Project no.:	**ICT-FP7-STREP-214755**
Project full title:	**Quantitative System Properties in Model-Driven Design**
Project Acronym:	**QUASIMODO**

Deliverable no.: D5.12a

Title of Deliverable: Industrial Handbook version 1

Contractual Date of Delivery to the CEC:	**M24**
Actual Date of Delivery to the CEC:	**M24**
Organisation name of lead contractor for this deliverable:	**ESI**
Author(s):	**Jan Tretmans**
Participants(s):	**All**
Work package contributing to the deliverable:	**WP5**
Nature:	**R**
Version:	**1.0**
Total number of pages:	**10**
Start date of project:	1 Jan. 2008 **Duration:** 36 month

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)

Dissemination Level

PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Abstract:

This deliverable contains the detailed contents for the ”Handbook on Quantitative Model-Driven Development for Embedded Systems” to be published at the end of the Quasimodo project.

Keyword list: Quasimodo, Handbook.

Contents

Abbreviations

AAU: Aalborg University, DK

CFV: Centre Fèdèrè en Vèrification, B

CNRS: National Center for Scientific Research, FR

ESI: Embedded Systems Institute, NL

ESI/RU: Radboud University Nijmegen, NL

RWTH: RWTH Aachen University, D

SU: Saarland University, D

1 Introduction

The Quasimodo project develops methods, techniques, and tools for handling quantitative properties in model-driven development of real-time embedded systems. To promote the exploitation and use of the project results Quasimodo also invests in activities for communication, dissemination, and use of project results. Different activities are organized in this respect, see Deliverable D5.6 ”Dissemination and Exploitation”. One of these activities is writing and publishing a *Handbook on Quantitative Model-Driven Development for Embedded Systems* to be published at the end of the project. This deliverable starts the work on this handbook by defining the scope, focus, intended audience, and detailed contents of this joint effort.

2 Scope of the Handbook

The goal of the handbook is to provide an overview of the Quasimodo quantitative analyisis methods, techniques, and tools for model-based development of real-time embedded systems.

The selection of methods, techniques, and tools is guided by what has been developed or used in the Quasimodo project, that is, the handbook will not give a comprehensive and complete overview of all quantitative analysis methods. The presented methods, techniques, and tools are those that are mature enough so that they can be applied in an embedded systems development project, and evidence of this maturity will be provided in the handbook, peferably by an elaborated Quasimodo case study. In this way the handbook will be an easily accessible entry point to successful and applicable knowledge from the Quasimodo project.

The handbook is targeted towards the professional embedded systems engineer, who would like to know what is available in the area of quantitative, model-based analysis of embedded systems, what is mature enough to be applied, how it can be applied, and what benefits can be expected from using such a method, technique, or tool. Such an engineer typically has an MSc. in engineering, has some experience in making or reading some kind of models, but does not know about timed automata or model checking. The handbook is not primarily intended for scientists, nor for managers (except perhaps the introduction). This means that scientific originality is not a criterion for the selection of topics, but demonstrated applicability is. It also implies that some of the presented methods, techniques, and tools may have been available already before the start of Quasimodo, but are not yet applied in the embedded systems engineering practice.

at end of each chapter

3 Global Contents of the Handbook

Taking into account the scope of the handbook described above, and the research and case studies done in Quasimodo, we come to the following topic areas:

1. modelling
2. model checking

3. tools
4. scheduling
5. controller synthesis
6. probilistic analysis
7. model-based testing

In addition, there will be chapters on Introduction and Perspectives, The next section will describe these topic areas in more detail, and will give a division over chapters.

4 Detailed Contents of the Handbook

We describe the contents of the handbook in terms chapters with titles, topics, estimated number of pages, and authors. There are 15 chapters with an estimated total number of pages of 225.

All topics mentioned in the previous chapter are covered with both an explanatory, introductory chapter, and a case study chapter. The topic of tools is dispersed over the various chapters.

A couple of chapters are identified as risky, because the work on which they will be based is not finished yet. Where appropriate alternatives are proposed.

