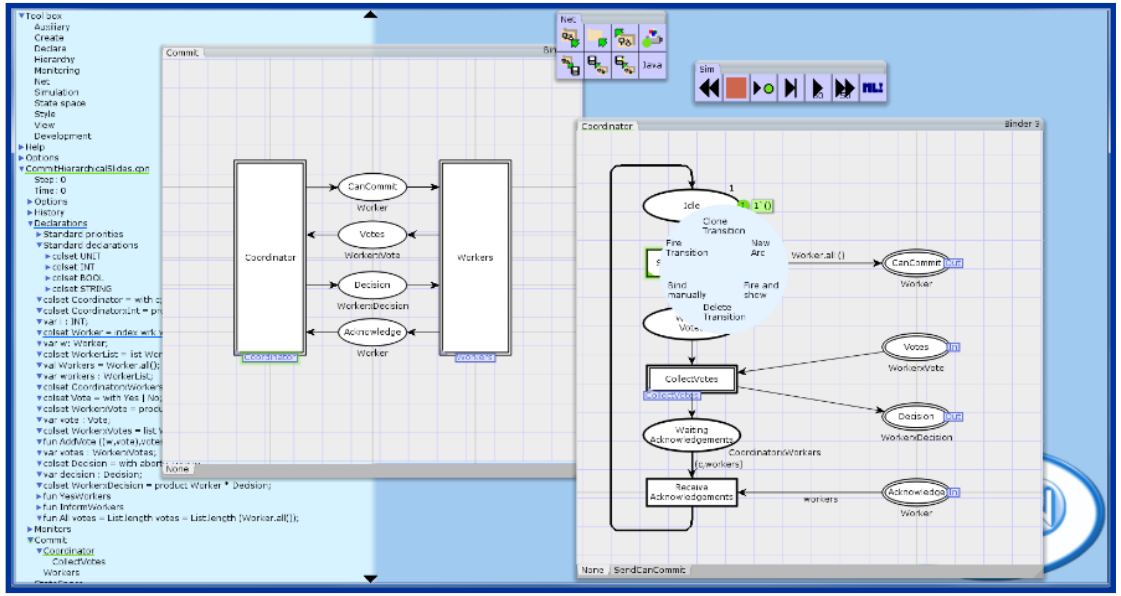
Lectures on

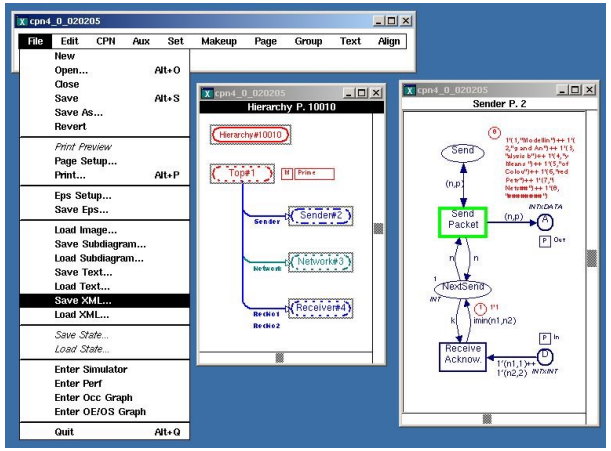
Coloured Petri Nets for Concurrent Software Systems Engineering

