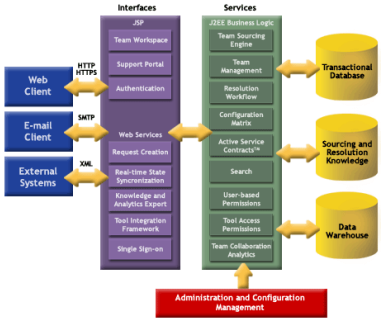
Coloured Petri Nets
Modelling and Validation of Concurrent Systems

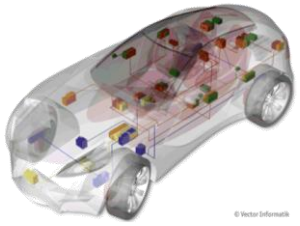
Chapter 1: Modelling and Validation

Lars M. Kristensen
Department of Computing
Bergen University College, NORWAY
Email: lmkr@hib.no / WWW: www.hib.no/ansatte/lmkr

Concurrent Systems

- **The vast majority of IT 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

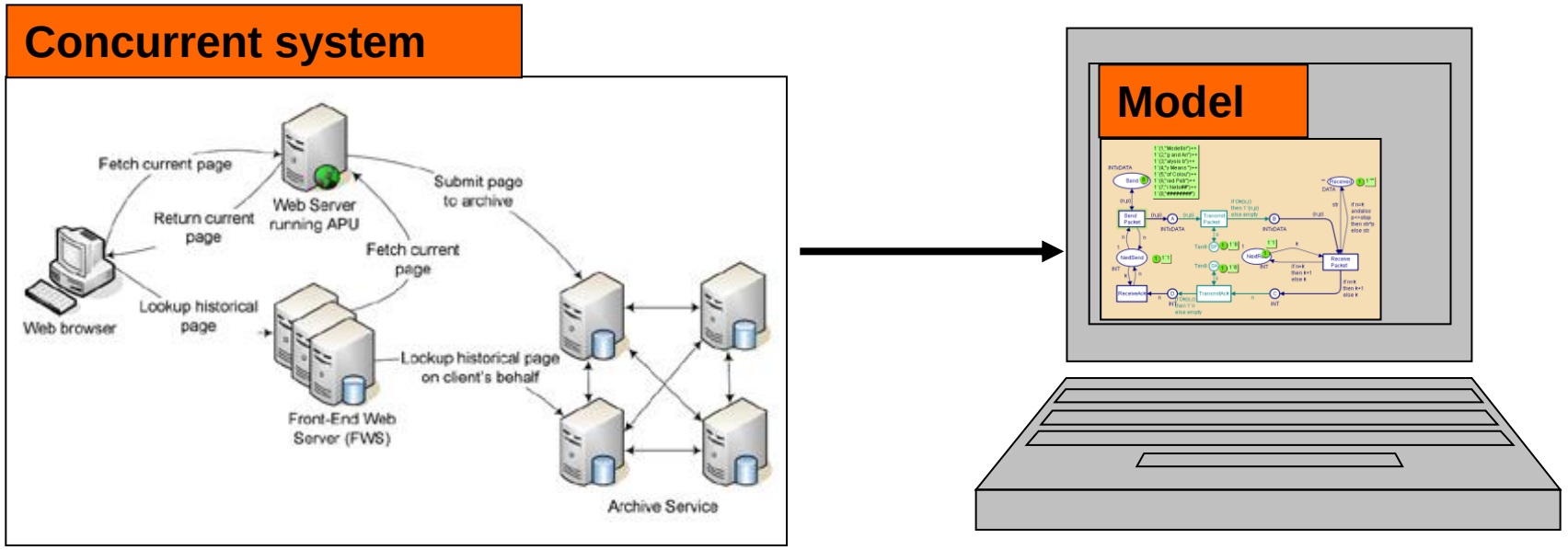
Concurrent Systems

- **Most software development projects are concerned with concurrent software 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.**

Modelling

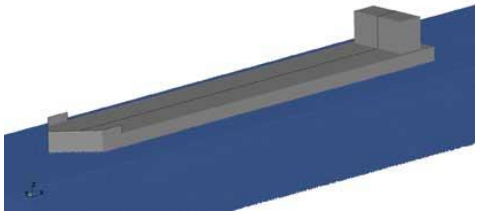
- One way to approach the challenges posed by concurrent systems is to build a model.
- A model is an abstract representation which can be manipulated by a computer software tool:



- Using a model it becomes possible to investigate properties of the system prior to implementation.