The preliminary title of the book:

Quasimodo: Quantitative, Model-Based Analysis of Real-Time Embedded Systems

1. **Introduction**
   - Goal: (managerial) introduction to topic and book
   - Length: 10 pages.
   - Authors: *Jan Tretmans*, Kim Larsen, Brian Nielsen
   - Sources: Quasimodo leaflet, introductory texts project description
   - Topics:
     - goal of book
     - motivation book and topic: embedded systems, problems, needs, quantitative analysis, timing, probabilities, energy, ...
     - topics in design of ES, topics covered in this book, topics not covered
     - approach: model-based/driven
     - overview of chapters, reading guide, role of case studies
2. **Modeling Real-Time Systems with UPPAAL**
   - Goal: modeling with UPPAAL for beginners

- Length: 60 pages
- Authors: *Frits Vaandrager*
- Topics:
  - tutorial like chapter, focusing not so much on the specific syntax and semantics of UPPAAL, but rather on the modeling of real-systems.
  - introductory section on "What is a good model?", abstraction levels, choices in modeling
  - followed by a series of up to 10 case studies of increasing complexity
  - untimed and real-time modeling
  - editing, simulation, verification with UPPAAL
  - basics of property checking with UPPAAL

3. **Timed Automata and UPPAAL**

- Goal: more thorough and formal description of timed automata, model checking, and UPPAAL & sons, as far as required for subsequent chapters.
- Length: 20 pages
- Authors: *Kim Larsen,* AAU
- Sources: existing tutorial material.
- Topics:
  - timed automata, syntax, a bit of semantics
  - specification language
  - applied model checking
  - overview UPPAAL tool suite: UPPAAL, CORA, TIGA, TRON, PRO
- Note: difference between Chapters 2 and 3 is the starting point of view. Starting point in Chapter 2 is the user with his problems and modeling wishes. Chapter 3 starts from a technical/formal point of view.

4. **Model-Checking for Wireless Sensor Networks: The gMAC Synschronization Protocol Case**

- Goal: present a realistic case study with UPPAAL modeling and model checking of the CHESS WSN synchronization protocol, illustrating chapters 2 and 3.
- Length: 10 pages
- Authors: *Frits Vaandrager*, Julien Schmaltz, Faranak Heidarian
- Sources: existing papers
- Topics:

  - modelling and model checking of the gMAC synchronization protocol
  - explanation of possibility of loss of synchronization

5. **A Timed Automata Approach to System Level Specification: The Self-Balancing Scooter Case**

   - Goal: present a realistic case study with UPPAAL modeling and model checking of some simple properties, illustrating chapters 2 and 3.
   - Length: 10 pages
   - Authors: *Bert Bos*, Jiansheng Xing, Teun van Kuppeveld
   - Topics:
     - This chapter focuses on system level design, in particular the interaction between user and system. System behaviour is modeled, nominally as well as for non-nominal situations. The design of a self-balancing scooter will serve as case study subject. We demonstrate that modeling of system specifications, which are generally written as text documents supported by figures, unveils issues that were not foreseen in the specifications, but that are important for the user behaviour. Making a model using timed automata is shown to be a powerful means to complete the system specification and forms a solid starting point to implement the system behaviour, which reduces the development cost.
   - Note: risky, since work is not yet finished. No alternative since this chapter can be left out without problem: The Chapter about Model-Checking the gMAC Synschronization Protocol illustrates more or less the same aspects, only not on system level.

6. **Schedulability Analysis**

   - Goal: present principles of schedulability analyis as model-checking problem with UPPAAL
   - Length: 15 pages
   - Authors: *Kim Larsen*, AAU
   - Sources: A. David et al., "Model-based framework for schedulability analysis using UPPAAL 4.1".
   - Topics:
     - scheduling and schedulability problem
     - principles and issues of schedulability analysis
     - schedulability analysis as model checking
     - using UPPAAL with the timed automata schedulability template

7. **Schedulability Analysis using UPPAAL: The Herschel/Planck Satellite Software Case**

- Goal: present TERMA case as realistic case study for schedulability analysis.
- Length: 10 pages
- Authors: *Jacob Illum*, Kim Larsen, Marius Mikucionis
- Sources: TERMA case study
- Topics:
  - This chapter considers the ACC ASW software, a system for satellite attitude and orbit control used within the Herschel and Planck satellite systems. Schedulability analysis for this system was so far performed using classical worst-case response-time analysis. The Quasimodo model-based approach allowing schedulability analysis to be carried out as a model-checking problem, was applied to show that the classical approach is over-pessimistic. In this approach, tasks, resources, and scheduling principles are modeled as timed (stop-watch) automata that make it possible to perform more precise schedulability analysis leading to the conclusion that all configurations of the software are schedulable, which was not possible using the classical approach. This case study illustrates that the UPPAAL model checker can be applied for schedulability analysis.

