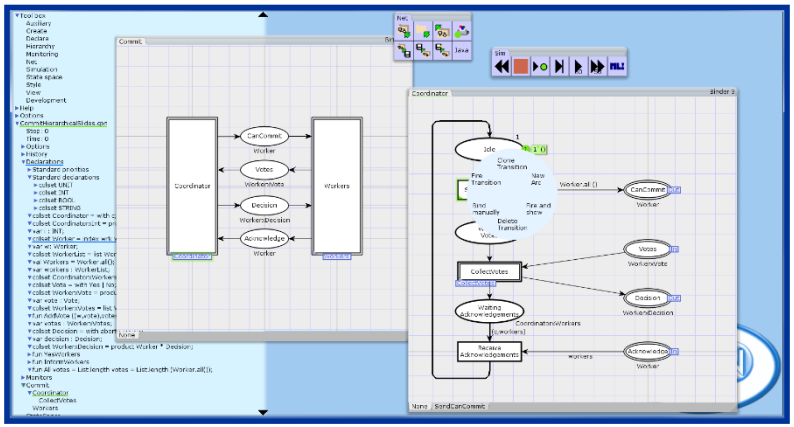
Model-driven Engineering of Concurrent Systems with Coloured Petri Nets

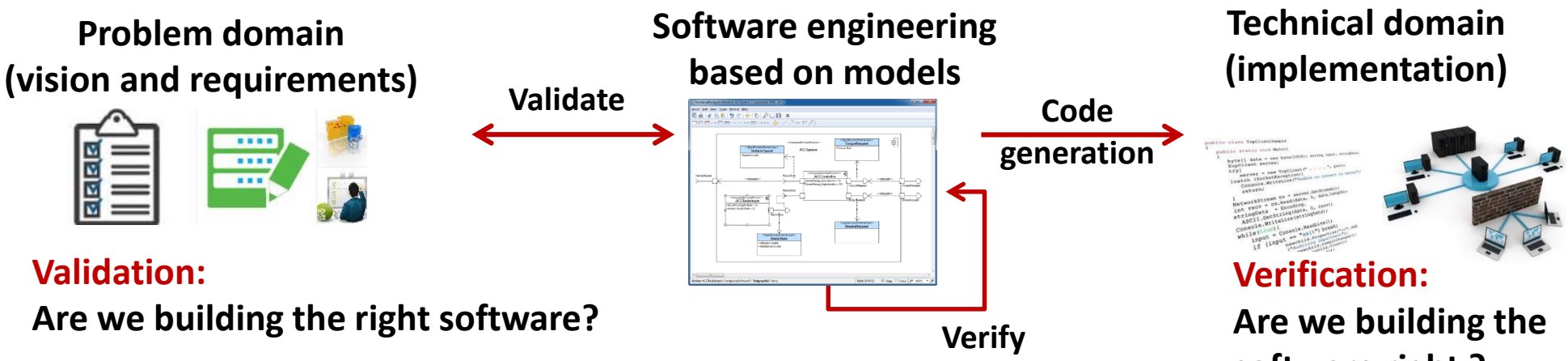


Lars M. Kristensen
Department of Computing
Bergen University College, NORWAY
lmkr@hib.no / home.hib.no/ansatte/lmkr
ICT Engineering: prosjekt.hib.no/ict

HØGSKOLEN I BERGEN
BERGEN UNIVERSITY COLLEGE

Model-driven Engineering (MDE)

- **An prominent approach to software engineering based on the construction of models:**



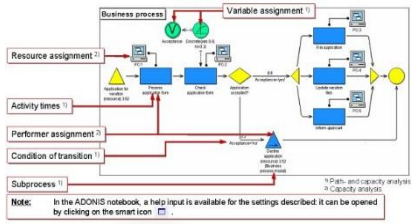
- **Advantages:**
  - **Adaptability:** Use of high-level and domain-specific languages in the development of systems.
  - **Productivity:** Automated code generation for a wide range of platforms based on the same model.
  - **Reliability:** Verification prior to implementation and deployment.