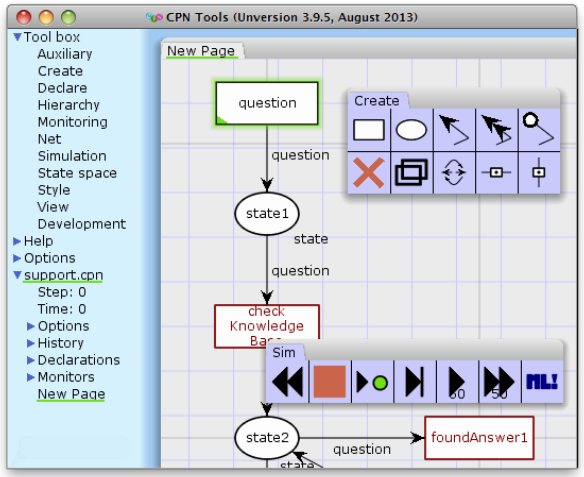


Lars Michael Kristensen
Department of Computer Science, Electrical Engineering, and Mathematical Sciences
Western Norway University of Applied Sciences
Email: lmkr@hvl.no | https://www.hvl.no/person/?user=Lars.Michael.Kristensen

My background [1:3]

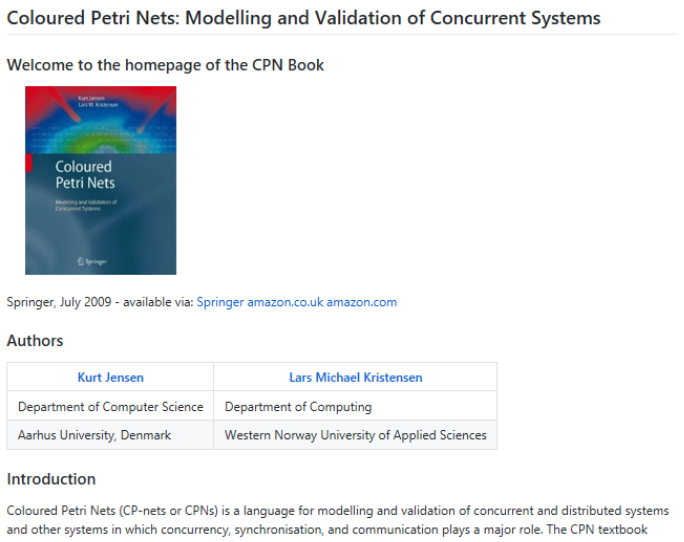
- **2000: PhD from the CPN research centre at Aarhus University (Denmark) on Coloured Petri Nets and software verification**





My background [2:3]

- **2000-2002: Post-doctoral researcher at University of South Australia | Australian Defence and Technology Organisation**
  - **Software tool support for military command and control**
  - **Modelling and implementation of real-time avionics missions systems**
- **2002-2009: Associate professor at Aarhus University**
  - **Capacity planning for web servers with Hewlett-Packard**
  - **Development of mobile ad-hoc network protocols and IPv6 with Ericsson Telebit**



Coloured Petri Nets: Modelling and Validation of Concurrent Systems

Welcome to the homepage of the CPN Book

Springer, July 2009 - available via: Springer amazon.co.uk amazon.com

Authors

Kurt Jensen	Lars Michael Kristensen
Department of Computer Science	Department of Computing
Aarhus University, Denmark	Western Norway University of Applied Sciences

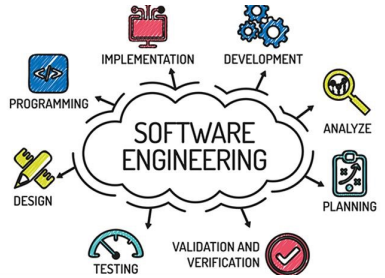
Introduction

Coloured Petri Nets (CP-nets or CPNs) is a language for modelling and validation of concurrent and distributed systems and other systems in which concurrency, synchronisation, and communication plays a major role. The CPN textbook

- **K. Jensen and L.M. Kristensen. Coloured Petri Nets: Modelling and Validation of Concurrent Systems, Springer, 2009.**
- **Book website: www.cpnbook.org**

My background [3:3]

- **Since 2009: Professor of computer science and software engineering at Western Norway University of Applied Sciences**
  - **Establishment of a PhD programme in Computer Science and a master programme in Applied Computer Science and Engineering**
  - **Teaching and research: network technology and distributed systems, IoT and cloud software systems, model-driven software engineering and verification**