Modelling ...

- **Modelling is also used in many other disciplines:**
  - **When engineers construct a bridge.**
  - **When architects design a building.**



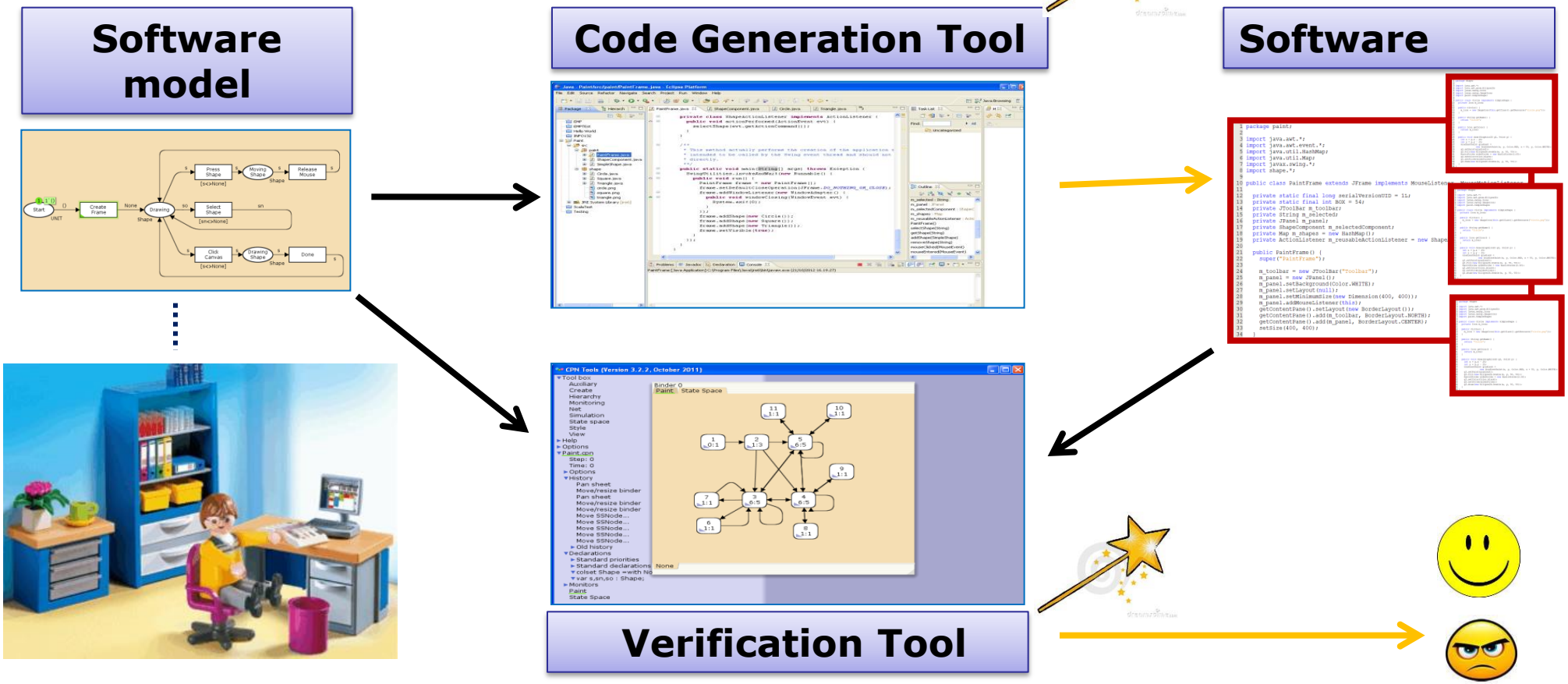
- **Modelling is typically done in the early phases of system development.**

Why Software Modelling?