MDEV Research Group

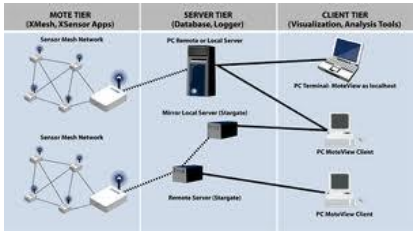
@ Bergen University College http://prosjekt.hib.no/ict/research/model-based-software-engineering/

Complex simulation systems (energy)

Process-aware software systems (health)



Protocols and networked embedded systems (IoT)



Cloud- and mobile applications

Applications

DPF
DIAGRAM PREDICATE FRAMEWORK

eclipse

CPN Tools

Software tools

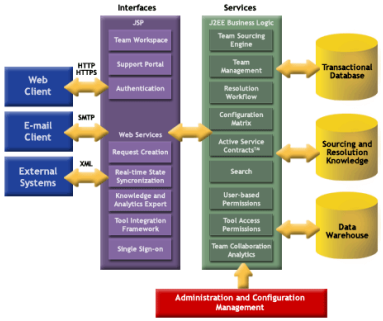
MDEV: Model-Driven software Engineering and Verification

Foundations

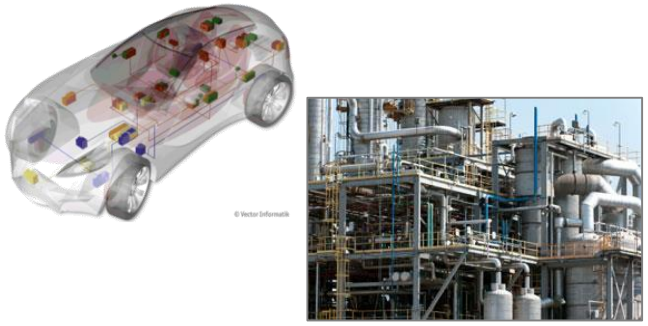
Concurrent Systems

- **The vast majority of ICT 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 and Web-based applications, protocols



Multi-core platforms and multi-threaded software



Embedded systems and networked control systems

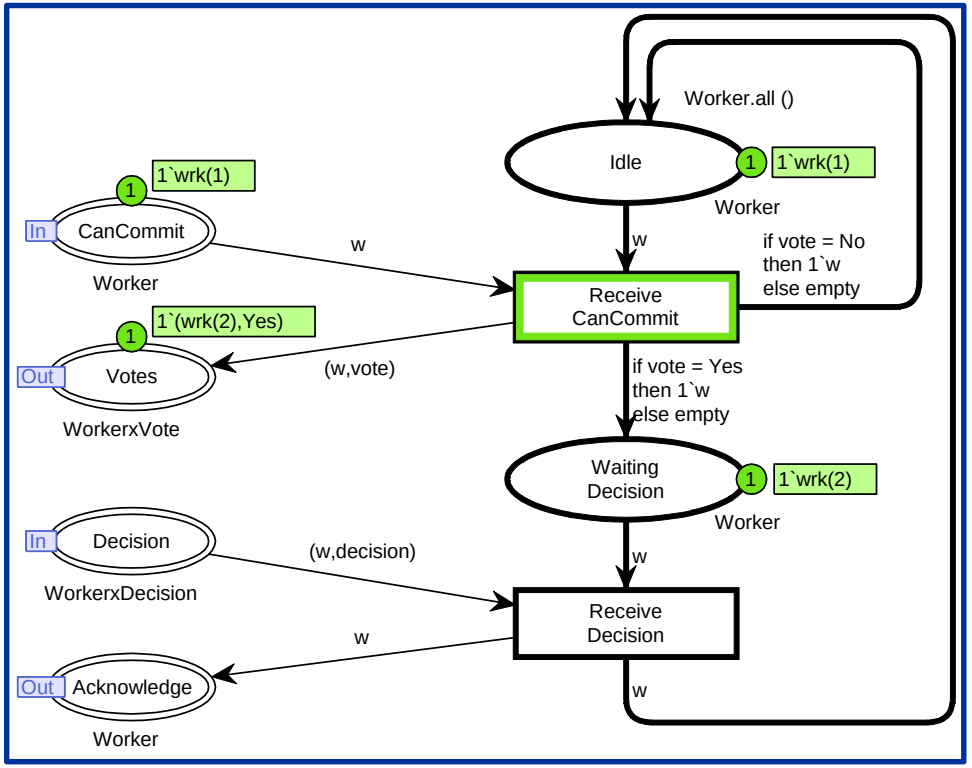
Concurrent Systems

- **The engineering of concurrent systems is challenging due to their complex behaviour:**
  - Concurrently executing and independently scheduled software 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.
  - Reproducing errors is often difficult.

