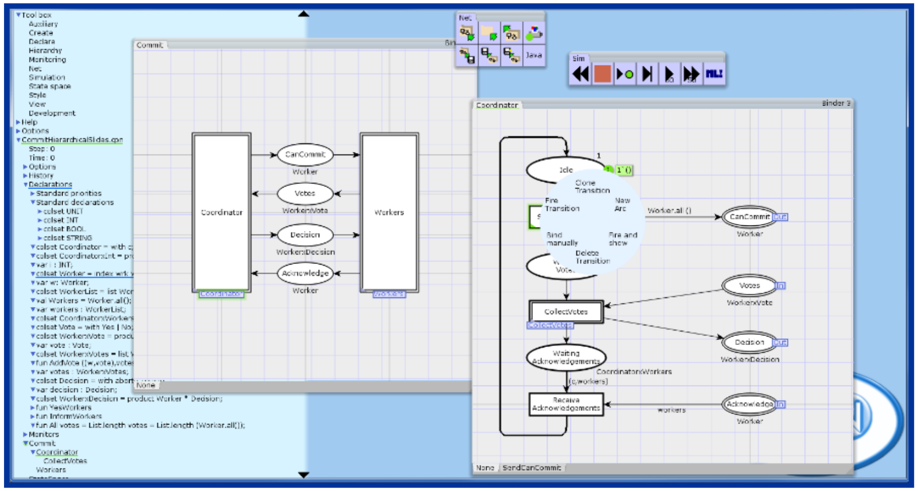
Lecture 1

Overview of Coloured Petri Nets and CPN Tools



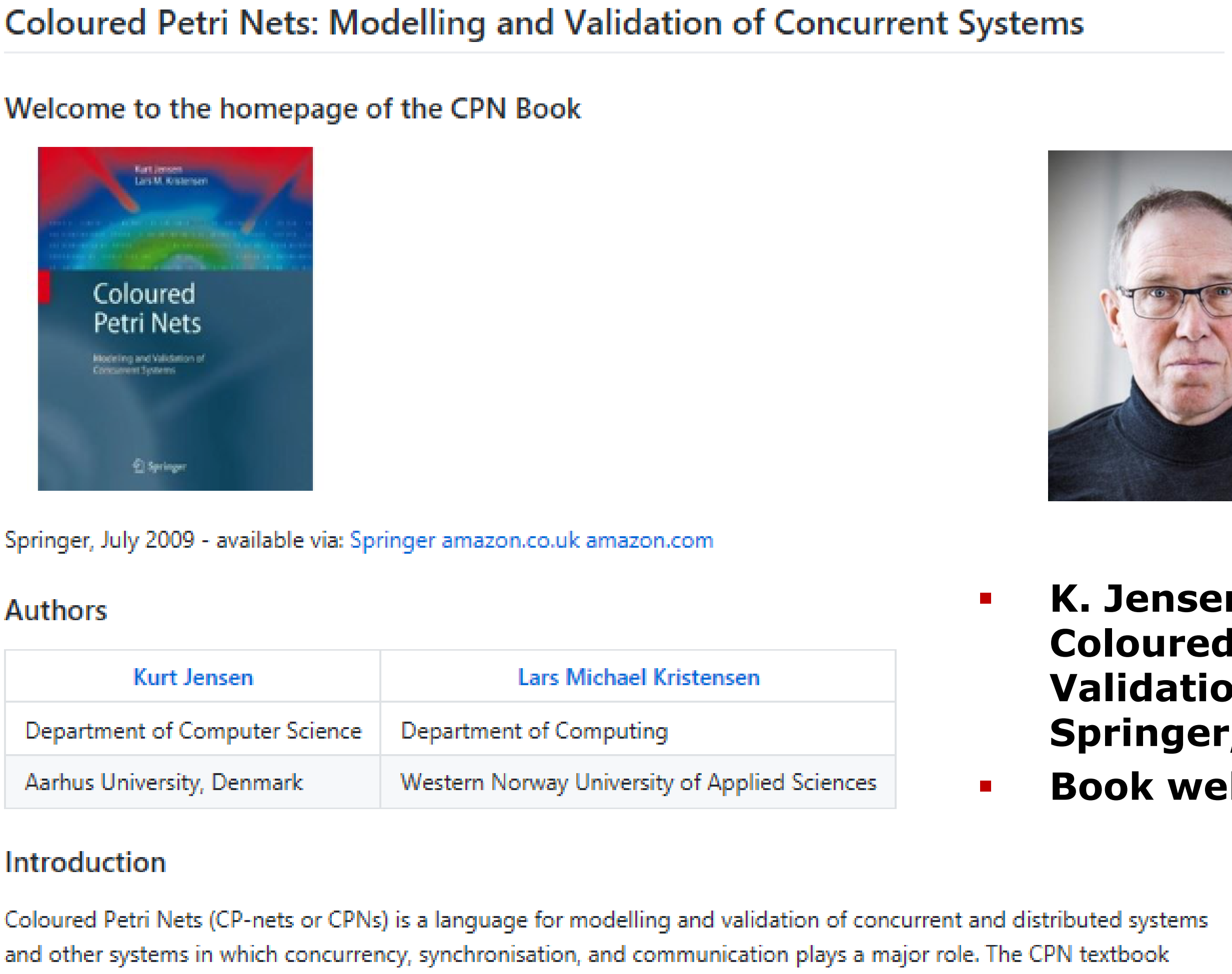
```
colset Coordinator = with c;
colset CoordinatorxInt = pr
var i : INT;
colset Worker = index wrk w
var w: Worker;
colset WorkerList = list Wor
val Workers = Worker.all();
var workers : WorkerList;
colset CoordinatorxWorkers
colset Vote = with Yes | No;
colset WorkerxVote = produ
var vote : Vote;
colset WorkerxVotes = list V
fun AddVote ((w,vote),votes
var votes : WorkerxVotes;
colset Decision = with abort
var decision : Decision;
colset WorkerxDecision = product Worker * Decision;
fun YesWorkers
fun InformWorkers
fun All votes = List.length votes = List.length (Worker.all());
```