- **Supports abstraction and the use of domain-specific concepts in software development.**
- **Benefits of constructing a model:**
  - 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.
- **Properties can be validated prior to implementation and deployment:**
  - Functional properties (e.g., deadlocks, timing requirements,…).
  - Performance properties (e.g., delay, throughout, scalability,…).
- **The software models can be used as a basis for automated generation of implementations.**

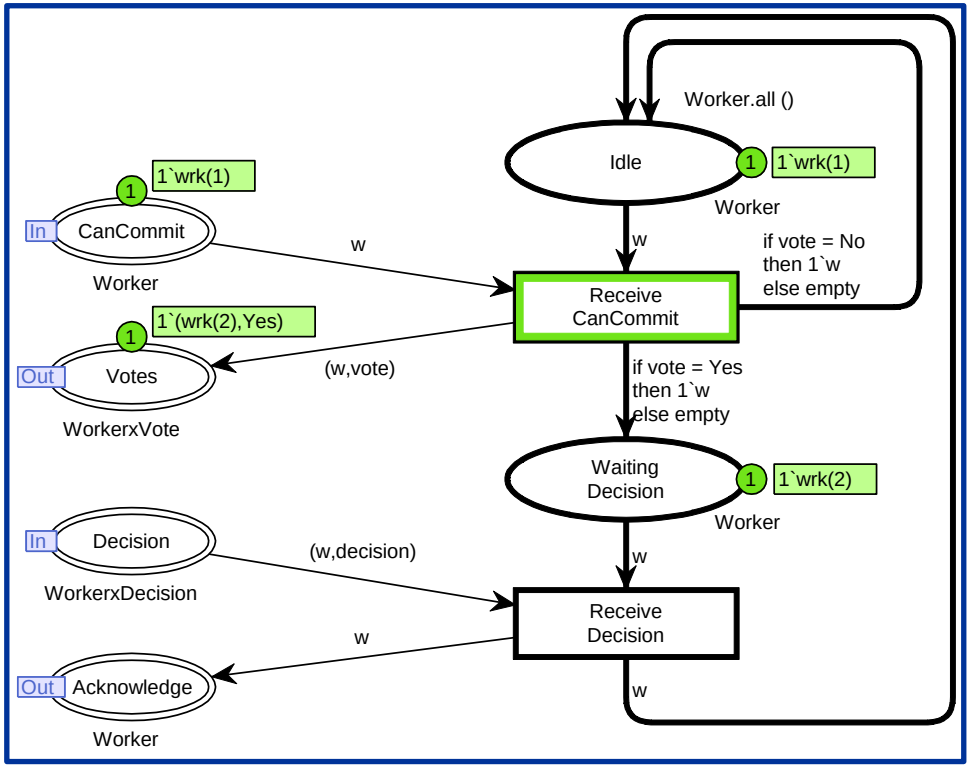
Model-driven Software Engineering

- **Enables automatic code generation of software and automated verification:**



Coloured Petri Nets (CPNs)

- **General-purpose 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

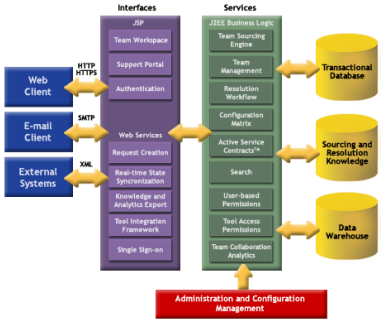
CPN ML (Standard ML):

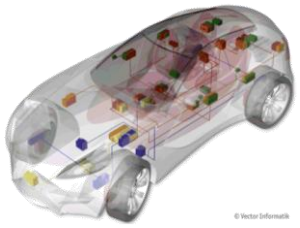
- data manipulation
- compact modelling
- parameterisable models

