking*, SMC [74]–[76]) is considered. This approach is based on the simulation of a probabilistic system, subsequently denoted S. A simulation of S generates an execution trace corresponding to a sequence of states from S. Using the probabilities inherent in S, the execution traces are generated randomly. They therefore constitute representative samples making it possible to estimate the probability according to which an execution trace satisfies a given property, noted φ. Two types of algorithms make it possible to estimate either the probability with which S satisfies φ (*quantitative analysis*), or whether the probability with which S satisfies φ is greater than a given bound θ (*qualitative analysis*). Such analyzes can be useful in the process of sizing resources and verifying non-functional properties.

The algorithms used to conduct a quantitative analysis come from Monte-Carlo techniques [77], [78]. Let γ be the probability with which S satisfies φ and γ_N the estimate obtained for N samples. It is possible to bound the precision of γ_N and the probability that an error will occur [79]. We can then calculate the minimum number of samples that guarantee a given precision, noted δ and therefore such that $|\gamma_N - \gamma| \leq \delta$. We introduce α, called the confidence parameter, such as $Pr(|\gamma_N - \gamma| \leq \delta) \geq 1 - \alpha$. In [78], [80], it is shown that considering $N \geq \frac{4}{\delta^2} ln(\frac{2}{\alpha})$ then it is guaranteed that $Pr(|\gamma_N - \gamma| \leq \delta) \geq 1 - \alpha$. Therefore, it is possible to estimate the probability that S satisfies φ with a precision δ and a confidence level $1 - \alpha$ by averaging the number of simulations satisfying φ with the number of total simulations.

The approach used to conduct a qualitative analysis derives from the so-called hypothesis testing technique (*hypothesis testing*). In order to establish whether γ is greater than a limit θ, it is necessary to compare the hypothesis H_0 such that $\gamma \geq \delta$ with the hypothesis H_1 such that $\gamma < \delta$. The strength of such a test is determined by the parameters α and β respectively bounding the probabilities that H_0 is accepted while H_1 is true (type I error) and that H_1 is accepted while H_0 is true (Type II error). However, it is impossible to ensure a low probability for both types of error. A symmetric indifference region denoted $[\gamma_1, \gamma_0]$ is then used, centered on θ and for which $p_1 = \theta - \delta$ and $p_0 = \theta + \delta$. The SPRT algorithm (*sequential probability ratio test*) presented in [81] makes it possible to perform this hypothesis test.

These algorithms have been implemented in various probabilistic model analysis tools, including UPPAAL [82] and PRISM [83]. Also, the SMC approach has been applied to different types of [84] systems. Compared to the formal methods applied to probabilistic models, known as probabilistic model-checking (*probabilistic model checking*, PMC), SMC does not allow exact quantification of the probability that the given property is satisfied. However, SMC methods offer strict guarantees on the accuracy of the γ_N estimates. Also, the approximation made is obtained with a lower analysis time compared to full resolution methods, SMC not suffering from the state space explosion problem inherent to model-checking in general. The application of the SMC method to the analysis of the temporal properties of multiprocessor systems is recent, in particular through the work presented by Nouri et al. [85]–[87]. In [86], the application of SMC is presented for the qualitative analysis of the temporal properties of an image processing application running on a multicore platform. The SPRT method was then used within the SBIP [87] analysis tool. In [88], the UPPAAL-SMC tool is used to jointly evaluate the timing and power properties of allocated applications on multiprocessor platforms. This same tool was used in [89] to conduct an analysis similar to the one we conducted. This work illustrates the use of the UPPAAL-SMC formalism and the difficulties in modeling the different aspects

of the systems studied. In [90], the use of SMC for the analysis of SystemC models is considered for a control-command system distributed on three processors. First, our approach differs from these works by the fact to establish a scalable approach, with the objective to favour the analysis of different possible applications and allocations. This objective therefore implies specific hypotheses on the calculation and architecture models used. Secondly, we adopted the SMC approach to the case of SystemC models of multiprocessor systems, in order to have a greater flexibility in comparison with tools like UPPAAL-SMC. Finally, the experiments carried out on different applications, platforms and possible allocations have made it possible to evaluate the effectiveness of the SMC method with regard to the criteria of accuracy and simulation speed. The modeling flow previously presented has been extended to allow the implementation of SMC according to the principle presented in Figure 5.4.

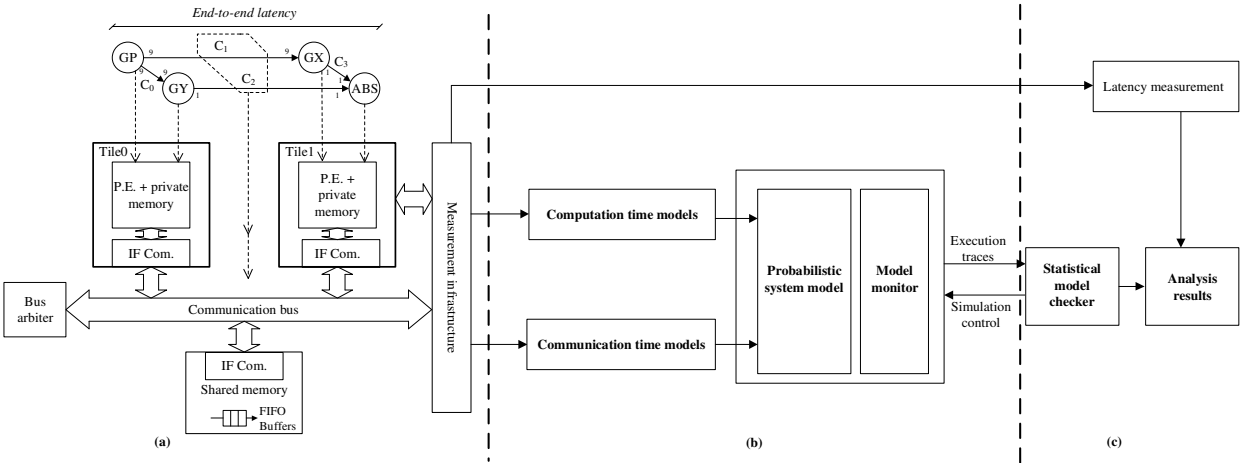


Figure 5.4: Extended modelling flow to allow application of SMC methods.

The flow considers a probabilistic model created by following the approach previously described. A monitor is added to observe the variables of interest of the model during the simulation. For each observed variable, the monitor delivers an execution trace of the form: $\omega = (s_0,t_0),(s_1,t_1),...,(s_{N-1},t_{N-1})$, $N \in \mathbb{N}$, where each couple (s_i,t_i) indicates a particular state s_i and an instant $t_i \in \mathbb{R}_{\geq 0}$. The generated execution traces are analyzed during the simulation by the SMC analysis tool in order to establish whether the analyzed properties are respected or not. The analysis tool successively triggers the simulations of the model, the number of simulations depending on the statistical algorithms used and their specific parameters. Within the framework of our work, we more particularly considered the use of the Plasma-lab analysis environment [91], as it allows the analysis of models written in SystemC [90].

Figure 5.5 details the creation of the analyzed SystemC model as well as its control by the SMC analysis tool. This flow considers as input a probabilistic model written in SystemC and created by following the

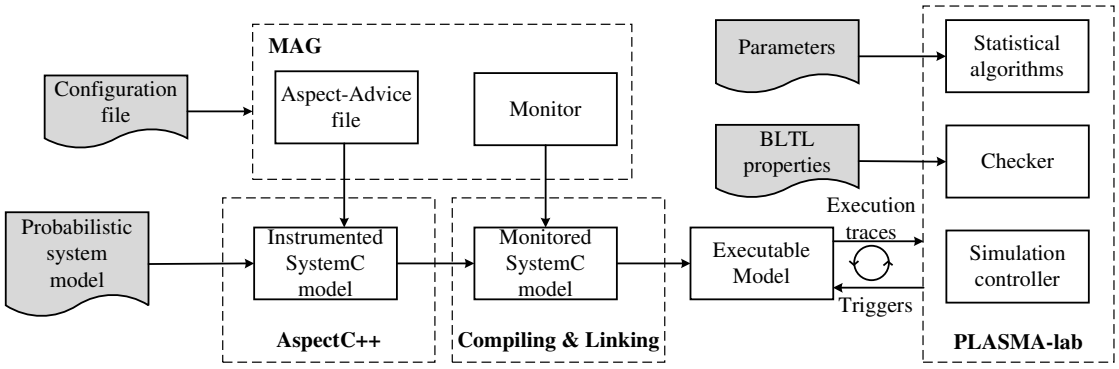


Figure 5.5: Details about the generation and simulation flow of instrumented and monitored SystemC models.

process previously described in this chapter. The creation of a monitor is operated from the definition of

the variables to be observed within the probabilistic model. The SystemC description of this monitor is obtained using the *Monitor and Aspect-advice Generator* (MAG) tool described in [90]. The instrumentation of SystemC models aims to allow communication between these models and the generation of execution traces [92]. This instrumentation is operated automatically using the AspectC++ tool. The obtained description can then be compiled in order to have an executable model that can be interfaced with the Plasma-lab environment.