Lars Michael Kristensen
Department of Computing, Mathematics, and Physics
Western Norway University of Applied Sciences
Email: lmkr@hvl.no / WWW: home.hib.no/ansatte/lmkr

My Background

- 2000: PhD from the CPN research centre at Aarhus University (DK) on Coloured Petri Nets and software verification.
- 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 protocols for IPv6 with Ericsson Telebit
- Since 2009: Professor of computer science and software engineering at Western Norway Univ. of Applied Sciences
  - Establishment of a PhD programme in Computer Science: Software Engineering, Sensor Networks and Engineering Computing [http://ict.hvl.no]
  - T&R: programming, network technology and distributed systems, internet-of-things, model-driven software engineering and verification

CPN Textbook



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

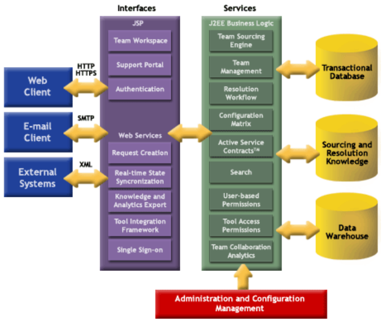
Introduction

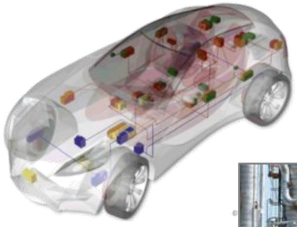
Coloured Petri Nets (CP-nets or CPNs) is a language for modelling and validation of concurrent and distributed systems and other systems in which concurrency, synchronisation, and communication plays a major role. The CPN textbook

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

Concurrent Systems

- **The vast majority of software systems today can be characterised as concurrent systems**
  - Structured as a collection of concurrently executing software components and applications (parallelism)
  - Operation relies on communication, synchronisation, and resource sharing





