

CM0604 Embedded Systems Specification and Design

Embedded Systems Modelling Image

This module is designed to serve as an introduction to formal and object-oriented methods for specifying, designing and reasoning about embedded systems. The module uses state-of-the-art tools and techniques, including VisualState, SPIN and UPPAAL. It emphasises the importance of constructing and analysing models in the early stages of system development.

The main ideas are communicated in a series of 12 lectures. The laboratory sessions are used to put these ideas into practice in a variety of modelling and analysis exercises. By the end of the module, you should be able to take an informal specification of an embedded system, construct a formal/object-oriented model of (part of) a design and reason about the likely behaviour of the system. The understanding gained in this process can be invaluable in the implementation of a reliable system.

The principles and techniques taught in this module are used extensively in the modules CM034 Industrial Case Project and EN0728 Embedded Systems Case Project.

1. News

02-11-2011, 12.00

Assignment Specification Published [[Specification](#)]

The assignment specification is now available. The due date for submission of the work is **15th December 2011 @ 14.00**. Please address any queries about the specification to [David Kendall](#).

02-11-2011, 12.00

Solutions to exercises (1-4)

Solutions for (most of) the exercises for weeks 1 to 4 are now available. See week 4 practical below. Any questions to [David Kendall](#).

25-09-2011, 15.00

Start of teaching

Teaching starts in the week commencing Mon 26 September 2011.

2. Module Team

Module Tutor[David Kendall](mailto:david.kendall@northumbria.ac.uk) david.kendall@northumbria.ac.uk**Lecturer**[David Gee](mailto:david.gee@northumbria.ac.uk) david.gee@northumbria.ac.uk**3. Teaching Arrangements****Lecture** Tue 11.00 - 13.00 EB A009**Lab/Seminar** Mon 09.00 - 11.00 PB Lab F1**Lab/Seminar** Tue 14.00 - 16.00 PB Lab F1**4. Synopsis**

The aim of the module is to provide an understanding of the theory and practice of modelling, specification and analysis in the design of embedded systems.

On completion of this module, students will be able to:

1. Evaluate the advantages and disadvantages of the application of formal and object-oriented methods in the development of embedded systems, and justify their use where appropriate.
2. Construct and evaluate formal and object-oriented models of a variety of small embedded systems.
3. Compose formal specifications of system properties and analyse system models with respect to them.
4. Identify, apply and evaluate appropriate software tools to support the construction and analysis of formal and object-oriented system models and properties.

5. Teaching Plan

The following is a *provisional* guide to the organisation of the module for this year. These arrangements are subject to change during the course of the module.

Week	W/c	Lecture	Practical
1	26-Sep	Module overview. Introduction to Embedded Systems Specification & Design; Formal Modelling tools. [Slides] <i>Required Reading:</i> [RUS95] , [STA95]	The SPIN modelling tool and its PROMELA modelling and specification language [Slides] [Exercises]
2	03-Oct	Modelling and Specification with	More PROMELA. Checking safety

		Finite State Automata Part I [Slides]	properties with SPIN; [Slides] [Exercises]
3	10-Oct	Modelling and Specification with Finite State Automata Part II [Slides] [Mutex model]	Checking liveness properties with SPIN; [Exercises]
4	17-Oct	Temporal logic. Specification patterns. [Slides] <i>Required Reading:</i> [DAC99] [SAC03]	Temporal Logic Specification and Analysis [Exercises] [Outline solutions (1-4)] [Case study]
5	24-Oct	Introducing VisualState: What is VisualState? VisualState's version of statecharts. The design approach. Example: home security system. [Slides]	Introduction to VisualState. Exercise: Car Compartment Light. [Exercises]
6	31-Oct	Further Visual State: Verification & Validation. Using the verification tools and the Validator. [Slides]	Further VisualState exercises, including Validation & Verification. [Exercises]
7	07-Nov	Visual State Implementation & Prototyping Tools. [Slides] [Example]	Further VisualState exercises, including code generation. [Exercises]
8	14-Nov	Overview of Structured & Object-Oriented Methods: Why use structured and OO methods? Standard methods: Ward & Mellor, UML. Extensions for Real-Time Systems. State transition diagrams. Reducing STD complexity. [Slides]	Applying methods to a simple example.
9	21-Nov	Timed Automata and Timed Transition Systems [Slides]	Introduction to UPPAAL [Exercises]

		<i>Required Reading:</i> [Uppaal Tutorial] [LEV95]	
10	28-Nov	Networks of Timed Automata. Specifying real-time properties [Slides] <i>Required Reading:</i> [HOL01]	Modelling a box sorter [Exercises] [Box sorter model]
11	05-Dec	Principles of automata-theoretic verification of real-time systems [Slides]	Modelling and verifying a real-time communication protocol [Exercises] [TTP model]
12	12-Dec	Example: Uppaal model of controller area network. Revision [Slides]	The Vikings problem [Exercises] [Vikings model]

Note:

In addition to the taught sessions, you are expected to undertake independent and directed learning. On average, you should be spending 10 to 12 hours per week on this module.