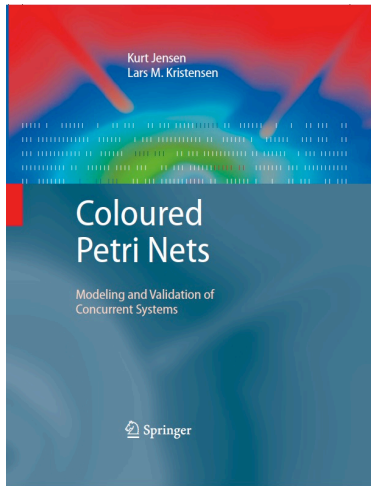
Outline

- **DAY 1: Theory-Tool**
  - Motivation and overview of the Coloured Petri Nets modelling language
  - Background on Coloured Petri Nets: From Place/Transition Nets to Coloured Petri Nets
  - Syntax and semantics of the basic constructs of Coloured Petri Nets
  - Modules for hierarchical structuring of large Coloured Petri Nets models
- **DAY 2: Case Study**
  - Two examples on the application of Coloured Petri nets within smart software systems
  - Installation and getting started with CPN Tools
  - Hands-on experiments with simulation of CPN models in CPN Tools
  - Hands-on experiments with editing and construction of CPN models in CPN Tools

You are also most welcome to work on your own case study!

Do not hesitate to ask questions along the way!

Resources



K. Jensen and L.M. Kristensen. Coloured Petri Nets: Modelling and Validation of Concurrent Systems, Springer, 2009.

www.cpnbook.org

Practical use of CPN Tools is extensively documented at www.cpntools.org


Implementing Coloured Petri Nets
Using a Functional Programming Language

LARS MICHAEL KRISTENSEN*
SØREN CHRISTENSEN

Department of Computer Science, University of Aarhus, Aabogade 34, DK-8200 Aarhus N, Denmark

Abstract. Coloured Petri Nets (CPNs) are a graphically oriented modelling language for concurrent systems based on Petri Nets and the functional programming language Standard ML. Petri Nets provide the primitives for modelling concurrency and synchronisation. Standard ML provides the primitives for modelling data manipulation and for creating compact and parameterisable CPN models.

Functional programming and Standard ML have played a major role in the development of CPNs and the CPN computer tools supporting modelling, simulation, verification, and performance analysis of concurrent systems. At the modelling language level, Standard ML has extended Petri Nets with the practical expressiveness required for modelling systems of the size and complexity found in typical industrial projects. At the implementation level, Standard ML has been used to implement the formal semantics of CPNs that provide the theoretical foundation of the CPN computer tools.