- **Techniques to support the engineering of reliable concurrent systems are important.**

Coloured Petri Nets (CPNs)

- **Graphical modelling language for the engineering of concurrent systems.**
- **Combines Petri Nets and a programming language:**



Petri Nets: [C.A. Petri'62]

- graphical notation
- concurrency
- communication
- synchronisation
- resource sharing

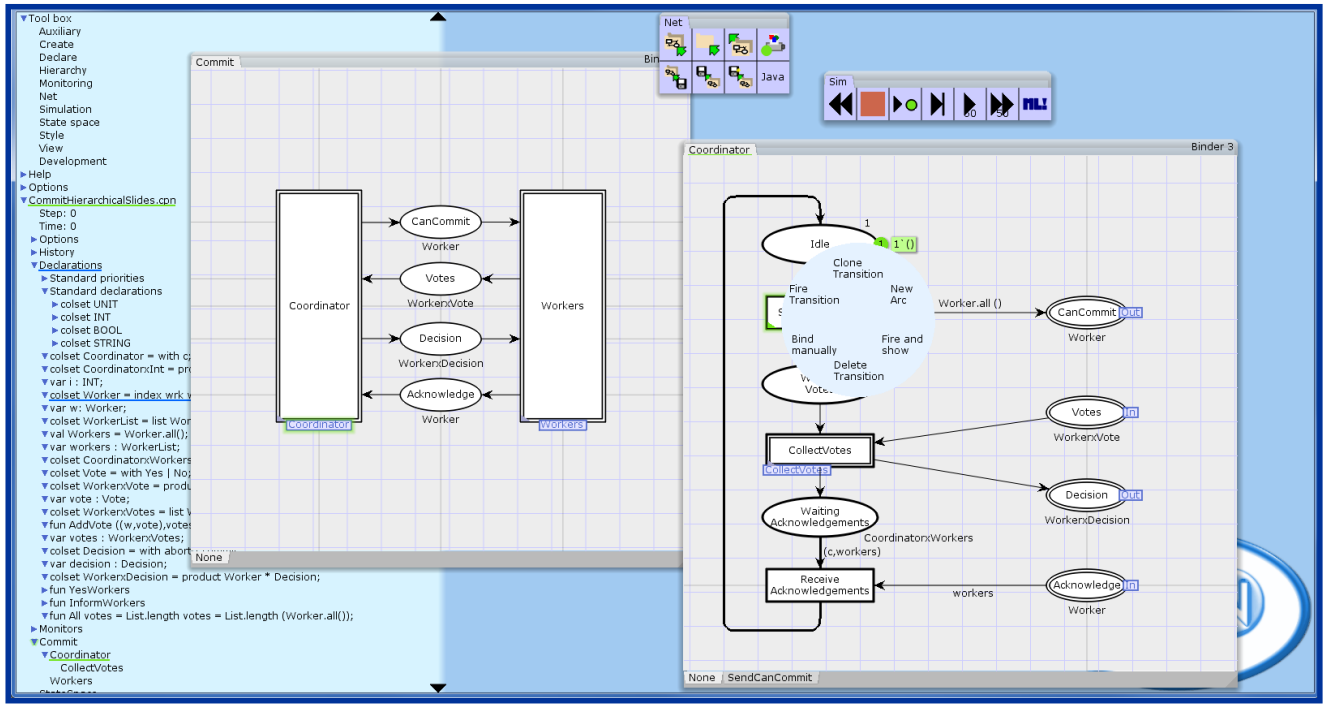
CPN ML (Standard ML):

- data manipulation
- compact modelling
- parameterisable models

High-Level Petri Net

CPN Tools [www.cpntools.org]

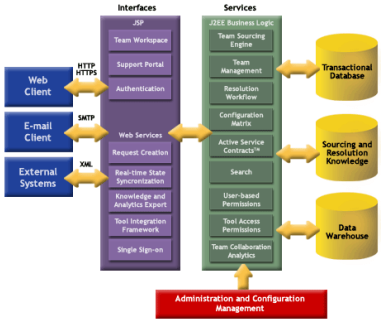
- **Practical use of CPNs is supported by CPN Tools:**

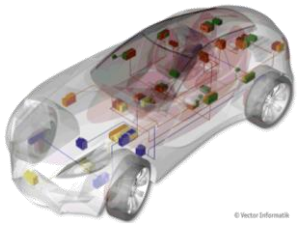


- **Editing and syntax check.**
- **Interactive- and automatic simulation.**
- **Application domain visualisation.**
- **Verification based on state space exploration.**
- **Simulation-based performance analysis.**

Application Areas








- **Communication protocols and data networks.**
- **Distributed algorithms and software systems.**
- **Embedded systems and control software.**
- **Business processes and workflow modelling.**
- **Manufacturing systems.**
- **… [http://cs.au.dk/cpnets/industrial-use/]**