In the considered approach, the properties are expressed in the so-called *Bounded Linear Temporal Logic* (BLTL) language. This language is an extension of the *Linear Temporal Logic* (LTL) language considering bounded temporal operators. These properties apply to finite execution traces, taking into account the fact that the simulation itself is of finite duration. These properties will be used to conduct both quantitative and qualitative analyzes on the created models. In the considered case studies, we have expressed properties in this language in order to estimate the average execution latency of applications.

5.3.2 Results

The experiments carried out aimed to evaluate the contribution of SMC techniques to the simulation-based analysis of the temporal properties of multiprocessor architectures. These experiments extend the studies previously described in this chapter. The applications considered correspond to those illustrated in Figure 3.10, that is to say the Sobel filter and the JPEG decoder. Two platform configurations were studied. The first configuration corresponds to the fully compositional configuration considered in the previous section. The objective was to evaluate the contribution of the SMC method and its influence on the precision delivered and on the duration of the analysis. A second platform configuration, non-compositional this time, was considered in order to consider the possible extension of the developed approach. This second configuration considers a global DDR memory, shared by all the tiles of the platform and external to the FPGA used for prototyping. In addition, within each tile, the data and instruction caches are activated, thus ensuring exchanges with the global memory. This configuration corresponds to an organization that is more representative of current multiprocessor architectures and allows larger program sizes than for the first configuration. This configuration however implies having to take into consideration the effects of the caches and of the accesses to the global memory during the calculations carried out within each processor as well as during the execution of the communications between processors. These effects therefore imply possible contention during the communication and calculation phases. Furthermore, for the two configurations mentioned, a shared memory is dedicated to communications between tiles.

The creation of probabilistic models is carried out following the process previously used. The computation times associated with each actor were estimated as following Gaussian laws whose parameters (mean value and standard deviation) were established from the taken measurements. The message-level communication model presented in chapter 3 was used. For the second configuration of the platform, the measured durations of the elementary communication phases illustrated the influence of the new resources used. These durations were described in the form of random variables with value following a uniform law between the minimum and maximum values measured. Details of the quantities resulting from the characterization phase are given in [47] and [C5-4].

Subsequently, we present the results obtained for a quantitative analysis of the models created. This analysis aimed to estimate the average duration of the iterations of the studied applications, allocated on different resources of the platforms considered. To do this, the properties were established in order to estimate the probability that this duration is included in a given interval, this estimate being repeated for different time intervals.

First configuration of the platform

The first criterion addressed relates to the precision of the estimates obtained by adopting the SMC approach. Figure 5.6 illustrates the distribution of the estimates obtained for different intervals of time in the case of the Sobel filter distributed over 4 tiles. These estimates lead to the establishment of the estimated average value. Table 5.7 presents the average measured values of the execution times of the considered applications for different allocations on the platform and for a total of 1 000 000 iterations. These values are to be compared

with the values obtained in simulation for 1 000 000 iterations. The last column presents the average values obtained for controlled simulations according to the SMC approach. The SMC method used corresponds to the Monte-Carlo method with the parameters $\delta = 0.02$ and $1 - \alpha = 0.98$. In this case, only 5757 successive iterations are carried out. The results obtained by using the SMC approach show a satisfactory level of accuracy compared to an intensive simulation involving a higher number of iterations.

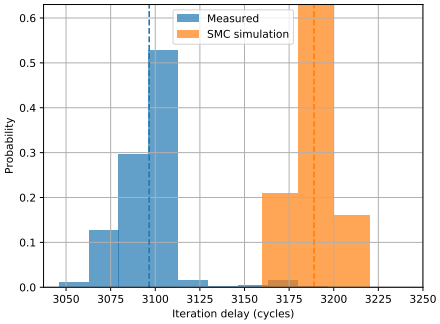


Figure 5.6: Distribution of measured data and estimates obtained by simulation for the experiment *Sobel4a* [47].

Exp.	Measure	1M simulation runs	SMC simulation
Sobel1a	3690	3797.5 (2.91 %)	3799.2 (2.96 %)
Sobel2a	2902.5	2977.5 (2.58 %)	2977.8 (2.58 %)
Sobel4a	3097.4	3194.5 (3.13 %)	3189.0 (2.96 %)
Jpeg1a	2385860	2382722.5 (-0.13 %)	2376010.1 (-0.42 %)
Jpeg3a	940836	912219.5 (-3.05 %)	911764.7 (-3.1 %)
Jpeg7a	941059	896367.7 (-4.75 %)	886538.5 (-5.8 %)

Figure 5.7: Average values expressed in cyces, obtained by measurement on a real target, simulation of 1 000 000 iterations of a probabilistic model and simulation controlled by the SMC approach. The values given in parentheses indicate the deviation from the mean value measured for the first configuration of the platform [47].

We were also able to observe the significant influence of the parameter α on the number of iterations required and on the level of precision reached [47], [C5-4]. It is thus possible to control the simulation effort with regard to the desired level of confidence.

However, the use of the SMC method involves an additional cost in terms of model analysis time. Thus, the simulation of 5757 iterations of the model for the Sobel4 configuration has a duration of 1 minute and 44 seconds, to be compared to 5 seconds for the simulation of 1 000 000 of iterations without SMC method[2]. For the configuration *Jpeg7a* the analysis time is 2 minutes and 16 seconds compared to 29 seconds for the simulation of 1 000 000 iterations without SMC method. This observation underlines the influence of the addition of the SMC method to the simulation of probabilistic models. This influence comes from the need for the Plasma-lab tool to note the evolution of the variables observed at each advancement in the SystemC simulation time. However, the analysis times remain satisfactory and such results validate the possible use of the SMC method for the evaluation and comparison of the performances of different possible architectures.

Second configuration of the platform

For this second platform configuration, only the Sobel filter application was considered, given the excessive execution times in the case of the Jpeg decoder. Figure 5.8 illustrates the distribution of the estimates obtained for different time intervals in the case of the Sobel filter distributed over 4 tiles. Table 5.9 presents the average values observed by the measurement on real target and by simulation. The results presented in the second column correspond to the average values measured for 10 000 000 iterations of the applications. The third column corresponds to the estimated value after the simulation of 1 000 000 iterations of the considered probabilistic model. Finally, the fourth column presents the average values resulting from the simulation of this model using the SMC method. The results presented are obtained for the simulation of 5757 iterations of the application. We always observe a very good level of precision for a lower number of simulated iterations. The simulation times noted for these experiments are of the order of 5 seconds for the simulation of 1 000 000 iterations of the probabilistic model created and of the order of 1 minute and 40 seconds for the simulation of 5757 iterations of the same model controlled by the SMC approach.

This second configuration allowed us to tackle a platform with a two-level memory hierarchy. The experiments carried out have shown the validity of the approach with, however, limits due to the non-compositionality of the platform. Thus, it would be appropriate to approach the modelling of the mechanisms of cache or DDR memory in order to limit then the effort of characterization of the resources of calculation

[2] Simulations carried out on an Intel core i7 2.5GHz machine, 8GBytes of RAM, under Ubuntu.

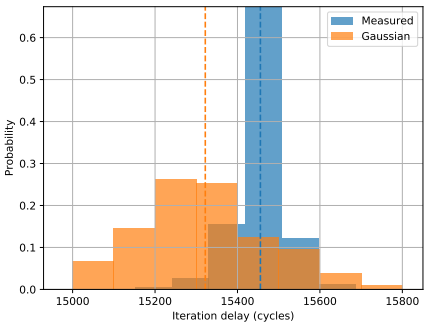


Figure 5.8: Distribution of measured data and estimates obtained by simulation for the experiment *Sobel4b* [47].

Exp.	Mesure	Simulation 1M	Simulation SMC
Sobel1b	20634.2	20547.5 (-0.42 %)	20529.2 (-0.51 %)
Sobel2b	15533.8	15459.7 (-0.48 %)	15479.0 (-0.35 %)
Sobel4b	15455.1	14907.7 (-3.55 %)	15321.9 (-0.87 %)

Figure 5.9: Average values expressed in cyces, obtained by measurement on a real target, simulation of 1 000 000 iterations of a probabilistic model and simulation controlled by the SMC approach. The values given in parentheses indicate the deviation from the mean value measured for the second configuration of the platform [47].

and communication. Moreover, through this second configuration, we were also able to observe the relevance of the proposed message-level communication model, even in the case of variations in the durations of the elementary communication phases.