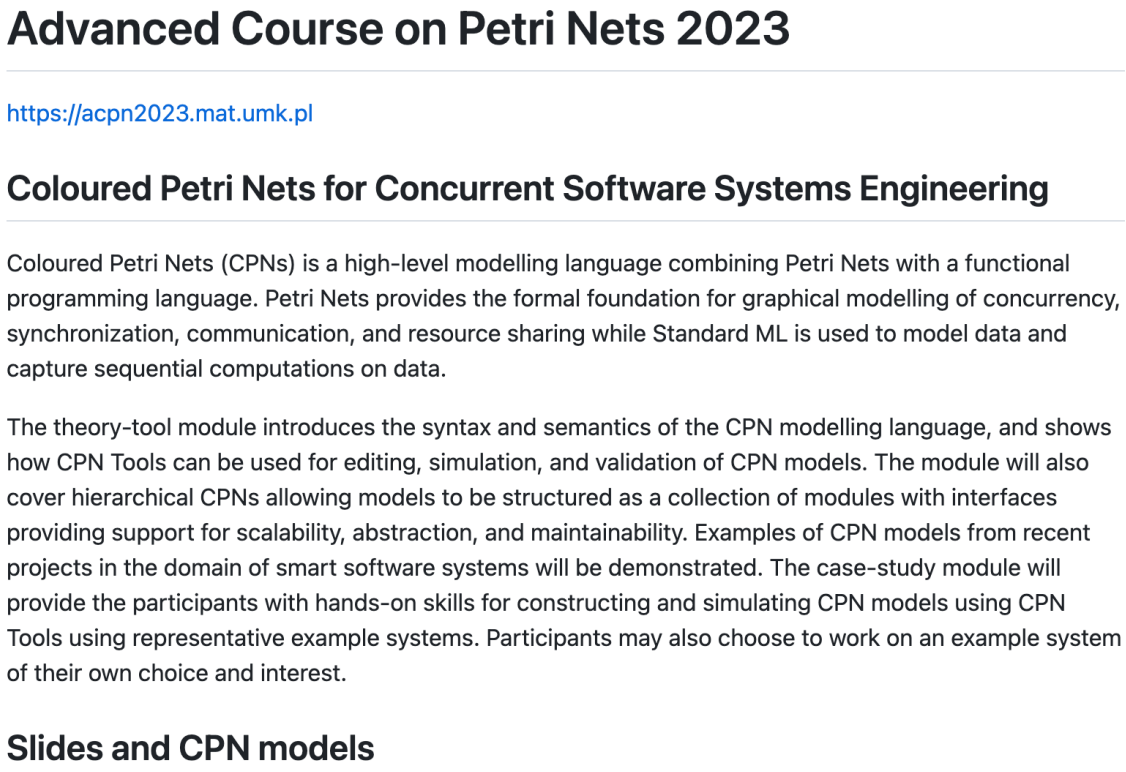
This paper provides an overview of how functional programming and Standard ML are applied in the CPN modelling language and the supporting computer tools. We give a detailed presentation of the key algorithms and techniques used for implementing the formal semantics of CPNs, and we survey a number of case studies where CPNs have been used for the design and analysis of systems. We also demonstrate how the use of a Standard ML programming environment has allowed Petri Nets to be used for the implementation of systems.

Keywords: distributed and concurrent computation, implementation techniques, programming environments and tools, Coloured Petri Nets, high-level Petri Nets, Petri Nets

- **Research papers on Coloured Petri Nets**
  - K. Jensen and L.M. Kristensen. Coloured Petri Nets: A Graphical Language for Modelling and Validation of Concurrent Systems. Communications of the ACM, Vol. 58, No. 6, pp. 61-70, 2015.
  - K. Jensen, L.M. Kristensen, L. Wells. Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems. Intl. Journal on Software Tools for Technology Transfer, Vol. 9, pp. 213-254, Springer, 2007.
  - L.M. Kristensen and S. Christensen: Implementing Coloured Petri Nets using a Functional Programming Language. In Higher-order and Symbolic Computation, Vol. 17, pp. 207-243, 2004.

Course material

- **Slides, models, and papers are available via the github repository at**



Advanced Course on Petri Nets 2023

https://acpn2023.mat.umk.pl

Coloured Petri Nets for Concurrent Software Systems Engineering

Coloured Petri Nets (CPNs) is a high-level modelling language combining Petri Nets with a functional programming language. Petri Nets provides the formal foundation for graphical modelling of concurrency, synchronization, communication, and resource sharing while Standard ML is used to model data and capture sequential computations on data.

The theory-tool module introduces the syntax and semantics of the CPN modelling language, and shows how CPN Tools can be used for editing, simulation, and validation of CPN models. The module will also cover hierarchical CPNs allowing models to be structured as a collection of modules with interfaces providing support for scalability, abstraction, and maintainability. Examples of CPN models from recent projects in the domain of smart software systems will be demonstrated. The case-study module will provide the participants with hands-on skills for constructing and simulating CPN models using CPN Tools using representative example systems. Participants may also choose to work on an example system of their own choice and interest.

Slides and CPN models

https://github.com/lmkr/cpncourse/blob/master/acpn/README.md