Internet protocols, cloud, IoT, web-based applications

Multi-core platforms and multi-threaded software

Automation systems and networked control systems

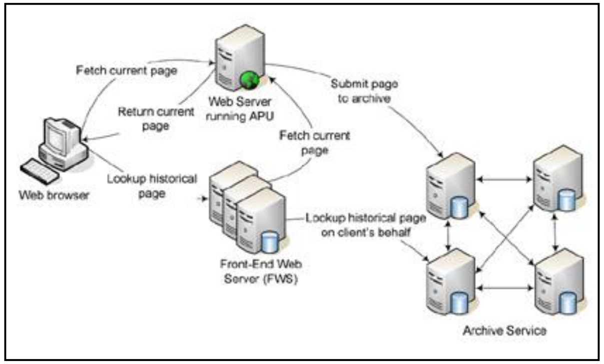
Complex Behaviour

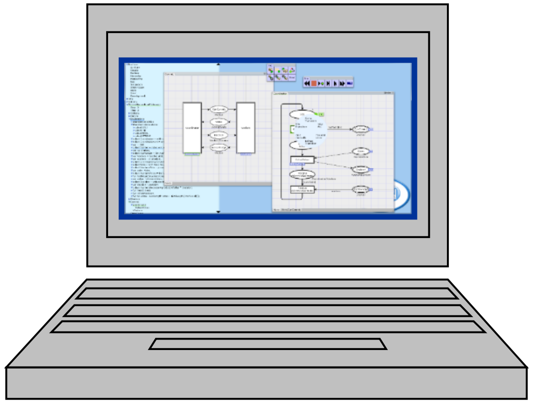
- **The engineering of concurrent systems is challenging due to their complex behaviour**
  - Concurrently executing and independently scheduled components
  - Non-deterministic and asynchronous behaviour (e.g., timeouts, message loss, external events, …)
  - Almost impossible for software developers to have a complete understanding of the system behaviour
  - Software testing is challenging and reproducing errors is often difficult

- **Methods to support the engineering of reliable concurrent systems are highly relevant**

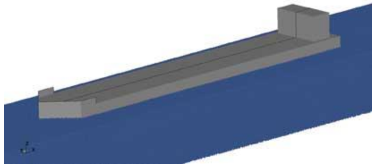
Modelling

- One way to approach the challenges posed by concurrent systems is the construction of models.
- A model is an abstract representation which can be manipulated by a computer software tool





- Modelling is used in most engineering disciplines

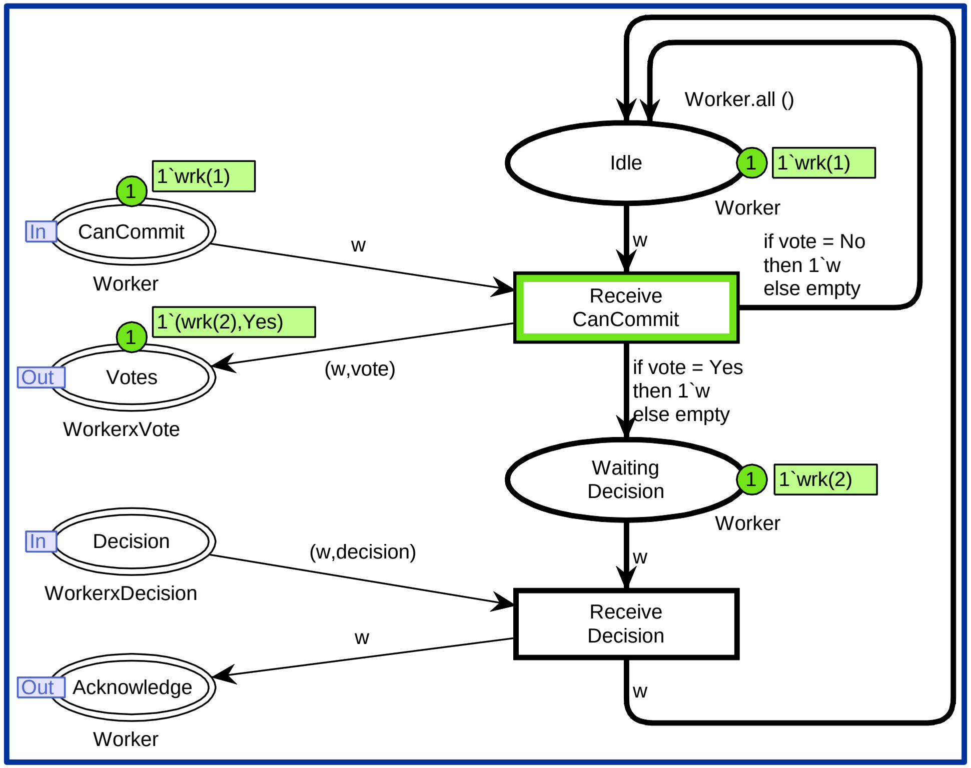


Why Modelling?