5.4 Conclusion

The work carried out on this theme has led to the establishment of a complete modelling flow for the analysis of the temporal properties of multiprocessor systems. The effectiveness was illustrated through various case studies implemented on a real target. The contributions made are methodological and experimental. The efficiency of the proposed flow is based on the strict separation between communications and calculations, separation established at the level of the application and the execution platform. The extension of this flow is however necessary to address more advanced platforms that do not guarantee this separation. Moreover, it would be interesting to extend the proposed approach in order to allow the modelling and the analysis of the energy consumed for multiprocessor embedded systems. This would make possible a joint optimization of performance and power consumption. These aspects will be addressed in particular through the thesis of Mr. Quentin Dariol, started in September 2020, about to the modelling and optimization of applications based on neural networks under time and energy constraints.

From an organizational point of view, beyond the supervision of Mr. Hai-Dang Vu's thesis work, this activity has enabled the establishment of fruitful exchanges with a research team from the OFFIS institute in Germany. The quality of this collaboration is based on the understanding and integration of the respective skills and interests of each partner. In addition, this collaboration has led us to pay particular attention to the dissemination and reproducibility of experiments, through specific tools (github, zenodo). Finally, this research project has led to simultaneous modelling and prototyping on a real target, which represents substantial efforts in the context of doctoral theses. In order to effectively pursue this approach, we will therefore need to be able to include it in projects allowing us to provide the technical support necessary for carrying out the experiments in connection with the research work.

5.5 Supervision

PhD thesis

- Hai-Dang VU, *Fast and accurate performance models for probabilistic timing analysis of SDFGs on MPSoCs*, University of Nantes, March 2021.

Master thesis

- Jiatong Li, *Extraction of stochastic models for computation and communication time on a multicore Zynq platform*, University of Nantes, June 2017.

- Nadia Ghazali, *Execution trace analysis methods for performance modelling of embedded system architectures*, University of Nantes, June 2015.

5.6 Publications related to Chapter 5

International journals with program committee

[J5-1] R. Stemmer, H.-D. Vu, S. Le Nours, K. Grüttner, S. Pillement, and W. Nebel, "A measurement-based message-level timing prediction approach for data-dependent sdfgs on tile-based heterogeneous mpsocs," *Applied Sciences*, vol. 11, no. 14, 2021, ISSN: 2076-3417. DOI: 10.3390/app11146649. [Online]. Available: https://www.mdpi.com/2076-3417/11/14/6649.

International conferences with program committee

[C5-1] J. Li, S. Le Nours, X. He, and Z. Jing, "A statistical characterization approach to estimate software execution time in multiprocessor systems," in *2017 International Conference on Computer Technology, Electronics and Communication (ICCTEC)*, 2017, pp. 848–852. DOI: 10.1109/ICCTEC.2017.00188.

[C5-2] R. Stemmer, H.-D. Vu, K. Grüttner, S. Le Nours, W. Nebel, and S. Pillement, "Towards Probabilistic Timing Analysis for SDFGs on Tile Based Heterogeneous MPSoCs," in *10th European Congress on Embedded Real Time Software and Systems (ERTS 2020)*, ser. ERTS 2020, Toulouse, France, Jan. 2020, #paper 59. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02305048.

[C5-3] ——, "Experimental Evaluation of Probabilistic Execution-Time Modeling and Analysis Methods for SDF Applications on MPSoCs," in *International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS XIX)*, Samos, Greece, Jul. 2019, paper #28. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02115121.

[C5-4] H.-D. Vu, S. Le Nours, and S. Pillement, "Experimental Evaluation of Statistical Model Checking Methods for Probabilistic Timing Analysis of Multiprocessor Systems," in *Euromicro Conference on Digital System Design (DSD 21)*, Palermo, Italy, Sep. 2021.

Chapter 6

Conclusion and prospects

6.1 Review of the work carried out

The work presented in this document falls within the field of the design of hardware-software architectures of embedded systems. This work is motivated by the need to control the design complexity of these systems. The definition of reliable high-level abstraction models thus represents an essential issue in order to support the analysis and the dimensioning of the architectures. The contributions brought within the framework of my works go in the direction of the improvement of the effectiveness of the models used for the simulation-based analysis of the architectures. Various case studies have made it possible to refine and demonstrate the relevance of the contributions made. For certain studies, we could compare with measurements on real targets. Different research directions have been drawn up in order to continue to consolidate and enhance this work.

A first axis of study identified in Chapter 1 of this document concerned the control of the simulation effort. Thus, the hybrid approach presented makes it possible to limit the number of calls to the simulation engine while preserving the precision of the models created. Through various co-supervised theses, the case studies carried out have made it possible to gradually enrich the approach and to quantify the gains obtained in terms of precision and simulation speed. In parallel, a systematic approach for the creation of models was proposed in order to consider an automation of the developed approach. In the medium term, it would be interesting to apply the proposed approach to new case studies addressing other advanced mechanisms of the operation of multiprocessor systems as well as developing the tools required to support such an approach.

A second area of study focused on controlling the modelling effort required in order to establish models that deliver a good compromise between analysis speed and accuracy. In this sense, the proposal of a generic execution model represents a guide for the creation of efficient high-level abstraction models. This model was completed by associating an approach based on measurement. The experiment carried out around the modelling of the ARM AXI4-Lite multiprocessor bus protocol has made it possible to illustrate the interest of this combination. Thus, probabilistic modelling based on measurement represents an approach with great potential in order to be able to understand complex architectures for which shared resources significantly influence the behaviour of applications. The experimental approach put in place could thus be extended to be able to consider such phenomena. This approach could also be extended to take into account the induced power consumption, opening the way to modelling, analysis and optimization at the system level of the energy consumed by such architectures.

Finally, on the axis of online management of multiprocessor architectures, a first set of proposals focused on the definition of methods for the modelling and simulation of such management within an industrial software environment. Subsequently, some allocation and scheduling strategies have been proposed in order to improve the energy efficiency of multiprocessor systems. It would be interesting to consider the application of this work in the case of the experimental platform set up through the PETA-MC project, thus making it possible to refine and validate the proposals made and the models established. Also, taking communication and storage resources into account would be necessary in order to optimize the proposed allocation strategies.

Beyond this scientific and technical assessment, some general observations can also be made.

In this document, the presentation made of the work particularly focuses on describing a modelling and simulation approach by first addressing the principles of this approach and then its application to different case studies. This work has been gradually built up over time in parallel with the co-supervised theses and the collaborative projects carried out. Such a presentation makes it possible to underline the mutual contributions between projects carried out over limited periods and more in-depth work. In the longer term, it seems useful to me to be able to continue with such an approach in addition to research by project.

On another level, the presented research work is particularly consistent with the teaching activities that I carry out. The interaction between these two facets is gradually proving fruitful: setting up student projects or practical work addressing aspects developed in research, enrichment of courses given in Master program, student internships with research partners, support of students trained in the engineering cycle towards research. In addition, the supervision of thesis students, and training in research through research, is a particularly enriching experience from the point of view of the teacher and researcher. I was able to co-supervise students of different backgrounds, nationalities and sensitivities. This supervision requires supporting students on research topics while ensuring the development of their own sensitivity.

Finally, the work carried out presupposes a large part of experimentation on complex targets. This can represent a significant effort in the context of theses that do not fit into an advanced collaborative context. Therefore, the establishment of CIFRE conventions[1] seems to me an important objective in order to establish closer collaborations with companies in the field. The achievements put in place will be able to convince of the relevance of the work carried out.

6.2 Prospects and directions envisaged

Current environmental issues tend to consider an increasing digitization of our societies. The proliferation of embedded systems should therefore continue in many areas, with strong constraints in terms of computation efficiency, power consumption, security and reliability. Mastering the complexity of future hardware-software architectures will therefore remain a key challenge. Also, it is important to consider solutions to limit the carbon footprint of such a digital transition and this for all phases of a product's life cycle: design optimization of hardware and software resources, operational functioning with the optimization of energy consumption and the lifespan of products, the recycling of these systems. The planned research project fits into this context by relying on the work previously carried out and by opening up new directions.

First, the thesis of Mr. Quentin Dariol, started in September 2020, addresses the prediction of the performance and energy of multiprocessor systems for applications based on neural networks. This thesis work, in collaboration with the OFFIS institute in Germany, is based on the prototyping and modeling environment previously set up as part of the PETA-MC project. Ultimately, this work should address the modeling and analysis of such architectures in order to optimize their performance and energy efficiency. The expected contributions go in the direction of the definition of performance and energy models of multiprocessor architectures for neural networks, with the development of a measurement infrastructure adapted to take into account the consumption of architectures. This thesis therefore represents an opportunity to consolidate and refine our work for new types of applications and constraints.