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

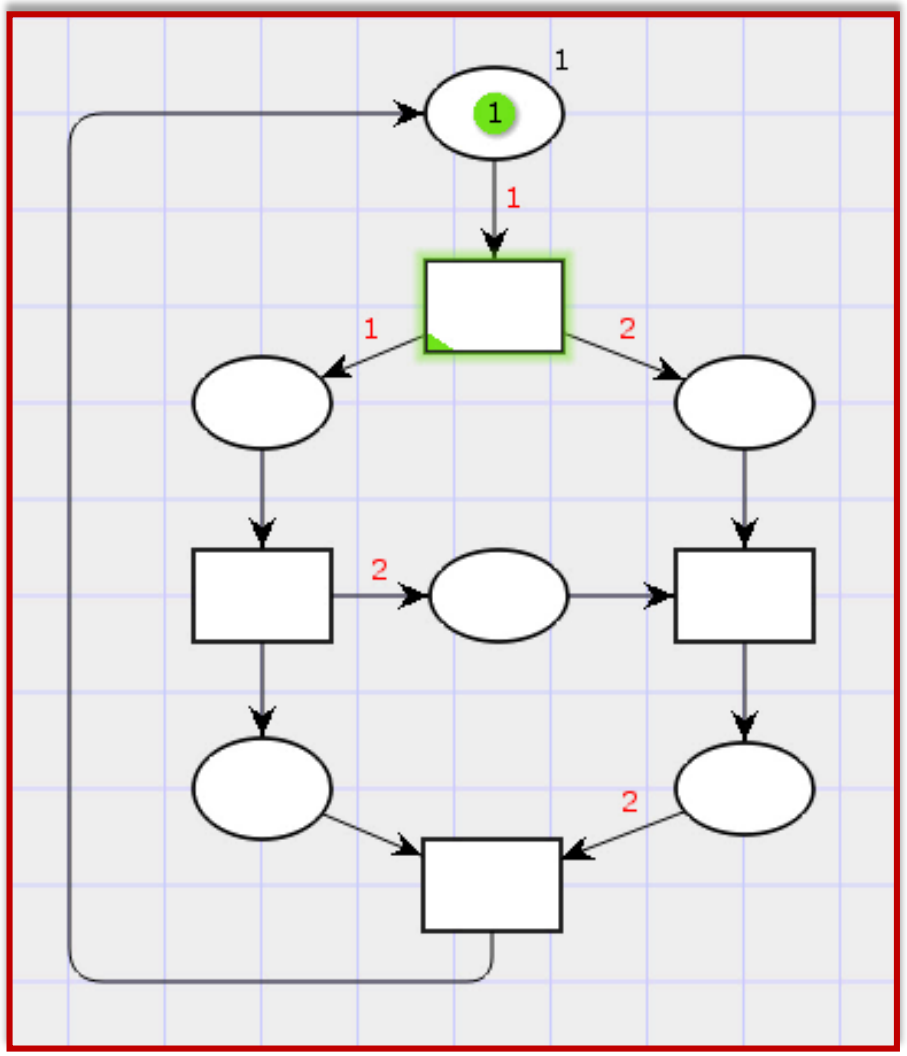
Europe

- 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

Asia

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

Quick Recap: Petri Net Concepts



State modelling:

- **Places (ellipses) that may hold tokens.**
- **Marking (state): distribution of tokens on the places.**
- **Initial marking: initial state.**

Event (action) modelling:

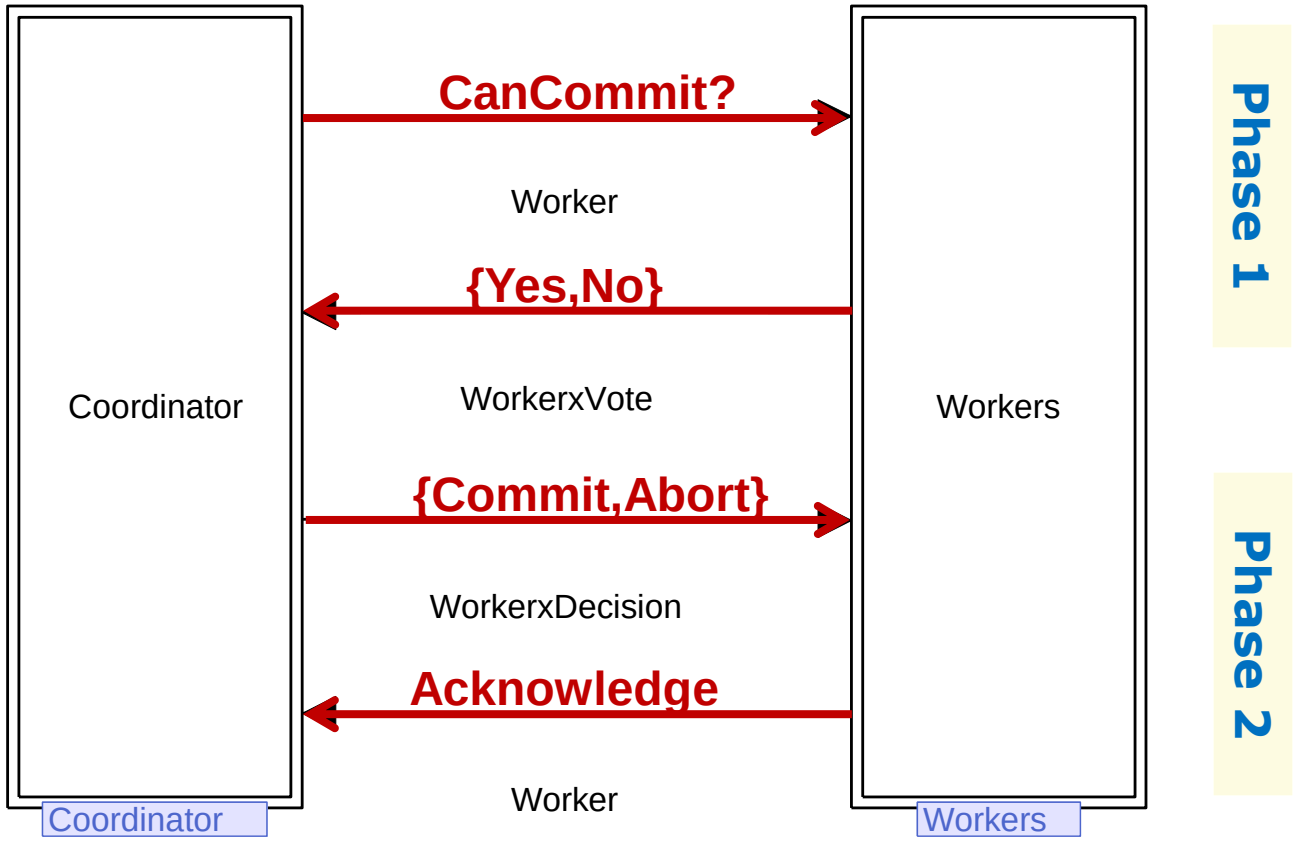
- **Transitions (rectangles)**
- **Directed arcs: connecting places and transitions.**
- **Arc weights: specifying tokens to be added/removed.**

Execution (token game):

- **Current marking**
- **Transition enabling**
- **Transition ocurrence**

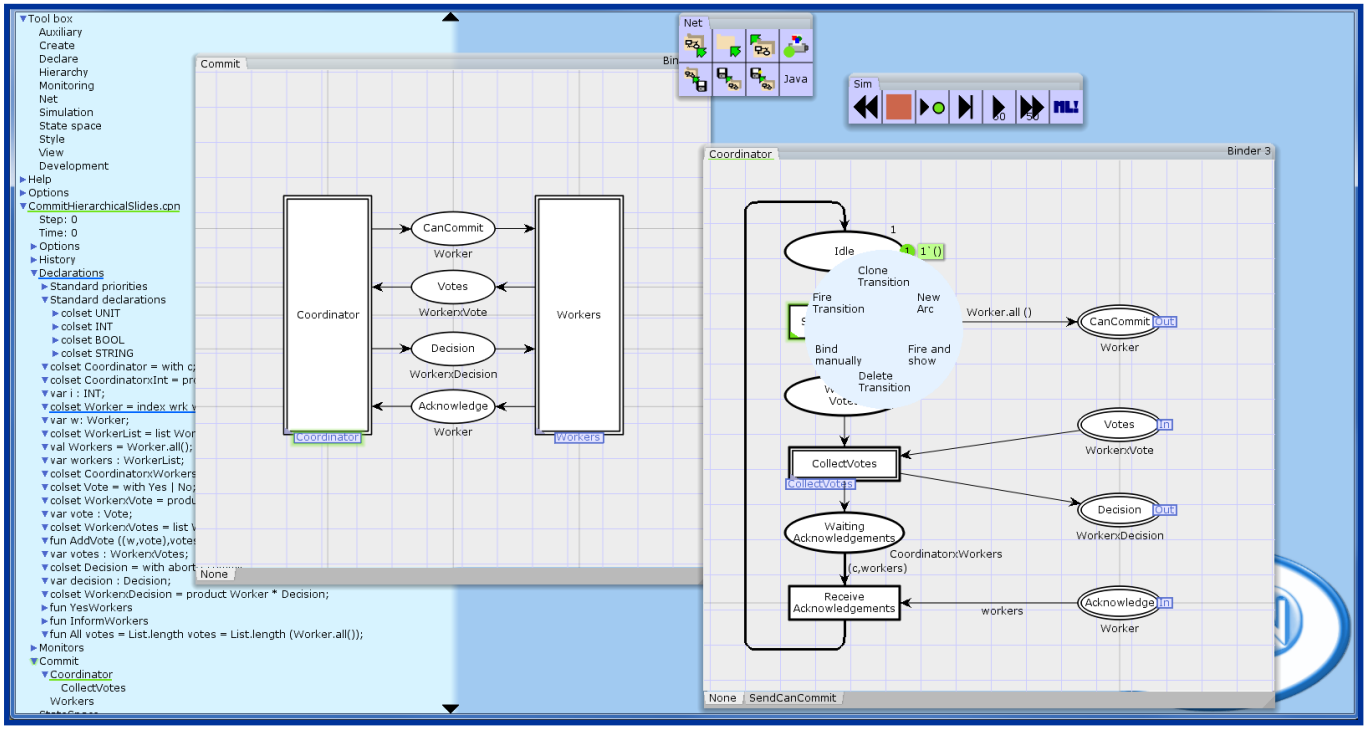
Example: Two-phase Commit Transaction Protocol