- **Benefits of constructing executable models**
  - Insight into the design and operation of the system
  - Completeness: results in a more complete design
  - Correctness: reveal errors and ambiguities in the design phase
- **Abstraction – validation using high-level and domain-specific concepts in development.**
- **Reliability – testing and verification and prior to implementation and deployment**
  - Functional properties (e.g., deadlocks, timing requirements,…)
  - Performance properties (e.g., delay, throughout, scalability,…)
- **Productivity - models can be used (directly or indirectly) as a basis for implementation.**

Coloured Petri Nets (CPNs)

- General-purpose graphical modelling language for the engineering of concurrent systems.
- Combines Petri Nets and a programming language



Petri Nets

graphical notation
concurrency
communication
synchronisation
resource sharing

CPN ML (Standard ML)

data and data manipulation
compact modelling
parameterisable models

High-level Petri Nets

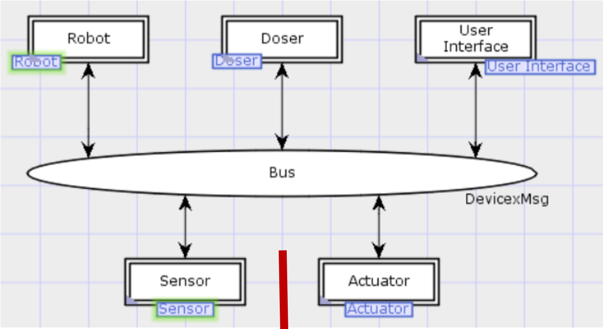
- Petri Nets are divided into low-level and high-level Petri Nets
  - Low-level Petri Nets (such as Place/Transitions Nets) are primarily suited as a theoretical model for concurrency, but are also applied for modelling and verification of hardware systems
  - High-level Petri Nets (such as CP-nets and Predicate/Transitions Nets) are aimed at practical use, in particular because they allow for construction of compact and parameterised models
- High-level Petri Nets is an ISO/IEC standard*
  - The CPN modelling language and the supporting CPN Tools conform to this standard.

* https://www.iso.org/standard/38225.html

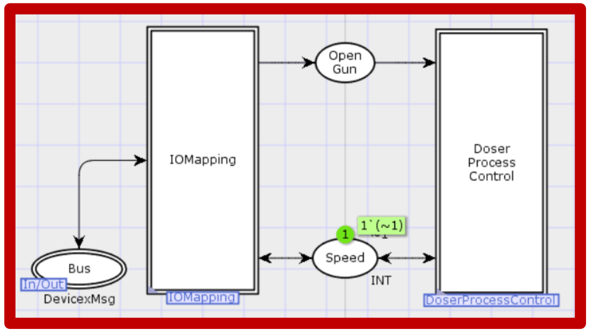
CPN @ Atlas Copco

- **Developing a model-driven software development approach and supporting infrastructure**

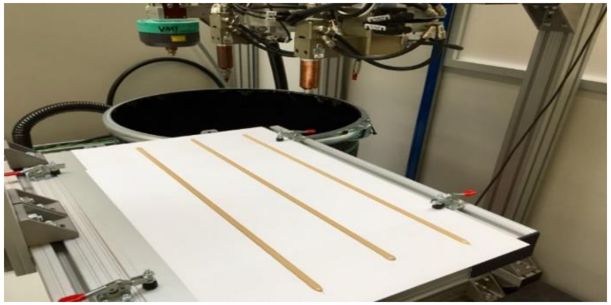
CPN Tools: editing, validation, and verification (design time)

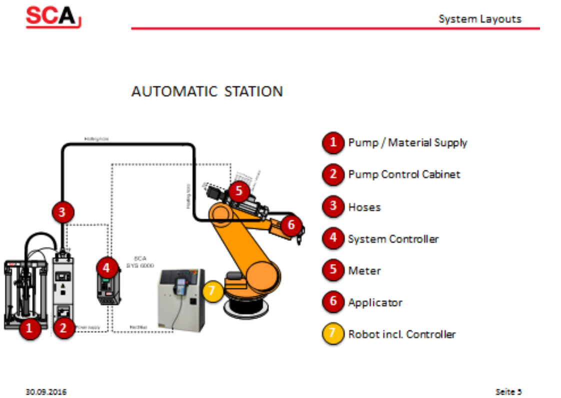


Environment modelling for (non-site) software testing



C++ execution engine for deployment and real-time execution (run-time)

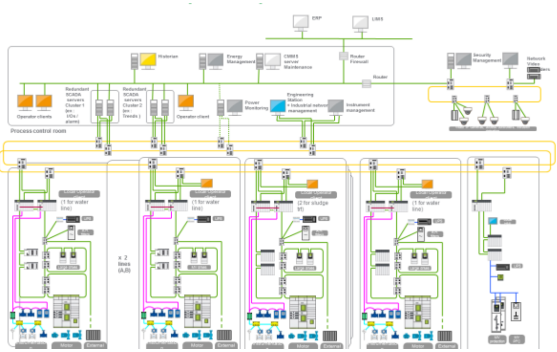




- **The CPN model is directly used as the pump controller software implementation.**