Also, in this field of the implementation of artificial intelligence algorithms for embedded systems, exchanges have been initiated with certain members of the Images department of the IETR in order to study the case of so-called dynamic neural networks. Such algorithms have strong dependencies on the processed data and presents adaptability of the network architectures [93]. Given the irregularities in their flow of execution, these algorithms prove to be inefficient for executions on graphic processors (GPU). The study and proposal of hardware-software architectures adapted to this type of network are the subject of an exploratory project supported internally by the IETR laboratory. This project will be carried out in 2022 through two supervised Master's level internships within the two teams involved.

The continuation of the collaborations initiated with the OFFIS institute[2] is currently envisaged according

[1] In France, CIFRE convention is an industrial agreement for training through research between a company, a laboratory and a thesis student. [2] On January 1, 2022, the Transport department of the OFFIS institute with which we collaborate was integrated into the DLR, the German center for aeronautics and astronautics. Collaborations will therefore continue with this centre.

to two themes. The first theme deals with the analysis of temporal properties for non-compositional multiprocessor platforms, for which shared resources significantly influence the execution of programs. Typically, platforms based on shared cache memory can be analyzed. The experimental approach set up within the framework of the PETA-MC project offers the possibility to estimate the worst case execution time in a probabilistic way (pWCET, *probabilistic worst case execution time*). The adoption of methods based on the EVT approach (*Extreme Value Theory*) would make it possible to establish a distribution of the pWCET from the execution times measured [94]. However, the application of this approach raises the question of the representativeness of the processed data and the measurement protocol considered so far in our approach should be adapted. The perspectives of such works would be to allow a reliable prediction of worst case execution times of data flow oriented applications for complex multiprocessor platforms. The second topic envisaged aims to address the study and optimization of neural networks for multiprocessor platforms under time, power consumption and reliability constraints. The aim is to define static and dynamic optimization methods for the allocation and scheduling of neural networks distributed over several computers. Computer overload or failure situations would be considered. On-line management of the resources would allow the adaptation of the distribution of the computations taking into account the constraints imposed on the complete system. For the two themes mentioned, experiments on real prototypes are planned in order to compare the models created.

These themes highlight two basic trends in the field of research:

- the integration of new artificial intelligence algorithms within embedded systems implies new use cases that should be taken into account and modelled;
- the ever-increasing complexity of platforms requires the development of original methods to favour the creation of efficient models for performance estimation.

Thus, in the medium term, it seems appropriate to me to approach these developments according to two complementary axes: the integration under constraints of artificial intelligence algorithms and the use of artificial intelligence for the system-level design of architectures.

On this first line of work, the integration of artificial intelligence algorithms within embedded systems imposes new needs within execution platforms. In order to dimension architectures under constraints, it will be necessary to consider more specifically the influence of the storage resources on the temporal behaviour of the architectures. The models proposed so far will have to be extended in order to allow a dimensioning taking into account the placement in memory of the data of the algorithms (for example, the parameters of neural networks) and the characteristics of the memories considered. The hybrid approach implemented, combining simulable models, calculations and characterization by measurement, can be used to provide reliable models of the influence of storage resources. Using the models then created, it will be possible to explore the design space with the aim of proposing implementations of artificial intelligence algorithms optimized under the constraints of available resources, performance and consumption. This work could be carried out in collaboration with some of the national and international teams addressing different aspects to optimize the hardware-software architectures of embedded systems for artificial intelligence. We are thinking here first of all of the methods of automatic neural architecture search (*neural architecture search*, NAS) which could be completed using the models developed. Secondly, if the platform developed within the framework of the PETA-MC project can be initially considered, the application of this work to the case of platforms built around RISC-V technology would be interesting. This technology offers many perspectives for optimizing hardware-software architectures under performance and energy constraints. It also receives strong interest from industry. Finally, this study may lead us to take an interest in other forms of calculation, which are gradually emerging. One thinks in particular of the case of neuromorphic computing, some implementations of which are beginning to emerge in the industrial environment (for example, the Loihi 2 architecture presented by Intel in 2021). Thus, the integration of these new forms of calculation in interaction with more traditional architectures and the joint dimensioning of resources opens the way to the proposal of new methods of analysis and dimensioning [95].

On the second line of work mentioned, we see that artificial intelligence methods benefit many engineering fields. In the field of the design of embedded systems, such methods could in particular promote the creation of reliable high-level abstraction models based on prior observations. One example is the use of learning

methods to help predict program performance. Such methods, notably described in [96] and [97], consider the prior training of a performance model from a set of programs executed on a given target. The trained model is then used to predict the performance of new programs. An extension of this work has recently been proposed for multiprocessor platforms [98]. The platform and the tools set up within the framework of the PETA-MC project could be used to study such methods. It would thus be interesting to evaluate the influence of the level of compositionality of the architectures on the learning methods studied. Also, artificial intelligence methods could be considered in order to promote the exploration of the architecture design space and the selection of suitable architectures, given the many constraints to be respected. If these methods are beginning to be considered for the design of hardware resources of integrated circuits [99], their adoption in the design flow of hardware-software architectures of embedded systems represents a particularly interesting perspective. The integration of such methods within design flows would then represent an additional step in the evolution of design practices for electronic systems associating hardware and software resources.

Future developments both within the architectures of embedded systems and in the design practices of these architectures will bring out new needs among industrial players. The training of students in these new technologies therefore appears essential in order to meet these needs. As such, it will be important to associate such training with the ethical issues associated with these new technologies (data security, environmental responsibility, technological sovereignty and independence, neutrality of algorithms). All of these developments and the associated questions represent, both for the teacher and the researcher, particularly rich and stimulating perspectives.

List of publications

International journals with program committee

1. R. Stemmer, H.D. Vu, S. Le Nours, K. Grüttner, S. Pillement, W. Nebel, *A Measurement-Based Message-Level Timing Prediction Approach for Data-Dependent SDFGs on Tile-Based Heterogeneous MPSoCs*, Applied Sciences, 2021, 11, 6649. https://doi.org/10.3390/app11146649.
2. S. Yang, S. Le Nours, M. Mendez Real, S. Pillement, *0-1 ILP-based Run-Time Hierarchical Energy Optimization for Heterogeneous Cluster-based multi/many-core Systems*, Journal of Systems Architecture, 2021, `https://hal.archives-ouvertes.fr/hal-03120210`.
3. S. Le Nours, D. Singh, *A Generic Executable Model for Fast Yet Accurate Contention Simulation in Multiprocessor Systems*, IEEE Embedded Systems Letters, vol. 12, p. 117-120, 2020, `https://hal.archives-ouvertes.fr/hal-02436682`.
4. S. Le Nours, A. Postula, *A Hybrid Simulation Approach for Fast and Accurate Timing Analysis of Multi-Processor Platforms Considering Communication Resources Conflicts*, Journal of Signal Processing Systems, Springer, vol. 90, p. 1667-1685, 2018, `https://hal.archives-ouvertes.fr/hal-01643250`.
5. A. Barreteau, S. Le Nours, O. Pasquier, *A case study of simulation and performance evaluation of a SDR baseband architecture*, Journal of Signal Processing Systems, vol. 73, p. 267-279, 2013.
6. T. Majdoub, S. Le Nours, F. Nouvel, O. Pasquier, *Performance evaluation of an automotive distributed architecture based on a high speed power line communication protocol using a transaction level modeling approach*, Journal of Real-Time Image Processing, vol. 9, p. 281-295, 2013.
7. A. Barreteau, S. Le Nours, O. Pasquier, *A state-based modeling approach for efficient performance evaluation of embedded system architectures at transaction level*, Journal of Electrical and Computer Engineering, vol. 2012, 2012.
8. S. Le Nours, F. Nouvel, J.F. Hélard, *Design and implementation of MC-CDMA systems for future wireless networks*, Eurasip Journal on Applied Signal Processing, vol. 2004, p. 1604-1615, 2004.
9. J.F. Hélard, F. Nouvel, S. Le Nours, *A MC-CDMA system analysis in a software radio context*, Annals of telecommunications, vol. 57, p. 699-720, 2002.

National journal with program committee

1. F. Nouvel, S. Le Nours, *Mise en oeuvre de l'outil SynDex pour la conception et l'implantation de systèmes sur plate-forme hétérogène*, Journal sur l'enseignement des sciences et technologies de l'information et des systèmes, EDP Sciences, 2004, `https://hal.archives-ouvertes.fr/hal-00783570`.

International conferences with program committee

1. Q. Dariol, S. Le Nours, S. Pillement, R. Stemmer, D. Helms, K. Grüttner, *A Hybrid Performance Prediction Approach for Fully-Connected Artificial Neural Networks on Multi-Core Platforms*, International

Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS XXII), Samos, Grèce, July 2022.