High-Level Petri Net

Application Areas

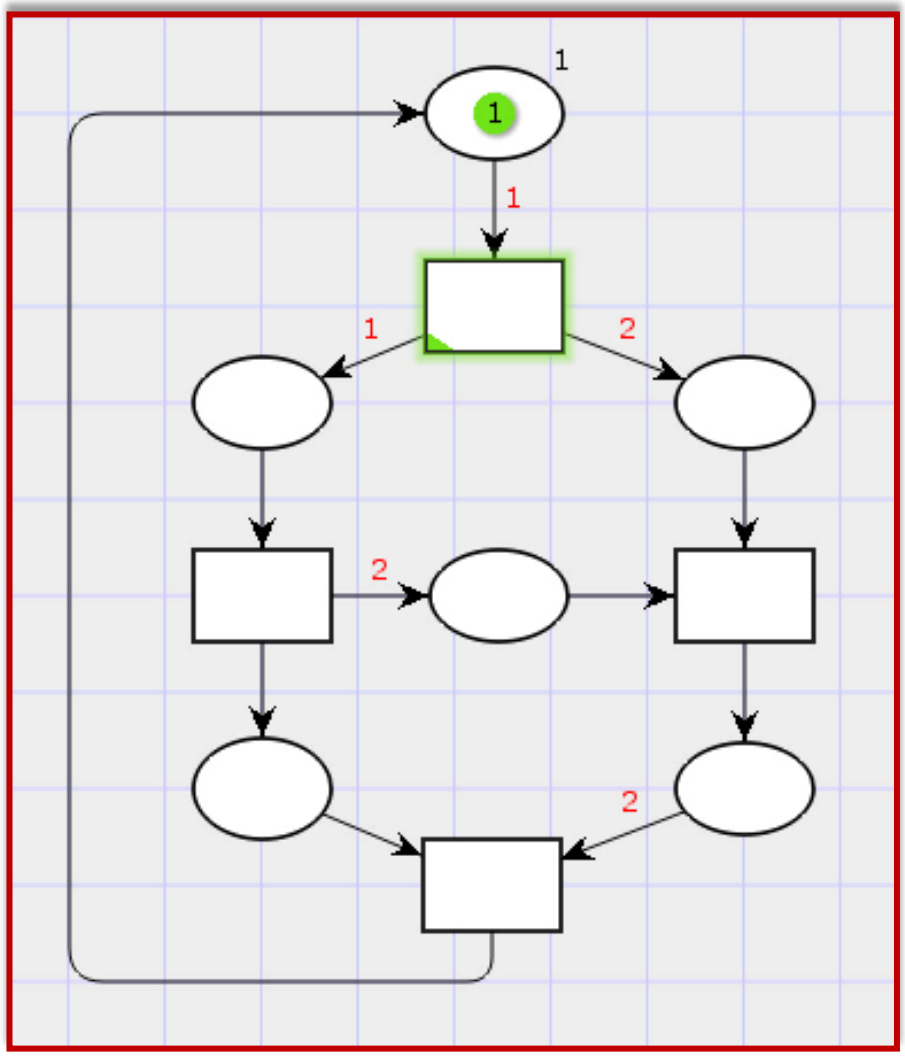







- **Communication protocols and data networks.**
- **Distributed algorithms and software systems.**
- **Embedded systems and control software.**
- **Business processes and workflow modelling.**
- **Manufacturing systems.**
- **… [http://cs.au.dk/cpnets/industrial-use/]**

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):