- **A concurrent system consisting of a coordinator process and a number of worker processes:**



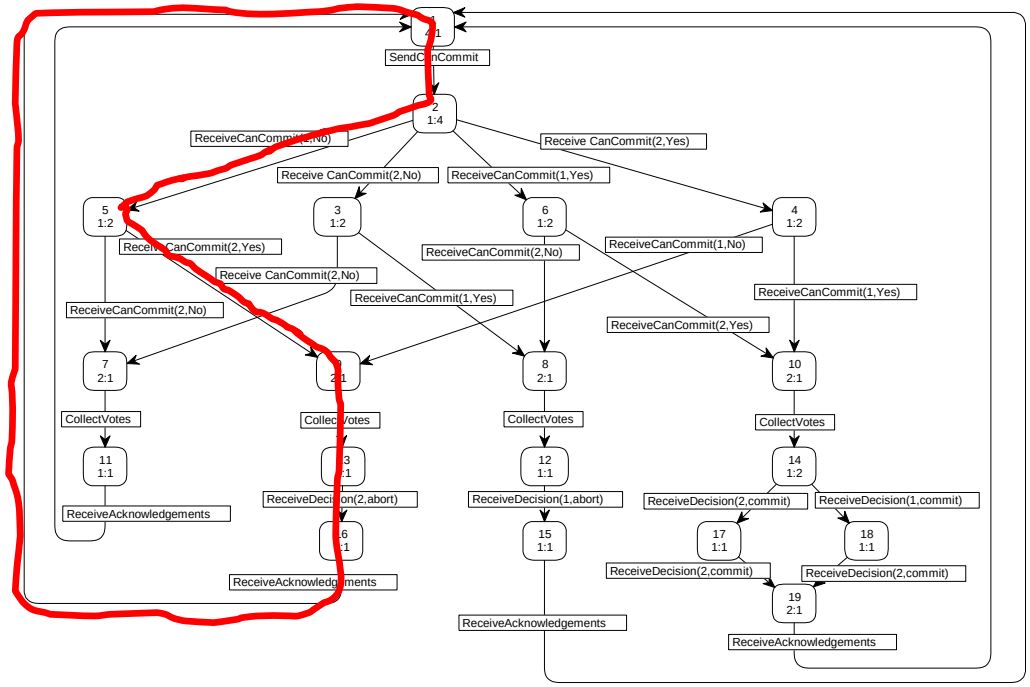
CPN Tools: Demo

- **Simulation**
- **Editing**



Verification and Model Checking

- **Formal verification of CPN models can be conducted using explicit state space exploration:**