2. H.D. Vu, S. Le Nours, S. Pillement, *Experimental Evaluation of Statistical Model Checking Methods for Probabilistic Timing Analysis of Multiprocessor Systems*, Euromicro Conference on Digital System Design (DSD), Palerme, Italy, September, 2021.

3. H.D. Vu, S. Le Nours, S. Pillement, R. Stemmer, K. Grüttner, *A Fast Yet Accurate Message-level Communication Bus Model for Timing Prediction of SDFGs on MPSoC*, Asia and South Pacific Design Automation Conference (ASP-DAC), Tokyo, Japan, January, 2021.

4. R. Stemmer, H.D. Vu, K. Grüttner, S. Le Nours, W. Nebel, S. Pillement, *Towards Probabilistic Timing Analysis for SDFGs on Tile Based Heterogeneous MPSoCs*, 10th European Congress on Embedded Real Time Software and Systems (ERTS 2020), Toulouse, France, January 2020.

5. S. Yang, S. Le Nours, M. Mendez Real, S. Pillement, '*Mapping and Frequency Joint Optimization for Energy Efficient Execution of Multiple Applications on Multicore Systems*, The Conference on Design and Architectures for Signal and Image Processing (DASIP), Montreal, Canada, October, 2019.

6. R. Stemmer, H.D. Vu, K. Grüttner, S. Le Nours, W. Nebel, S. Pillement, *Experimental Evaluation of Probabilistic Execution-Time Modeling and Analysis Methods for SDF Applications on MPSoCs*, International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS XIX), Samos, Greece, July 2019.

7. S. Yang, S. Le Nours, M. Mendez Real, S. Pillement, *System-Level Modeling and Simulation of MPSoC Run-Time Management using Execution Traces Analysis*, International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS XIX), Samos, Grèce, July, 2019.

8. J. Li, S. Le Nours, X. He, Z. Jing, *A Statistical Characterization Approach to Estimate Software Execution Time in Multiprocessor Systems* 2017 International Conference on Computer Technology, Electronics and Communication (ICCTEC), Dalian, China, Décember 2017.

9. S. Le Nours, *Timing correction technique for fast and accurate state-based performance models*, Forum on specification & Design Languages (FDL), Barcelona, Spain, September, 2015.

10. S. Le Nours, A. Postula, N. Bergmann, *A dynamic computation method for fast and accurate performance evaluation of multi-core architectures*, proceedings of Design, Automation and Test in Europe (DATE'14), Dresden, Germany, March, 2014.

11. T. Majdoub, S. Le Nours, O. Pasquier, F. Nouvel, *Application of temporal decoupling to the creation of efficient performance models of automotive architectures*, proceedings of Conference on Design and Architectures for Signal and Image Processing (DASIP'12), Karlsruhe, Germany, September, 2012.

12. A. Barreteau, S. Le Nours, O. Pasquier, *A simulation-based approach for performance evaluation of SDR baseband architectures*, proceedings of Wireless Innovation Forum European Conference on Communication Technologies and Software Defined Radio (WinnComm'12), Brussels, Belgium, June, 2012.

13. S. Le Nours, A. Barreteau, O. Pasquier, *A state-based modeling approach for fast performance evaluation of embedded system architectures*, proceedings of IEEE International Symposium on Rapid System Prototyping (RSP'11), Karlsruhe, Germany, October, 2011.

14. S. Le Nours, A. Barreteau, O. Pasquier, *A generic execution model for efficient performance evaluation of system architectures at transaction level*, in proceedings of Forum on Speficiation and Design Languages (FDL'11), Oldenburg, Germany, September, 2011.

15. T. Majdoub, S. Le Nours, O. Pasquier, F. Nouvel, *Transaction level modeling of a networked embedded system based on a power line communication protocol*, in proceedings of Euromicro Conference on Digital System Design (DSD'11), Oulu, Finland, September, 2011.

16. M. Pham, S. Pillement, O. Pasquier, S. Le Nours, *A framework for the design of reconfigurable fault tolerant architectures*, in proceedings of Conference on Design and Architectures for Signal and Image Processing (DASIP'11), Tampere, Finland, November, 2011.
17. S. Le Nours, A. Barreteau, O. Pasquier, *Modeling technique for simulation time speed-up of performance computation in transaction level models*, in proceedings of Forum on Specification and Design Languages (FDL'10), Southampton, UK, September, 2010.
18. S. Le Nours, O. Pasquier, D. Aoun, *Model-based approach for performance assessment of a video transmission application for automotive*, in proceddings of Workshop on Model-based engineering for embedded systems design (M-BED'10), Dresen, Germany, March, 2010.
19. A. Barreteau, S. Le Nours, O. Pasquier, *Executable models for performance assessments of adaptive mobile systems*, in proceedings of Wireless Innovation Forum Conference on Wireless Communications Technologies and Software Defined Radio, Washington, USA, 2009.
20. A. Barreteau, S. Le Nours, O. Pasquier, J.P. Calvez, *Transaction level modeling of an adaptive multi-standard and multi-application radio communication system*, in proceddings of Forum on Specification and Design Languages (FDL'09), Nice, France, September, 2009.
21. M. Cheik Wafa, S. Le Nours, O. Pasquier, J.P. Calvez, *Transaction level modeling of a flexray communication network*, in proceddings of Forum on Specification and Design Languages (FDL'09), Nice, France, September, 2009.
22. L. Dorie, O. Pasquier, S. Le Nours, J.F. Diouris, *A high level modeling approach for reconfigurable system architecting*, in proceedings of Workshop on Discrete-Event Systems Design (DESDES'09), Valencia, Spain, 2009.
23. S. Huet, O. Pasquier, S. Le Nours, *Granularity issues in transaction level modeling digital signal processing applications*, in proceedings of Forum on Specification and Design Languages (FDL'07), Barcelona, Spain, 2007.
24. L. Dorie, S. Le Nours, O. Pasquier, J.F. Diouris, *A system level model for software defined radio design*, in proceedings of IEEE Radio and Wireless Symposium (RWS'06), San Diego, USA, January, 2006.
25. L. Dorie, S. Le Nours, O. Pasquier, J.F. Diouris, *A high-level methodology for software defined radio hardware/software co-design*, in proceedings of Workshop on Software Radio, Karlsruhe, Germany, 2006.
26. F. Nouvel, S. Le Nours, I. Hermann, *AAA methodology and SynDEx tool capabilities for designing on heterogeneous architecture*, in proceedings of Conference on Design of Circuits and Integrated Systems (DCIS'03), Ciudad Real, Spain, 2003.
27. S. Le Nours, F. Nouvel, J.F. Hélard, *Efficient implementation of a MC-CDMA transmission system for the downlink*, in proceedings of IEEE Vehicular Technology Conference (VTC'01), Atlantic City, USA, 2001.
28. S. Le Nours, F. Nouvel, J.F. Hélard, *Rapid prototyping of MC-CDMA transmission technique on HW/SW architecture*, in proceedings of Conference on Design of Circuits and Integrated Systems (DCIS'01), Porto, Portugal, 2001.

National conferences with program committee

1. M. Pham, S. Pillement, O. Pasquier, S. Le Nours, *Modélisation et implémentation de calculateurs reconfigurables tolérants aux fautes et communications flexibles intra-véhicules*, in proceedings of Symposium en Architectures nouvelles de machines (Sympa'11), 2011.
2. A. Barreteau, S. Le Nours, O. Pasquier, *Démarche d'exploration d architectures pour le dimensionnement d un terminal mobile LTE*, in proceedings of Colloque GRETSI, Bordeaux, France, 2011.

3. L. Dorie, S. Le Nours, O. Pasquier, J.F. Diouris, *Une modélisation de niveau système pour la conception de SoC reconfigurables*, in proceedings of Congrès francophone de doctorants en STIC MajecStic'06), 2006.
4. A. Massiani, F. Nouvel, S. Le Nours, O. Pasquier, *Méthodologie de conception appliqué aux systèmes de radio logicielle*, in proceedings of Journées Françaises Adéquation Algorithme Architecture (JFAAA'05), 2005.
5. S. Le Nours, F. Nouvel, J.F. Hélard, *Example of Co-Design approach for a MC-CDMA transmission system*, in proceedings of Journées Françaises Adéquation Algorithme Architecture (JFAAA'02), 2002.