6. Assessment

Summative assessment comprises:

1. a substantial exercise in specification, design and analysis in which students will be required to demonstrate a practical ability to apply appropriate languages, techniques and tools, and
 2. an examination in which students will be required to demonstrate a grasp of fundamental concepts and a critical understanding of a variety of approaches to the specification and design of embedded systems.
- [Assignment Specification](#)

Past Exam Papers: [\[2007-08\]](#), [\[2008-09\]](#), [\[2009-10\]](#)

7. Recommended Reading

There is no essential text for this module. Below is a list of references to texts and other resources that are relevant to the content of the module. You may find many of them useful and/or interesting. In addition, you are required to undertake some background reading about the use of formal methods in the development of embedded and critical

systems. [[More](#)]

- Alavi, H. et al., [SPEC Patterns](#), 2005
- Barland, I.; Vardi, M.; Greiner, J. Model Checking Concurrent Programs, Connexions Web site. <http://cnx.org/content/col10294/1.3/>, Oct 27, 2005.
- Behrmann, G., David, A., and Larsen, K. [A Tutorial on Uppaal](#), Department of Computer Science, Aalborg University, Denmark, 2004
- Berard, B. et al., *Systems and Software Verification: Model Checking Techniques and Tools*, Springer Verlag, 2001
- Cooling, J.E. *Software Engineering for Real-Time Systems*. Addison-Wesley, 2003.
- Douglass, B.P. *Real-Time UML*. Addison-Wesley, 1999.
- Douglass, B.P. *Doing Hard Time*. Addison-Wesley, 1999.
- Gomaa, H. *Designing Concurrent Distributed and Real Time Applications with UML*. Addison-Wesley, 2000.
- Harel, D. [On Visual Formalisms](#) and [Biting the Silver Bullet: Toward a Brighter Future for System Development](#)
- Holzmann, G., *The SPIN Model Checker: Primer and Reference Manual*, Addison Wesley, 2003, ISBN 0321228626
- Huckle, T. [Collection of Software Bugs](#), 2004
- Huth, M., Ryan, M., *Logic in Computer Science: Modelling and Reasoning about Systems* 2nd edition, Cambridge University Press, 2004.
- Technical journals and conference papers, e.g. Computer Aided Verification (CAV), Tools and Algorithms for the Construction of Systems (TACAS)

8. Other resources

[SPIN Logo](#)

[SPIN](#)

"Spin is a popular open-source software tool, used by thousands of people worldwide, that can be used for the formal verification of distributed software systems. The tool was developed at Bell Labs in the original Unix group of the Computing Sciences Research Center, starting in 1980. The software has been available freely since 1991, and continues to evolve to keep pace with new developments in the field. In April 2002 the tool was awarded the prestigious System Software Award for 2001 by the ACM." [[Link](#)]

The main focus of this module, in weeks 1-4, is on the theory and practical application of this tool to the modelling and analysis of embedded systems. It is essential that you use the tool extensively on a variety of lab exercises. The tool is installed in PB Lab F1 on the Linux machines. It is also available free for use on your own machines -- see the [download](#) instructions.

- [Getting Started with SPIN](#)
- [Basic SPIN Manual](#)
- [Concise PROMELA Reference](#)

UPPAAL

"Uppaal is an integrated tool environment for modeling, validation and verification of real-time systems modeled as networks of timed automata, extended with data types (bounded integers, arrays, etc.).

The tool is developed in collaboration between the Department of Information Technology at Uppsala University, Sweden and the Department of Computer Science at Aalborg University in Denmark." [[Link](#)]

The main focus of this module, in weeks 9-12, is on the theory and practical application of this tool to the modelling and analysis of embedded systems. It is essential that you use the tool extensively on a variety of lab exercises. The tool is installed in PB Lab F1 on the Linux machines. It is also available free for use on your own machines -- get the [local Uppaal installation archive](#). There is a [tutorial](#) that gives a good introduction to the theory and practice of modelling with timed automata and analysis with the Uppaal tool.

visualSTATE

Get the [local copy](#) of the 20 state evaluation version of visualSTATE. This is a self-installing executable for use with Windows XP. It should be adequate for building the models in the exercises and assignments for this module.

9. Professionalism, ethics and standards

- [ACM Code of Conduct](#)
- [BCS Code of Conduct and Good Practice](#)
- [IEEE Code of Ethics](#)
- [Teaching engineering ethics with case studies](#)
- [Ethical issues in empirical studies of software engineering](#)
- European Organisation for Civil Aviation Equipment. [Software Considerations in Airborne Systems and Equipment Certification \(ED-12B/DO-178B\)](#), EUROCAE, 17 rue Hamelin, F-75783 Paris Cedex 16, France, (1992).
- Hennell,M., Woodcock,J. and Woodward,M., [The Safety Integrity Levels Of IEC 61508 And A Revised Proposal](#), Embedded Systems Show, 2006. ([Local copy](#))
- Redmill,F., [An Introduction To The Safety Standard IEC 61508](#), Journal of the System Safety Society, Volume 35, No. 1, 1999
- [Computer Professionals for Social Responsibility](#)