CPN @ Schneider Electric

- **Dependability evaluation and capacity planning of large industrial automation architectures**

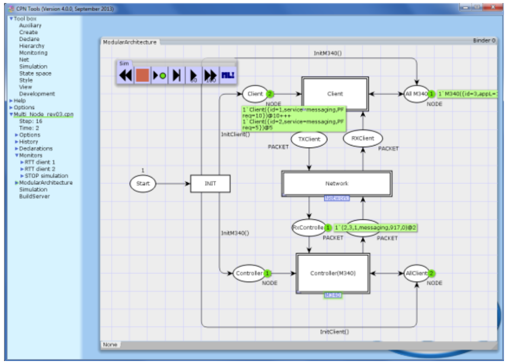


Dependability analysis software tools

Performance - Reliability
Availability - Safety

Modelling

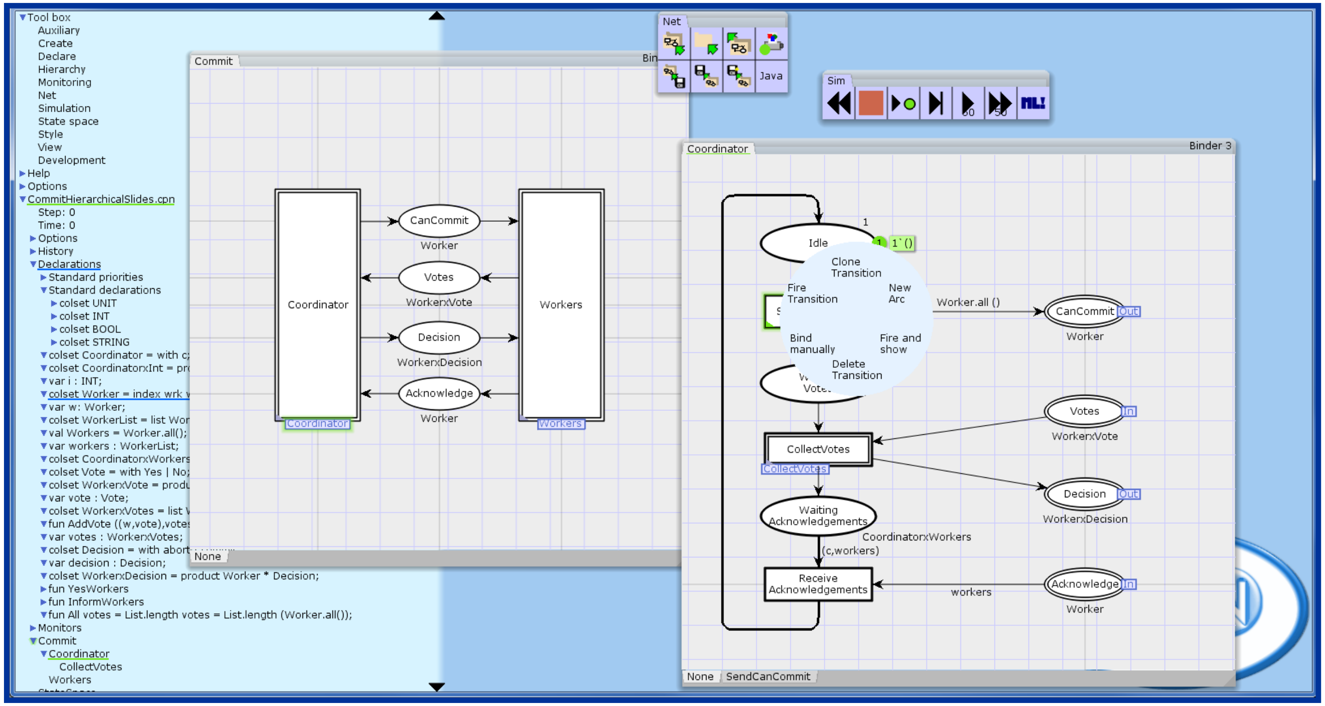
Automated
code generation



Tools for modelling driven engineering

CPN Tools [www.cpntools.org]