Research reports

1. R. Stemmer, H.D. Vu, M. Fakih, K. Grüttner, S. Le Nours, S. Pillement, *Feasibility Study of Probabilistic Timing Analysis Methods for SDF Applications on Multi-Core Processors* , IETR, OFFIS, Research report, 2019.
2. S. Le Nours, *Experimentation of a software code generator tool for Intel Galileo 2 prototyping platform*, Rapport de prestation, 2017.
3. S. Le Nours, *Development of a generation tool of embedded software codes for Intel CoFluent Studio framework*, Rapport de prestation, 2016.
4. S. Le Nours, Pasquier, O., T. Majdoub, *Modélisation de l'architecture reconfigurable et du réseau de communication*, IETR, Livrable du projet ANR CIFAER, 2012.
5. S. Le Nours, Pasquier, O., T. Majdoub, *Bilan sur les techniques mises au point pour la modélisation du système reconfigurable*, IETR, Livrable du projet ANR CIFAER, 2012.
6. S. Le Nours, *Utilisation de l'outil Matlab au sein de l'environnement CoFluent Studio*, IREENA, 2005.

Bibliography

[1] W. J.B., *Nano-informatique et intelligence ambiante*. Hermes Science, 2007.

[2] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, no. 8, pp. 114–117, Apr. 1965.

[3] Gajski and Kuhn, "Guest editors' introduction: New vlsi tools," *Computer*, vol. 16, no. 12, pp. 11–14, 1983. DOI: 10.1109/MC.1983.1654264.

[4] J. Staunstrup and W. Wolf, *Hardware/Software Co-Design, Principles and practice*. Springer US, 1997.

[5] L. Cai and D. Gajski, "Transaction level modeling: An overview," in *First IEEE/ACM/IFIP International Conference on Hardware/ Software Codesign and Systems Synthesis (IEEE Cat. No.03TH8721)*, 2003, pp. 19–24. DOI: 10.1109/CODESS.2003.1275250.

[6] A. Donlin, "Transaction level modeling: Flows and use models," in *International Conference on Hardware/-Software Codesign and System Synthesis, 2004. CODES + ISSS 2004.*, 2004, pp. 75–80. DOI: 10.1109/CODESS.2004.240821.

[7] J. Calvez, *Embedded real-time systems. A specification and design methodology*. John Wiley, 1993.

[8] D. D. Gajski, F. Vahid, S. Narayan, and J. Gong, *Specification and Design of Embedded Systems*. USA: Prentice-Hall, Inc., 1994, ISBN: 0131507311.

[9] K. Keutzer, A. R. Newton, J. M. Rabaey, and A. Sangiovanni-Vincentelli, "System-level design: Orthogonalization of concerns and platform-based design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 12, pp. 1523–1543, 2000. DOI: 10.1109/43.898830.

[10] I. S. Association *et al.*, "Ieee standard for standard systemc language reference manual," *IEEE Computer Society*, 2012.

[11] ——, "Ieee standard for system verilog - unified hardware design, specification and verification language," *IEEE Computer Society*, 2017.

[12] D. D. Gajski, J. Zhu, R. Dömer, A. Gerstlauer, and S. Zhao, *SpecC: specification language and design methodology*. Norwell, MA: Kluwer, 2000.

[13] F. Ghenassia, *Transaction level modeling with SystemC: TLM concepts and applications for embedded systems*. Springer, 2005.

[14] A. Sangiovanni-Vincentelli, "Quo vadis, sld? reasoning about the trends and challenges of system level design," *Proceedings of the IEEE*, vol. 95, no. 3, pp. 467–506, 2007. DOI: 10.1109/JPROC.2006.890107.

[15] J. Sifakis, "System design automation: Challenges and limitations," *Proceedings of the IEEE*, vol. 103, no. 11, pp. 2093–2103, 2015. DOI: 10.1109/JPROC.2015.2484060.

[16] D. D. Gajski, S. Abid, A. Gerstlaeur, and G. Schirner, *Embedded system design: modeling, synthesis and verification*, Springer, Ed. 2009.

[17] AFIS, *Découvrir et comprendre l'Ingénierie Système*, Cépaduès-Editions, Ed. 2012.

[18] L. Lavagno, A. Sangiovanni-Vincentelli, and E. Sentovich, "Models of computation for embedded system design," in *System-Level Synthesis*, A. A. Jerraya and J. Mermet, Eds. Dordrecht: Springer Netherlands, 1999, pp. 45–102, ISBN: 978-94-011-4698-2. DOI: 10.1007/978-94-011-4698-2_2. [Online]. Available: https://doi.org/10.1007/978-94-011-4698-2_2.

[19] M. Gries, "Methods for evaluating and covering the design space during early design development," *Integr. VLSI J.*, vol. 38, no. 2, pp. 131–183, Dec. 2004, ISSN: 0167-9260. DOI: 10.1016/j.vlsi.2004.06.001. [Online]. Available: https://doi.org/10.1016/j.vlsi.2004.06.001.

[20] A. Gerstlauer, C. Haubelt, A. D. Pimentel, T. P. Stefanov, D. D. Gajski, and J. Teich, "Electronic system-level synthesis methodologies," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 10, pp. 1517–1530, Oct. 2009, ISSN: 0278-0070. DOI: 10.1109/TCAD.2009.2026356.

[21] T. Mitra, J. Teich, and L. Thiele, "Time-critical systems design: A survey," *IEEE Design Test*, vol. 35, no. 2, pp. 8–26, 2018. DOI: 10.1109/MDAT.2018.2794204.

[22] C. Maiza, H. Rihani, J. M. Rivas, J. Goossens, S. Altmeyer, and R. I. Davis, "A survey of timing verification techniques for multi-core real-time systems," *ACM Comput. Surv.*, vol. 52, no. 3, Jun. 2019, ISSN: 0360-0300. DOI: 10.1145/3323212. [Online]. Available: https://doi.org/10.1145/3323212.

[23] R. Wilhelm, D. Grund, J. Reineke, M. Schlickling, M. Pister, and C. Ferdinand, "Memory hierarchies, pipelines, and buses for future architectures in time-critical embedded systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 7, pp. 966–978, 2009. DOI: 10.1109/TCAD.2009.2013287.

[24] G. Fernandez, J. Abella, E. Quiñones, C. Rochange, T. Vardanega, and F. J. Cazorla, "Contention in Multicore Hardware Shared Resources: Understanding of the State of the Art," in *14th International Workshop on Worst-Case Execution Time Analysis*, H. Falk, Ed., ser. OpenAccess Series in Informatics (OASIcs), vol. 39, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2014, pp. 31–42, ISBN: 978-3-939897-69-9. DOI: 10.4230/OASIcs.WCET.2014.31. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2014/4602.

[25] S. Hahn, J. Reineke, and R. Wilhelm, "Towards compositionality in execution time analysis: Definition and challenges," *SIGBED Rev.*, vol. 12, no. 1, pp. 28–36, Mar. 2015. DOI: 10.1145/2752801.2752805. [Online]. Available: https://doi.org/10.1145/2752801.2752805.

[26] C. Cullmann, C. Ferdinand, G. Gebhard, D. Grund, C. Maiza, J. Reineke, B. Triquet, and R. Wilhelm, "Predictability considerations in the design of multi-core embedded systems," May 2010, pp. 36–42.

[27] Intel, *Intel cofluent studio*, http://www.intel.com/content/www/us/en/cofluent/intel-cofluent-studio.html.

[28] C. Cassandras and S. Lafortune, *Introduction to discrete event systems*, Second edition. Springer, 2008.

[29] R. Dömer, A. Gerstlauer, J. Peng, D. Shin, L. Cai, H. Yu, S. Abdi, and D. Gajski, "System-on-chip environment: A specc-based framework for heterogeneous mpsoc design," *EURASIP Journal on Embedded Systems*, vol. 2008, 2008.

[30] C. Erbas, A. D. Pimentel, M. Thompson, and S. Polstra, "A framework for system-level modeling and simulation of embedded systems architectures," *EURASIP Journal on Embedded Systems*, 2007.

[31] J. Keinert, M. Streubühr, T. Schlichter, J. Falk, J. Gladigau, C. Haubelt, J. Teich, and M. Meredith, "Systemcodesigner-an automatic esl synthesis approach by design space exploration and behavior synthesis for streaming applications," *ACM Trans. Des. Autom. Electro. Syst.*, vol. 14, no. 1, pp.1–23, Jan. 2009.

[32] SpaceCoDesign, www.spacecodesign.com.

[33] MirabilisDesign, www.mirabilisdesign.com.

[34] J. Kreku, M. Hoppari, T. Kestilä, Y. Qu, J. Soininen, P. Andersson, and K. Tiensyrjä, "Combining uml2 application and systemc platform modelling for performance evaluation of real-time embedded systems," *EURASIP Journal on Embedded Systems*, vol. 2008, 6:1–6:18, Jan. 2008.

[35] S. Stattelmann, O. Bringmann, and W. Rosenstiel, "Fast and accurate resource conflict simulation for performance analysis of multi-core systems," in *Proc. Design, Automation and Test in Europe (DATE'11)*, Grenoble France, Mar. 2011.