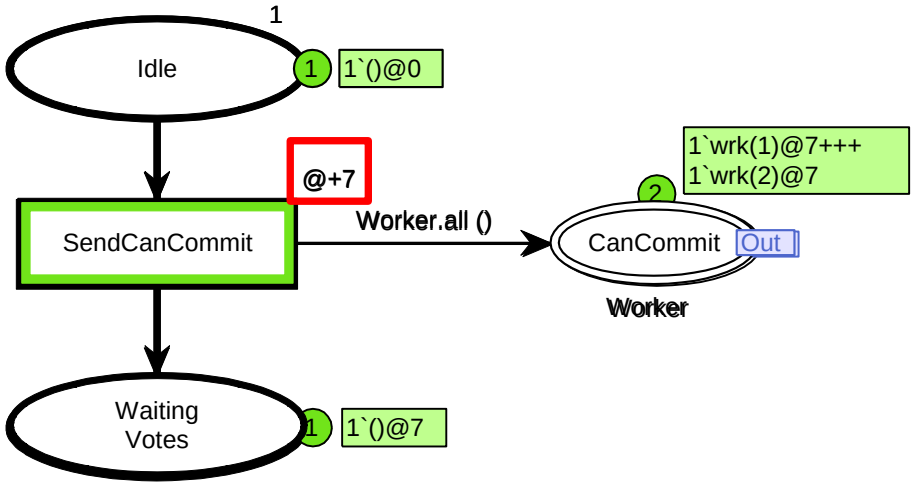


- **Represents all possible executions of the model.**
- **Standard behavioural properties can be investigated using the state space report.**
- **Model-specific properties can be verified using temporal logic model checking.**
- **Diagnostic information can be provided fully automatically.**

- **Several advanced techniques available to alleviate the inherent state explosion problem.**

Performance Analysis

- **CPNs include a concept of time that can be used to model the timed taken by activities:**



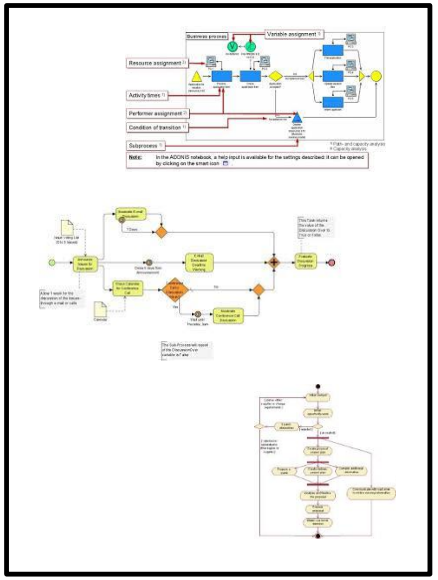
  - **A global clock representing the current model time.**
  - **Tokens carry time stamps describing the earliest possible model time at which they can be removed.**
  - **Time inscriptions on transitions and arcs are used to give time stamps to the tokens produced on output places.**

- **Random distribution functions can be used in arc expressions (delays, packet loss, ...).**
- **Data collection monitors and batch simulations can be used to compute performance metrics.**

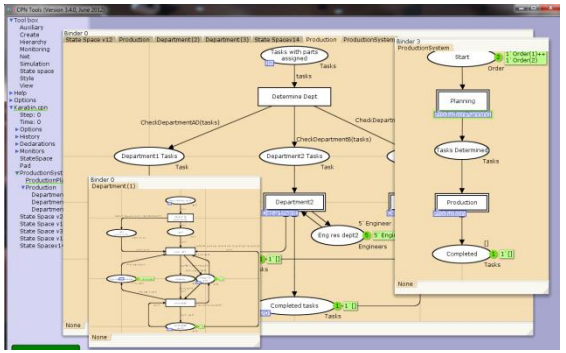
Example: Karabin

- **Process modelling, documentation and improvement (oil/gas/production/service industry)**
- **Project objective: automated analysis of process models for production planning:**

