

## Tutorial on Formal Methods for Distributed and Cooperative Systems

Christine Choppy<sup>1</sup>, Serge Haddad<sup>2</sup>, Hanna Klaudel<sup>3</sup>, Fabrice Kordon<sup>4</sup>, Laure Petrucci<sup>1</sup>, and Yann Thierry-Mieg<sup>4</sup>

<sup>1</sup> Université Paris 13, LIPN, CNRS UMR 7030

99 avenue Jean-Baptiste Clément, F-93430 Villetaneuse, France

{Christine.Choppy,Laure.Petrucci}@lipn.univ-paris13.fr

<sup>2</sup> Université Paris-Dauphine, LAMSADE, CNRS UMR 7024

place du Maréchal de Lattre de Tassigny, F-75775 Paris Cedex 16

haddad@lamsade.dauphine.fr

<sup>3</sup> Université d'Evry-Val-d'Essone, IBISC, CNRS FRE 2873

523, place des Terrasses de l'Agora, 91000 Evry - France

Hanna.Klaudel@ibisc.fr

<sup>4</sup> Université P. & M. Curie, LIP6/MoVe, CNRS UMR 7606

4, place Jussieu, F-75252 Paris Cedex 05, France

Fabrice.Kordon@lip6.fr, Yann.Thierry-Mieg@lip6.fr

### 1 Introduction

This tutorial is proposed by representatives of the MeFoSyLoMa<sup>5</sup> group. MeFoSyLoMa is an informal group gathering several teams from various universities in the Paris area:

- Université Paris-Dauphine (LAMSADE laboratory),
- Université P. & M. Curie (LIP6 laboratory),
- Université Paris 13 (LIPN laboratory),
- ENST (LTCI laboratory),
- Conservatoire National des Arts et Métiers (CEDRIC laboratory).

These teams have extensive knowledge and experience in the design, analysis and implementation of distributed systems. The cooperation within the group aims at joining forces, sharing experiences and building joint projects to solve issues in the design of reliable distributed systems.

One of the major actions of this community is a collective book due to appear in the fall 2006, and entitled “*Formal Methods for Distributed Cooperative Systems*” (*Méthodes formelles pour les systèmes répartis et coopératifs* in french, published by Hermès). The purpose of this book is to gather a state of the art of the most advanced techniques for modelling and formal analysis of distributed systems.

The book is divided into three parts, which constitute the basis for the tutorial we propose at ICTAC 2006. Following the design process, the first part

<sup>5</sup> MeFoSyLoMa stands for *Formal Methods for Software and Hardware Systems* (*Méthodes Formelles pour les Systèmes Logiciels et Matériels* in french).

deals with specification of distributed systems, the second one with analysis techniques and the third one presents actual experimentations of such modelling and verification techniques in real systems (i.e. industrial size case studies).

Each part of the book will correspond to a two hours tutorial presented by two of the authors of the corresponding part.

## 2 Part I: dedicated specification languages and models

This part is devoted to giving guidelines for designing a consistent system model. First, we present criteria to consider so as to build a specification satisfying the demands. Then, methodologies to write specifications are introduced.

*Presenters:* This part of the tutorial will be presented by:

- Laure PETRUCCI, Professor at University Paris 13, member of the CNRS laboratory LIPN,
- Christine CHOPPY, Professor at University Paris 13, member of the CNRS laboratory LIPN.

*Outline:* Many sorts of models can be used to specify a complex system. Hence, it might prove difficult, for a non-experienced designer, to choose among this large collection. Therefore, the first part of the presentation is dedicated to criteria that should be taken into account before choosing a modelling formalism: relevant concepts, abstraction level, specification goals, structuring, expected properties.

The *relevant concepts* are the different data types important in the system, timing issues, the structure of the system, i.e. its decomposition into subsystems, sequential or parallel execution mode, synchronous/asynchronous communication. When considering the *abstraction level*, one should keep in mind that the specification process may include incremental development and refinement. The *relevance* of the concepts and the abstraction level must be considered w.r.t. the goals for designing the specification, e.g. time is relevant when checking scheduling issues but may not be when verifying the correctness of a communication protocol. *Structuring the specification* into subsystems allows for both a better view of the different components and reusability. Finally, the *expected properties* must be written in a language consistent with the modelling technique.