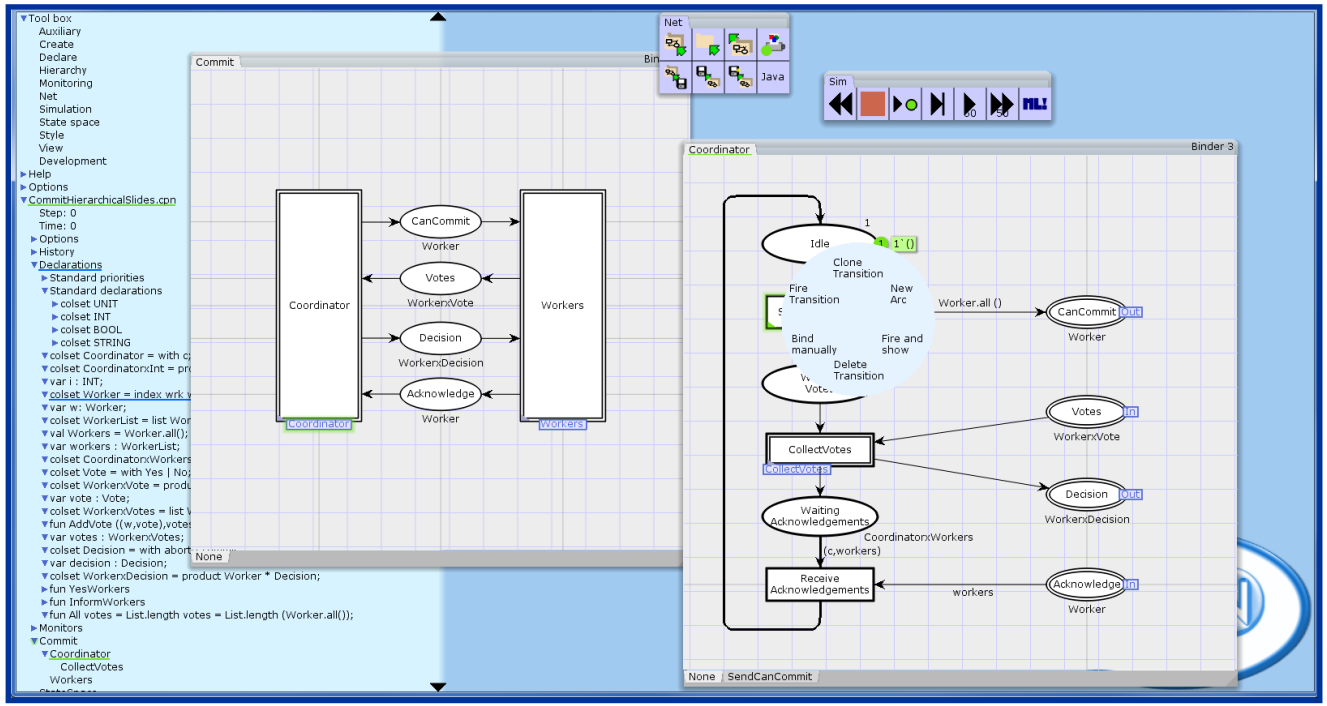
- **Current marking**
- **Transition enabling**
- **Transition ocurrence**

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 and the CPN modelling language and supporting tools conform to this standard.**

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.**

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

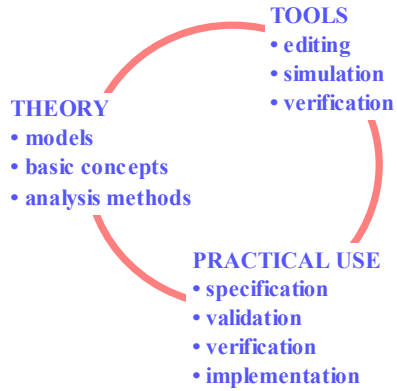
- Mitsubishi Electric Corp., Japan
- Toshiba Corp., Japan
- SHARP Corp., Japan
- Nippon Steel Corp., Japan
- Hongkong Telecom Interactive Multimedia System

CPN models are formal

- **The CPN modelling language has a mathematical definition of both its syntax and semantics.**
- **The formal representation is important as it:**
  - Provides the foundation for the definition of the different behavioural properties and the analysis methods.
  - Would have been impossible to develop a sound and powerful CPN language without it.
- **Formal models can be used to verify system properties such as:**
  - Proving that certain desired properties are fulfilled
  - Proving that certain undesired properties are guaranteed to be avoided.

Role of CP-nets

- **The development of CP-nets has been driven by the desire to develop:**
  - **an industrial strength modelling language, which is**
  - **theoretically well-founded and**
  - **versatile enough to be used in practice for systems of the size and complexity found in typical industrial projects.**



- **CP-nets is not a modelling language designed to replace other modelling languages (such as UML).**
- **CP-nets is often used in conjunction with other to existing modelling languages and methodologies.**

Questions