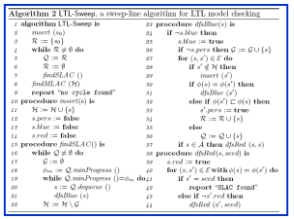
Process models

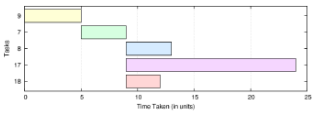


Modelling tools



Software for automated process verification and analysis

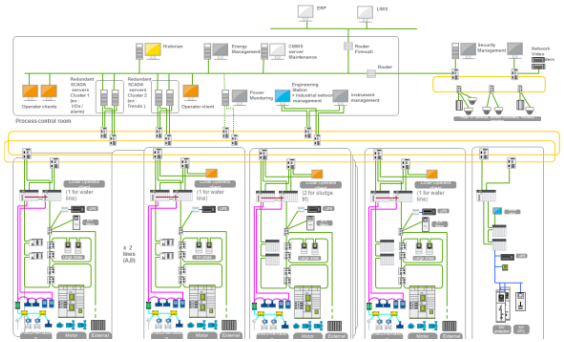




Process improvement

Example: Schneider Electric

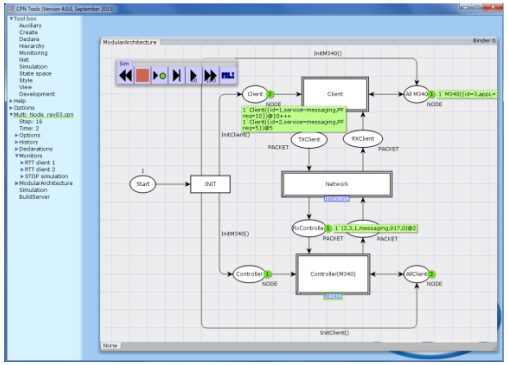
- **Develops complex automation systems for the energy domain: verification is essential.**



Software and analysis tools

Performance - Reliability
Availability - Safety

Modelling



Automated code generation

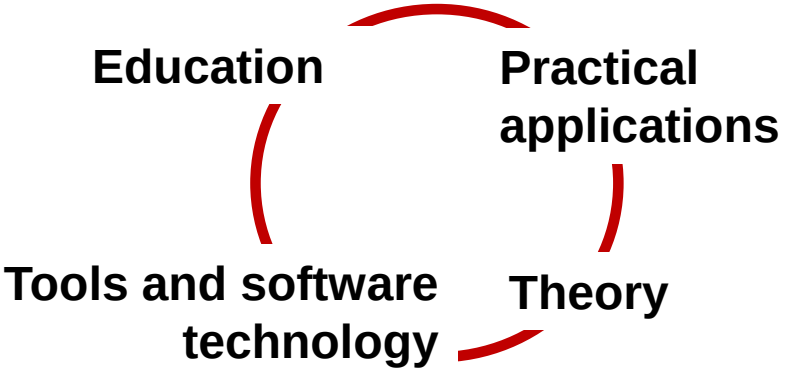
Research

Innovation

Tools for model-driven software engineering

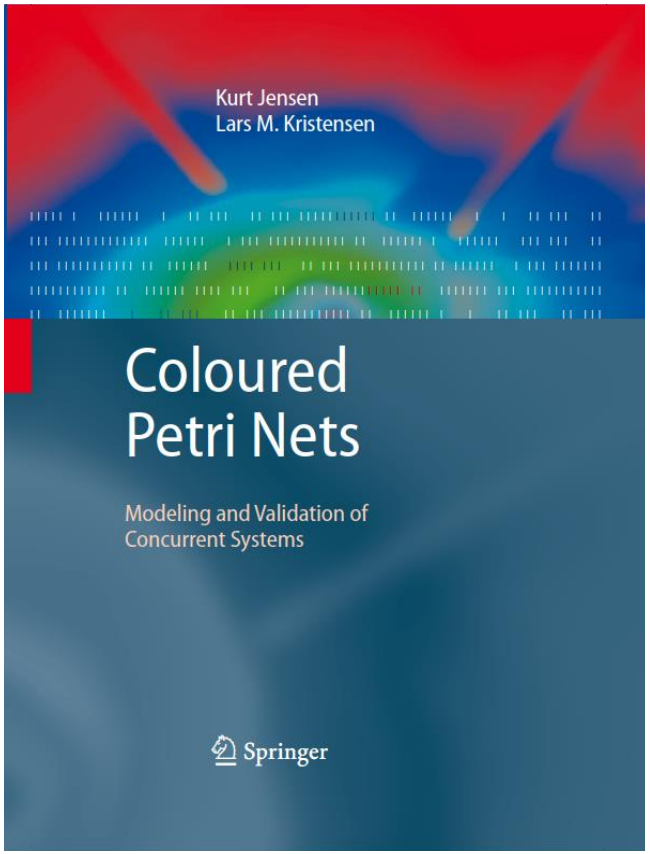
Perspectives on CPNs

- **Modelling language combining Petri Nets with a programming language.**
- **The development has been driven by an application-oriented research agenda**



- **Key characteristics:**
  - **Few but still powerful and expressive modelling constructs.**
  - **Implicit concurrency inherited from Petri nets: everything is concurrent unless explicit synchronised.**
  - **Verification and performance analysis supported by the same modelling language.**

CPN Literature



www.cpnbook.org

- **K. Jensen and L.M. Kristensen. Coloured Petri Nets: Modelling and Validation of Concurrent Systems, Springer, 2009.**
- **K. Jensen and L.M. Kristensen. Coloured Petri Nets: A Graphical Language for Modeling and Validation of Concurrent Systems. Vol. 58, No. 6 of Communications of the ACM, pp. 61-70, July 2015.**