- **Practical use of CPNs is supported by CPN Tools**



```
Standard priorities
Standard declarations
  colset UNIT
  colset INT
  colset BOOL
  colset STRING
colset Coordinator = with c;
colset CoordinatorxInt = pr
var i : INT;
colset Worker = index wrk w
var w: Worker;
colset WorkerList = list Wor
val Workers = Worker.all();
var workers : WorkerList;
colset CoordinatorxWorkers
colset Vote = with Yes | No;
colset WorkerxVote = produ
var vote : Vote;
colset WorkerxVotes = list W
fun AddVote ((w,vote),votes
var votes : WorkerxVotes;
colset Decision = with abort
var decision : Decision;
colset WorkerxDecision = product Worker * Decision;
fun YesWorkers
fun InformWorkers
fun All votes = List.length votes = List.length (Worker.all());
```

- **Editing and syntax check**
- **Interactive- and automatic simulation**
- **Verification based on state space exploration**
- **Simulation-based performance analysis**

Western Norway University of Applied Sciences

CPN Tools Demo

- User-interaction with CPN Tools
  - Index and workspace
  - Binders and tool palettes - drag-and-drop
  - Contextual menus - right click

Examples of CPN Tools users

North America

- Boeing
- Hewlett-Packard
- Samsung Information Systems
- National Semiconductor Corp.
- Fujitsu Computer Products
- Honeywell Inc.
- MITRE Corp.,
- Scalable Server Division
- E.I. DuPont de Nemours Inc.
- Federal Reserve System
- Bell Canada
- Nortel Technologies, Canada

Asia

- Mitsubishi Electric Corp., Japan
- Toshiba Corp., Japan
- SHARP Corp., Japan
- Nippon Steel Corp., Japan
- Hongkong Telecom Interactive Multimedia System

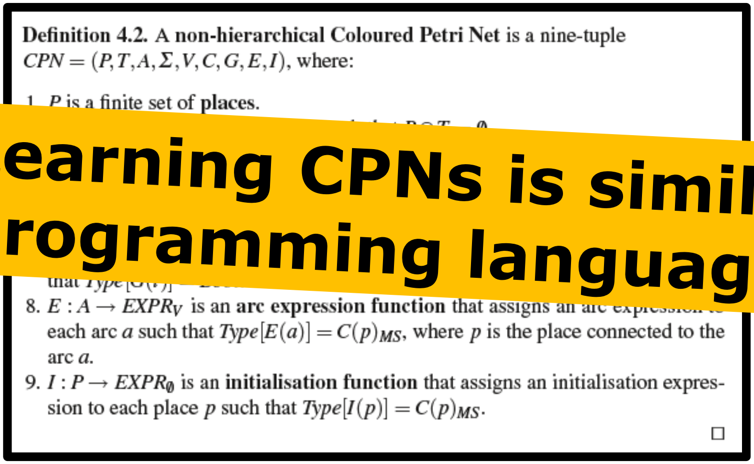
Europe

- Alcatel Austria
- Siemens Austria
- Bang & Olufsen, Denmark
- Nokia, Finland
- Alcatel Business Systems, France
- Peugeot-Citroën, France
- Dornier Satellitensysteme, Germany
- SAP AG, Germany
- Volkswagen AG, Germany
- Alcatel Telecom, Netherlands
- Rank Xerox, Netherlands
- Sydkraft Konsult, Sweden
- Central Bank of Russia
- Siemens Switzerland
- Goldman Sachs, UK

http://cs.au.dk/cpnets/industrial-use/

CPN models are formal

- **The CPN modelling language has a mathematical definition of both its syntax and semantics.**
- **The formal representation is important**
  - Would have been impossible to develop a sound and powerful CPN language without it
  - Provides the foundation for the definition of the behavioural properties and for the formal analysis and verification methods

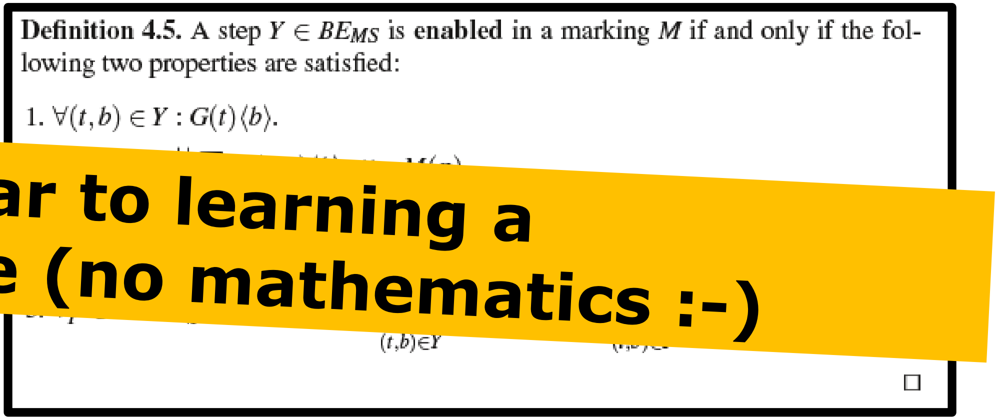


Definition 4.2. A **non-hierarchical Coloured Petri Net** is a nine-tuple $CPN = (P, T, A, \Sigma, V, C, G, E, I)$, where:

1. P is a finite set of **places**.

8. $E : A \rightarrow EXPR_V$ is an **arc expression function** that assigns an arc expression to each arc a such that $Type[E(a)] = C(p)_{MS}$, where p is the place connected to the arc a.
9. $I : P \rightarrow EXPR_\emptyset$ is an **initialisation function** that assigns an initialisation expression to each place p such that $Type[I(p)] = C(p)_{MS}$.

□



Definition 4.5. A step $Y \in BE_{MS}$ is **enabled** in a marking M if and only if the following two properties are satisfied:

1. $\forall (t,b) \in Y : G(t)\langle b \rangle$.

□

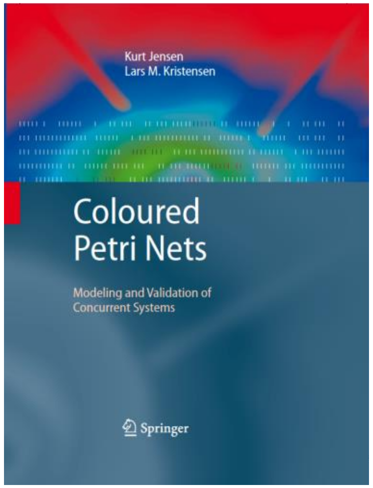
Learning CPNs is similar to learning a programming language (no mathematics :-)

Outline

- **Module I: Modelling and CPN Tools [today]**
  - Motivation and overview of Coloured Petri Nets
  - The syntax and semantics of the basic constructs of the Coloured Petri Nets (CPNs) modelling language
  - Modules for hierarchical structuring of large CPN models
  - Application of CPN Tools for construction and simulation of CPN models
- **Module II: Verification and Applications [tomorrow]**
  - The basic concepts of state spaces and how they are computed
  - Introduce standard behavioural properties of CPNs
  - Checking standard behavioural properties using state spaces
  - A larger example on the industrial use of CPNs and CPN Tools

Do not hesitate to ask questions along the way!