[36] K. Lu, D. Muller-Gritschneder, and U. Schlichtmann, "Analytical timing estimation for temporally decoupled TLMs considering resource conflicts," in *Proc. Design, Automation and Test in Europe (DATE'13)*, Grenoble, France, Mar. 2013, pp. 1161–1166.

[37] G. Schirner and R. Dömer, "Result-oriented modeling - a novel technique for fast and accurate TLM," *IEEE Transactions on computer-aided design of integrated circuits and systems*, vol. 26, no. 9, pp. 1688–1699, Sep. 2007.

[38] ——, "Introducing preemptive scheduling in abstract RTOS models using result oriented modeling," in *Proc. Design, Automation and Test in Europe (DATE'08)*, Munich, Germany, Mar. 2008, pp. 122–127.

[39] A. Bobrek, J. M. Paul, and D. E. Thomas, "Stochastic contention level simulation for single-chip heterogeneous multiprocessors," *IEEE Transactions on Computers*, vol. 59, no. 10, pp. 1402–1418, 2010.

[40] S. Künzli, F. Poletti, L. Benini, and L. Thiele, "Combining simulation and formal methods for system-level performance analysis," in *Design, Automation and Test in Europe (DATE)*, Munich, Germany, 2006, pp. 236–241.

[41] S.-Y. Chen, C.-H. Chen, and R.-S. Tsay, "An activity-sensitive contention delay model for highly efficient deterministic full-system simulations," in *In Proc. of Design, Automation and Test in Europe (DATE'14)*, 2014.

[42] F. Baccelli, G. Cohen, G. Olsder, and J. Quadrat, *Synchronization and linearity, an algebra for discrete event systems*. New York: Wiley & Sons Ltd, 1992.

[43] C. Seidner and O. H. Roux, "Formal methods for systems engineering behavior models," *IEEE Transactions on industrial informatics*, vol. 4, pp. 280–291, 2008.

[44] *Acceleo*, https://www.eclipse.org/acceleo/.

[45] G. Schirner and R. Dömer, "Quantitative analysis of the speed/accuracy trade-off in transaction level modeling," *ACM Trans. Embed. Comput. Syst.*, vol. 8, no. 1, Jan. 2009, ISSN: 1539-9087. DOI: 10.1145/1457246.1457250. [Online]. Available: https://doi.org/10.1145/1457246.1457250.

[46] T. Majdoub, "Technique de modélisation transactionnelle en vue de l'amélioration de la simulation des modèles de performances des architectures électroniques dans le domaine automobile," Ph.D. dissertation, University of Nantes, Oct. 2012.

[47] H.-D. Vu, "Fast and accurate performance models for probabilistic timing analysis of sdfgs on mpsocs," Ph.D. dissertation, University of Nantes, Mar. 2021.
[48] E. A. Lee and D. G. Messerschmitt, "Synchronous data flow," *Proceedings of the IEEE*, vol. 75, no. 9, pp. 1235–1245, 1987.
[49] A. Limited, "Amba® axi™ and ace™ protocol specification," AXI3, AXI4, and AXI4-Lite ACE and ACE-Lite.
[50] A. Barreteau, "Techniques de modélisation transactionnelle pour le dimensionnement des futurs systèmes de radiocommunications mobiles," Ph.D. dissertation, University of Nantes, Dec. 2010.
[51] 3. LTE, http://www.3gpp.org/LTE.
[52] A. K. Singh, M. Shafique, A. Kumar, and J. Henkel, "Mapping on multi/many-core systems: Survey of current and emerging trends," in *2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2013, pp. 1–10. DOI: 10.1145/2463209.2488734.
[53] A. D. Pimentel, "Exploring exploration: A tutorial introduction to embedded systems design space exploration," *IEEE Design Test*, vol. 34, no. 1, pp. 77–90, 2017. DOI: 10.1109/MDAT.2016.2626445.
[54] A. K. Singh, P. Dziurzanski, H. R. Mendis, and L. S. Indrusiak, "A survey and comparative study of hard and soft real-time dynamic resource allocation strategies for multi-/many-core systems," *ACM Comput. Surv.*, vol. 50, no. 2, Apr. 2017, ISSN: 0360-0300. DOI: 10.1145/3057267. [Online]. Available: https://doi.org/10.1145/3057267.
[55] S. Pagani, A. Pathania, M. Shafique, J.-J. Chen, and J. Henkel, "Energy efficiency for clustered heterogeneous multicores," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 5, pp. 1315–1330, 2017. DOI: 10.1109/TPDS.2016.2623616.
[56] H. Khdr, S. Pagani, É. Sousa, V. Lari, A. Pathania, F. Hannig, M. Shafique, J. Teich, and J. Henkel, "Power density-aware resource management for heterogeneous tiled multicores," *IEEE Transactions on Computers*, vol. 66, no. 3, pp. 488–501, 2017. DOI: 10.1109/TC.2016.2595560.
[57] A. K. Singh, M. Shafique, A. Kumar, and J. Henkel, "Resource and throughput aware execution trace analysis for efficient run-time mapping on mpsocs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 1, pp. 72–85, 2016. DOI: 10.1109/TCAD.2015.2446938.
[58] S. Yang, "Evaluation and design of a run-time manager for ultra-low power multiprocessor systems on chip," Ph.D. dissertation, University of Nantes, Jun. 2020.
[59] T. S. Muthukaruppan, M. Pricopi, V. Venkataramani, T. Mitra, and S. Vishin, "Hierarchical power management for asymmetric multi-core in dark silicon era," in *2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2013, pp. 1–9. DOI: 10.1145/2463209.2488949.
[60] W. Quan and A. D. Pimentel, "A hybrid task mapping algorithm for heterogeneous mpsocs," *ACM Trans. Embed. Comput. Syst.*, vol. 14, no. 1, Jan. 2015, ISSN: 1539-9087. DOI: 10.1145/2680542. [Online]. Available: https://doi.org/10.1145/2680542.
[61] J. Lemaitre and R. Le Moigne, "Dynamic migration and performance optimization of deterministic applications across platform components using intel cofluent studio," in *In Proc. DAC Workshop on System-to-Silicon Performance Modeling and Analysis*, 2015.
[62] R. Davis and L. Cucu-Grosjean, "A Survey of Probabilistic Timing Analysis Techniques for Real-Time Systems," *Leibniz Transactions on Embedded Systems*, vol. 6, no. 1, p. 60, 2019. DOI: 10.4230/LITES-v006-i001-a003. [Online]. Available: https://hal.inria.fr/hal-02158973.
[63] F. J. Cazorla, J. Abella, J. Andersson, T. Vardanega, F. Vatrinet, I. Bate, I. Broster, M. Azkarate-Askasua, F. Wartel, L. Cucu, F. Cros, G. Farrall, A. Gogonel, A. Gianarro, B. Triquet, C. Hernandez, C. Lo, C. Maxim, D. Morales, E. Quinones, E. Mezzetti, L. Kosmidis, I. Aguirre, M. Fernandez, M. Slijepcevic, P. Conmy, and W. Talaboulma, "Proxima: Improving measurement-based timing analysis through randomisation and probabilistic analysis," in *2016 Euromicro Conference on Digital System Design (DSD)*, 2016, pp. 276–285. DOI: 10.1109/DSD.2016.22.
[64] J.-P. Katoen and H. Wu, "Probabilistic model checking for uncertain scenario-aware data flow," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 22, no. 1, 15:1–15:27, Sep. 2016, ISSN: 1084-4309. DOI: 10.1145/2914788. [Online]. Available: http://doi.acm.org/10.1145/2914788.
[65] L. Santinelli and L. Cucu-Grosjean, "A Probabilistic Calculus for Probabilistic Real-Time Systems," *ACM Transactions on Embedded Computing Systems (TECS)*, RTAS2012 special issue, vol. 14, no. 3, 2015. DOI: 10.1145/2717113. [Online]. Available: https://hal.inria.fr/hal-01244333.
[66] K. Huang, W. Haid, I. Bacivarov, M. Keller, and L. Thiele, "Embedding formal performance analysis into the design cycle of mpsocs for real-time streaming applications," *ACM Trans. Embed. Comput. Syst.*, vol. 11, no. 1, 8:1–8:23, Apr. 2012, ISSN: 1539-9087. DOI: 10.1145/2146417.2146425. [Online]. Available: http://doi.acm.org/10.1145/2146417.2146425.
[67] C. Schlaak, M. Fakih, and R. Stemmer, "Power and execution time measurement methodology for sdf applications on fpga-based mpsocs," *arXiv preprint arXiv:1701.03709*, 2017.

[68] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *Ann. Math. Statist.*, vol. 27, no. 3, pp. 832–837, Sep. 1956. DOI: 10.1214/aoms/1177728190. [Online]. Available: https://doi.org/10.1214/aoms/1177728190.