Considering all these criteria should help the designer in choosing a specification paradigm, the appropriate level of abstraction and the relevant properties to be checked using the model.

The second part of the presentation is concerned with *guidelines* to start writing the detailed specification, considering data types structures, simple dynamic systems, and dynamic systems structured using subsystems. This approach can be combined with other approaches that guide the overall structuring of the specification using structuring concepts provided by *problem frames*, or *architectural styles*, and with a *component approach* to combine the specifications developed.

### 3 Part II: dedicated verification techniques

This part is dedicated to efficient verification methods for distributed applications and systems.

*Presenters:* This part of the tutorial will be presented by:

- Serge HADDAD, Professor at University Paris-Dauphine, member of the CNRS laboratory Lamsade,
- Yann THIERRY-MIEG, Associate Professor at University P. & M. Curie, member of the CNRS laboratory LIP6.

*Outline:* The presentation comprises two parts.

The diversity of verification methods may puzzle the engineer facing the choice of the appropriate technique for analysing her/his system. So the first part of the presentation aims at clarifying the bases of such a choice by discussing three critical questions associated with the verification process:

- How to choose the formalism for verification?
- How to express the expected properties of the system modelled?
- Which verification methods apply on the model?

The second part of the presentation details one of the most successful verification methods in order to tackle the increasing complexity of the systems: the decision diagram based methods. It starts with a general introduction on data representation and manipulation using such structures. Then it shows how the reachability problem (the main verification problem) can efficiently be handled and how to generalise its application to the model-checking of temporal logic formulae. It concludes by detailing experiments in order to understand when and why the method is successful.

The whole presentation illustrates the different concepts with the help of (extended) Petri nets.

### 4 Part III: application to distributed systems

This part is dedicated to the effective use of the techniques presented in the design and implementation of real systems.

*Presenters:* This part of the tutorial will be presented by:

- Fabrice KORDON, Professor at University P. & M. Curie and head of the MoVe (Modelling and Verification) team in the CNRS laboratory LIP6,
- Hanna KLAUDEL, Professor at University Evry-Val-d'Essonne and head of the LIS (Languages, Interaction, Simulation) team in the CNRS laboratory IBISC.

*Outline:* The presentation is divided into two parts. The first one is devoted to the PolyORB experience. PolyORB is a middleware dedicated to distributed real-time systems. It thus requires high reliability that is achieved by means of an original architecture on which formal verification of qualitative properties (such as absence of deadlock or livelock) is enforced thanks to Symmetric Petri Nets<sup>6</sup>. The presentation will explain how a strong interaction between the design of the software architecture, combined with new model-checking techniques, allows for coping with a high complexity of the formal specification.

The second part deals with the design of adaptive reactive systems, i.e. systems that dynamically adapt their architecture depending on the context of the execution. We use the formalism of timed automata for the design of the modules behaviour. Hence, it is possible to evaluate beforehand the properties of the system (regarding logical correctness and timelines), thanks to model-checking and simulation techniques. The approach is illustrated by a case study for which we show how to produce very quickly a running prototype satisfying the properties of the model, and how to evaluate *a priori* the pertinence of adaptive strategies.

## 5 Conclusion

This tutorial is thus intended for young researchers or engineers to have an overview of the specification process.

The first part is concerned with specification issues. Indeed, writing a specification from scratch is a difficult task for a non-experienced person. The criteria pointed out and the specification design methodology should help in choosing the appropriate formalism and starting the design of a system.

The second part is devoted to analysis issues. A complex model is intrinsically difficult to analyse. It is thus important to choose the appropriate technique to prove the expected properties of the system. Some advanced techniques are also shortly presented, which give a feeling on how to handle large systems.

Finally, the third part shows how these techniques have been successfully applied to real systems.

---

<sup>6</sup> formerly called in the literature Well-formed Petri Nets.