Resources


Kurt Jensen
Lars M. Kristensen

Coloured Petri Nets

Modeling and Validation of Concurrent Systems

Springer

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

1. Introduction

SPECIAL SECTION CPN 04/05

Coloured Petri Nets and CP
of concurrent systems

Kurt Jensen · Lars Michael Kristensen · Lis

Colored Petri Nets: A
Modeling and Vali

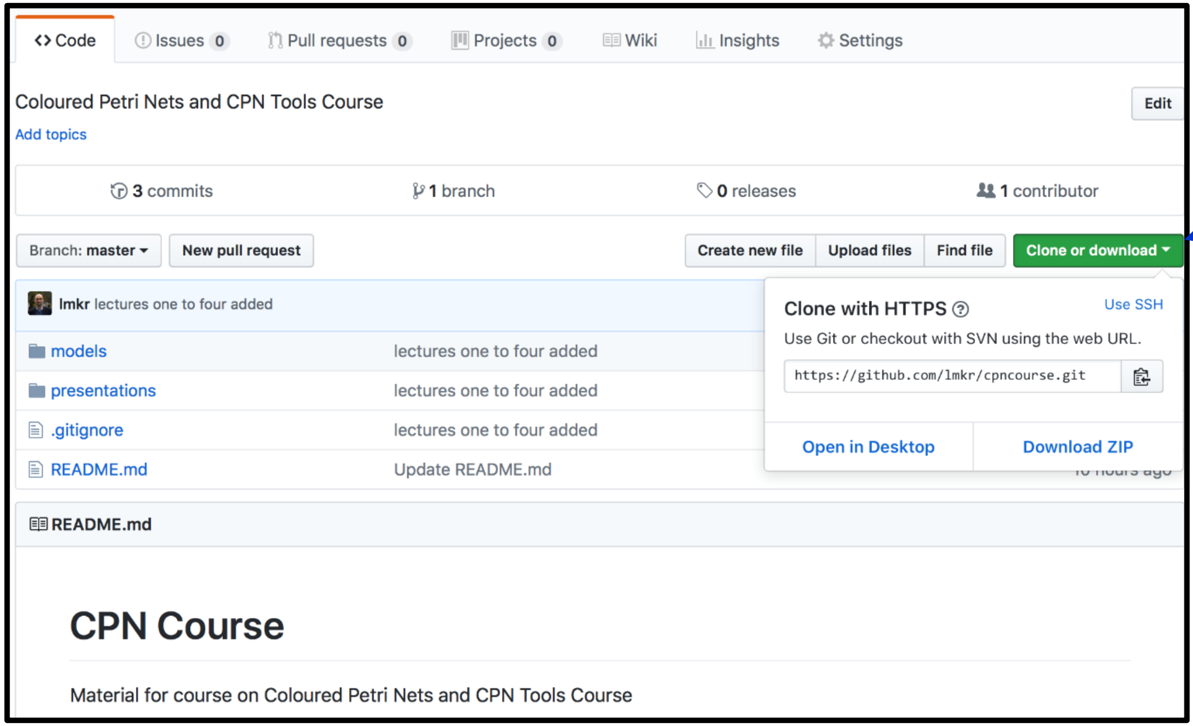
Kurt Jensen
Department of Computer Scie
Aarhus University, Denmar
kjensen@cs.au.dk

ABSTRACT

- **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** https://github.com/lmkr/cpncourse



Coloured Petri Nets and CPN Tools Course

lmkr lectures one to four added

models	lectures one to four added
presentations	lectures one to four added
.gitignore	lectures one to four added
README.md	Update README.md

Clone with HTTPS

Use Git or checkout with SVN using the web URL.

```
https://github.com/lmkr/cpncourse.git
```

README.md

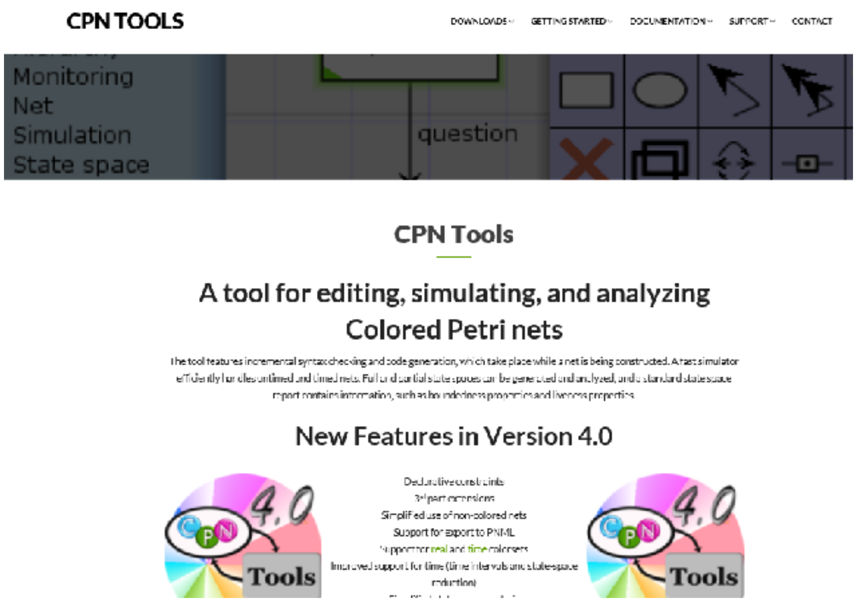
CPN Course

Material for course on Coloured Petri Nets and CPN Tools Course

Clone the git-repository or download as a zip-file

CPN Tools Installation

- CPN Tools can be downloaded and installed via www.cpntools.org



CPN Tools

A tool for editing, simulating, and analyzing Colored Petri nets

The tool features incremental syntax checking and code generation, which take place while a net is being constructed. A fast simulator efficiently handles untimed and timed nets. Full and partial state spaces can be generated and analyzed, and a standard state space report contains information, such as boundedness properties and liveness properties.

New Features in Version 4.0

Declarative constraints
3rd party extensions
Simplified use of non-colored nets
Support for export to PNML
Support for real and time colorsets
Improved support for time (time intervals and state-space reduction)

Running on Mac OS / Linux via a virtual machine or emulator.

- Some installations of Windows required the application to be run as administrator.