[69] A. D. Pimentel, M. Thompson, S. Polstra, and C. Erbas, "Calibration of abstract performance models for system-level design space exploration," *J. Signal Process. Syst.*, vol. 50, no. 2, pp. 99–114, Feb. 2008, ISSN: 1939-8018. DOI: 10.1007/s11265-007-0085-2. [Online]. Available: https://doi.org/10.1007/s11265-007-0085-2.

[70] A. Nouri, M. Bozga, A. Molnos, A. Legay, and S. Bensalem, "Astrolabe: A rigorous approach for system-level performance modeling and analysis," *ACM Trans. Embed. Comput. Syst.*, vol. 15, no. 2, Mar. 2016, ISSN: 1539-9087. DOI: 10.1145/2885498. [Online]. Available: https://doi.org/10.1145/2885498.

[71] A. Nouri, S. Bensalem, M. Bozga, B. Delahaye, C. Jegourel, and A. Legay, "Statistical model checking qos properties of systems with sbip," *International Journal on Software Tools for Technology Transfer*, vol. 17, no. 2, pp. 171–185, 2014.

[72] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distributions," *Bull. Calcutta Math. Soc.*, vol. 35, pp. 99–109, 1943.

[73] *Centre de calcul intensif des pays de la loire*, https://ccipl.univ-nantes.fr/.

[74] A. Legay, B. Delahaye, and S. Bensalem, "Statistical model checking: An overview," in *Runtime Verification*, H. Barringer, Y. Falcone, B. Finkbeiner, K. Havelund, I. Lee, G. Pace, G. Roşu, O. Sokolsky, and N. Tillmann, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 122–135, ISBN: 978-3-642-16612-9.

[75] K. G. Larsen and A. Legay, "Statistical Model Checking: Past, Present, and Future," in *6th International Symposium, ISoLA 2014*, Corfu, Greece, Oct. 2014. [Online]. Available: https://hal.inria.fr/hal-01406518.

[76] D. Benoit, "Vérification Classique et Statistique pour les Systèmes Probabilistes," in *Ecole d'été Temps Réel (ETR)*, Paris, France, 2017.

[77] C. P. Robert, *Monte Carlo Methods*. Wiley Online Library, 2004.

[78] T. Hérault, R. Lassaigne, F. Magniette, and S. Peyronnet, "Approximate probabilistic model checking," in *Verification, Model Checking, and Abstract Interpretation*, B. Steffen and G. Levi, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 73–84, ISBN: 978-3-540-24622-0.

[79] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–30, 1963. DOI: 10.1080/01621459.1963.10500830.

[80] S. Laplante, R. Lassaigne, F. Magniez, S. Peyronnet, and M. de Rougemont, "Probabilistic abstraction for model checking: An approach based on property testing," *ACM Trans. Comput. Logic*, vol. 8, no. 4, 20–es, Aug. 2007, ISSN: 1529-3785. DOI: 10.1145/1276920.1276922. [Online]. Available: https://doi.org/10.1145/1276920.1276922.

[81] H. L. Younes, "Verification and planning for stochastic processes with asynchronous events," Ph.D. dissertation, Carnegie Mellon University, 2005.

[82] A. David, K. G. Larsen, A. Legay, M. Mikučionis, and D. B. Poulsen, "UPPAAL SMC tutorial," *International Journal on Software Tools for Technology Transfer*, vol. 17, no. 4, pp. 397–415, 2015.

[83] M. Kwiatkowska, G. Norman, and D. Parker, "Prism 4.0: Verification of probabilistic real-time systems," in *Computer Aided Verification*, G. Gopalakrishnan and S. Qadeer, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 585–591, ISBN: 978-3-642-22110-1.

[84] P. Bulychev, A. David, K. G. Larsen, M. Mikučionis, D. Bøgsted Poulsen, A. Legay, and Z. Wang, "Uppaal-smc: Statistical model checking for priced timed automata," *Electronic Proceedings in Theoretical Computer Science*, vol. 85, pp. 1–16, Jul. 2012, ISSN: 2075-2180. DOI: 10.4204/eptcs.85.1. [Online]. Available: http://dx.doi.org/10.4204/EPTCS.85.1.

[85] A. Nouri, "Rigorous system-level modeling and performance evaluation for embedded systems design," Ph.D. dissertation, Université Grenoble Alpes, 2015.

[86] A. Nouri, M. Bozga, A. Moinos, A. Legay, and S. Bensalem, "Building faithful high-level models and performance evaluation of manycore embedded systems," in *ACM/IEEE International conference on Formal methods and models for codesign*, 2014.

[87] A. Nouri, S. Bensalem, M. Bozga, B. Delahaye, C. Jegourel, and A. Legay, "Statistical model checking qos properties of systems with sbip," *International Journal on Software Tools for Technology Transfer*, vol. 17, no. 2, pp. 171–185, 2014.

[88] M. Chen, D. Yue, X. Qin, X. Fu, and P. Mishra, "Variation-aware evaluation of mpsoc task allocation and scheduling strategies using statistical model checking," in *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*, Mar. 2015, pp. 199–204.

[89] R. Stemmer, H. Schlender, M. Fakih, K. Grüttner, and W. Nebel, "Probabilistic state-based rt-analysis of sdfgs on mpsocs with shared memory communication," in *2019 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2019, pp. 1715–1720. DOI: 10.23919/DATE.2019.8715052.

[90] V. C. Ngo, A. Legay, and J. Quilbeuf, "Statistical model checking for systemc models," *2016 IEEE 17th International Symposium on High Assurance Systems Engineering*, pp. 197–204, 2016.

[91] B. Boyer, K. Corre, A. Legay, and S. Sedwards, "PLASMA-lab: A Flexible, Distributable Statistical Model Checking Library," in *Quantitative Evaluation of Systems*, Joshi, Kaustubh, Siegle, Markus, Stoelinga, Mariëlle, D'Argenio, and P. R., Eds., ser. Lecture Notes in Computer Science, vol. 8054, Buenos Aires, Argentina, Aug. 2013, pp. 160–164. DOI: 10.1007/978-3-642-40196-1_12. [Online]. Available: https://hal.inria.fr/hal-01088411.

[92] D. Tabakov and M. Y. Vardi, "Monitoring temporal systemc properties," in *Eighth ACM/IEEE International Conference on Formal Methods and Models for Codesign (MEMOCODE 2010)*, 2010, pp. 123–132. DOI: 10.1109/MEMCOD.2010.5558640.

[93] Y. Han, G. Huang, S. Song, L. Yang, H. Wang, and Y. Wang, *Dynamic neural networks: A survey*, 2021. arXiv: 2102.04906 [cs.CV].

[94] R. Davis and L. Cucu-Grosjean, "A Survey of Probabilistic Timing Analysis Techniques for Real-Time Systems," *Leibniz Transactions on Embedded Systems*, vol. 6, no. 1, p. 60, 2019. DOI: 10.4230/LITES-v006-i001-a003. [Online]. Available: https://hal.inria.fr/hal-02158973.

[95] F. Staudigl, F. Merchant, and R. Leupers, "A survey of neuromorphic computing-in-memory: Architectures, simulators and security," *IEEE Design & Test*, pp. 1–1, 2021. DOI: 10.1109/MDAT.2021.3102013.

[96] X. Zheng, L. K. John, and A. Gerstlauer, "Lacross: Learning-based analytical cross-platform performance and power prediction," *International Journal of Parallel Programming*, vol. 45, pp. 1488–1514, 2017.

[97] A. Gamatié, X. An, Y. Zhang, A. Kang, and G. Sassatelli, "Empirical Model-Based Performance Prediction for Application Mapping on Multicore Architectures," *Journal of Systems Architecture*, vol. 98, pp. 1–16, Sep. 2019. DOI: 10.1016/j.sysarc.2019.06.001. [Online]. Available: https://hal-lirmm.ccsd.cnrs.fr/lirmm-02151502.

[98] A. Saeed, D. Mueller-Gritschneder, F. Rehm, A. Hamann, D. Ziegenbein, U. Schlichtmann, and A. Gerstlauer, "Learning based memory interference prediction for co-running applications on multi-cores," in *2021 ACM/IEEE 3rd Workshop on Machine Learning for CAD (MLCAD)*, 2021, pp. 1–6. DOI: 10.1109/MLCAD52597.2021.9531245.

[99] G. Huang, J. Hu, Y. He, J. Liu, M. Ma, Z. Shen, J. Wu, Y. Xu, H. Zhang, K. Zhong, X. Ning, Y. Ma, H. Yang, B. Yu, H. Yang, and Y. Wang, "Machine learning for electronic design automation: A survey," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 26, no. 5, Jun. 2021, ISSN: 1084-4309. DOI: 10.1145/3451179. [Online]. Available: https://doi.org/10.1145/3451179.