8. **Controller Synthesis**

- Goal: present principles of controller synthesis using timed games and UPPAAL TIGA
- Length: 15 pages
- Authors: *Jean-Francois Raskin*, Franck Cassez, Pierre-Alain Reynier, Kim Larsen
- Sources: papers timed games, HYDAC paper
- Topics:
  - In this chapter, we introduce in more general terms the ideas underlying controller synthesis in the untimed and timed setting.

9. **Using UPPAAL TIGA for Timed Controller Synthesis: The Hydac Case**

- Goal: present the HYDAC case study as illustration of successful controller synthesis
- Length: 10 pages
- Authors: *Jean-Francois Raskin*, Franck Cassez, Pierre-Alain Reynier, Kim Larsen
- Sources: HYDAC paper
- Topics:
  - In this chapter, we describe the Hydac case study.
  - Use of UPPAAL TIGA
  - Also use of Simulink, Phaver?

10. **Probabilistic Analysis of Embedded Systems**

    - Goal: present principles of probabilistic analysis and use of probabilistic tools
    - Length: 15 pages
    - Authors: *Joost-Pieter Katoen*, Holger Hermanns
    - Sources: papers
    - Topics:
      - This chapter will focus on the MODEST language with its accompanying tool support. The main focus of the chapter will be on what system aspects we can model, and analyze, and e.g., not on the semantic issues involved.

11. **Probabilistic Analysis of Embedded Systems: Energy Consumption in the Chess WSN**

    - Goal: present energy consumption in the Chess WSN as an illustration of successful application of probabilistic analysis
    - Length: 10 pages
    - Authors: *Joost-Pieter Katoen*, Holger Hermanns
    - Sources:
    - Topics:
      - Energy Consumption in the Chess WSN
      - Use of MODEST tool
      - Also Zigbee as example?

12. **Model-Based Testing**

    - Goal: present principles and practice of model-based testing
    - Length: 15 pages
    - Authors: *Jan Tretmans*, Brian Nielsen, Mariëlle Stoelinga
    - Sources: papers
    - Topics:
      - automatic generation of tests from models
      - steps in model-based testing
      - on-line and off-line testing
      - real-time testing
      - tools for model-based testing: UPPAAL TRON, JTORX, TORXAKIS

13. **Model-Based Protocol Conformance Testing: The Case of the Chess Wireless Sensor Network Node**

    - Goal: present the the conformance test of the Chess Wireless Sensor Network Node as an example of model-based testing.
    - Length: 10 pages
    - Authors: *Marcel Verhoef*, Jan Tretmans, Axel Belinfante
    - Sources:
    - Topics:
      - Wireless Sensor Network node
      - Conformance testing
      - Model of gMAC protocol entity
      - Test architecture
      - Use of test tools UPPAAL TRON, JTORX, TORXAKIS
      - test set-up on host and target, in simulated time and real-time
    - Note: High risk, since experiments are not yet finished. Alternatives: Passport testing, model-based design and testing of Bus Protocol with JTORX (UT); see next chapter.

14. **Formal Engineering: Model-Based Design and Testing of a Bus Protocol with JTORX**

    - Goal: present a trajectory of formal enineering from design to model-based testing
    - Length: 10 pages
    - Authors: *Mariëlle Stoelinga*
    - Sources: MSc. report

15. **Perspective**

    - Goal: concluding remarks
    - Length: 5 pages
    - Authors: *Kim Larsen*
    - Topics:
      - what has been achieved
      - perspective for quantitative, model based analysis of embedded systems
      - what has not been achieved, open issues, further research (what could be a next project)
      - obtaining additional information, further reading

5 Potential Additional Chapters

Proposed or potential chapters not (yet) included:

1. **From POOSL to UPPAAL: The ASML Case**
   - Authors: Jiangsheng Xing (UT)
2. **Adaptive Scheduling of Data Paths: The OCE Case**
   - Authors: Georgeta Igna, Frits Vaandrager (RU)
3. **Model-Based Testing with JTORX and CADP: Logfile Verification of OCE Printers using CADP**
   - Authors: Mariëlle Stoelinga