Industrial verification of system behaviour using the mCRL2 toolset

Aad Mathijssen

Design and Analysis of Systems group
Laboratory for Quality Software (LaQuSo)

Department of Mathematics and Computer Science
Technische Universiteit Eindhoven

Dutch Dependability Day
University of Twente

12th May 2009

Analysis techniques

Development of *distributed systems* is inherently complex:

- Needed: assessment and improvement of quality
- Means: analysis techniques

Analysis techniques used in distributed system development:

- Structure: what *things* are in the system?
- Behaviour: what *happens* in the system?

The two techniques complement each other because they focus on *different aspects* of the system.

Analysis of system behaviour

What is analysis of system behaviour about?

- Modelling: create an *abstract* model of the *behaviour* of the system
  - *gain insight* in the behaviour
  - *reduce complexity* to allow for validation and verification
- Validation: are we building the right product?
  - *test requirements* on the model for a number of paths and configurations
- Verification: are we building the product right?
  - *verify requirements* on the model for all possible paths and configurations

Analysis of system behaviour in industry

Ingredients for successful application in industry:

- Modelling language:
  - capable of expressing *real-world systems*
- Tool support:
  - *automation*
  - *scalability*
- Provide service:
  - share *expertise* and offer *training*
  - actually *perform* the analysis

Analysis of system behaviour in industry

Solution provided concerning the mCRL2 toolset:

- Modelling language:
  - *generic* language
  - combines *process algebra* with *functional programming*
- Tool support:
  - Supports *automated* validation and verification
  - Flexible tool *chain*
- Service:
  - *Laboratory for Quality Software* (LaQuSo)

mCRL2 toolset: overview

Overview of the mCRL2 toolset:

- 20 years of history:
  - Late 1980s: Common Representation Language (CRL)
  - From 1990: μCRL
  - During 1990s: μCRL toolset
  - From 2004: mCRL2 and mCRL2 toolset
- Collection of tools
- External languages and tools are supported:
  μCRL, CADP, χ, PNML, TorX, LySa, SystemC, LTSmin
- Multi-platform: Windows, Mac and UNIX variants
- Free software licence: Boost licence
- Release policy: fixed release cycle (January and July)

mCRL2 toolset: modelling

Ingredients for modelling:

- *Actions* (push_button, place_order, call_f)
- *Non-deterministic choice*
  (*either* push_button *or* place_order)
- *Sequence* (*first* push_button, *then* place_order)
- *Processes* (Client, WebShop)
- *Parallelism* (Client *in parallel with* WebShop)
- *Synchronous communication*
  (push_button *communicates with* place_order)
- *Data types*
  (push_button(on), Client(1), call_f$(\{x|prime(x)\})$)

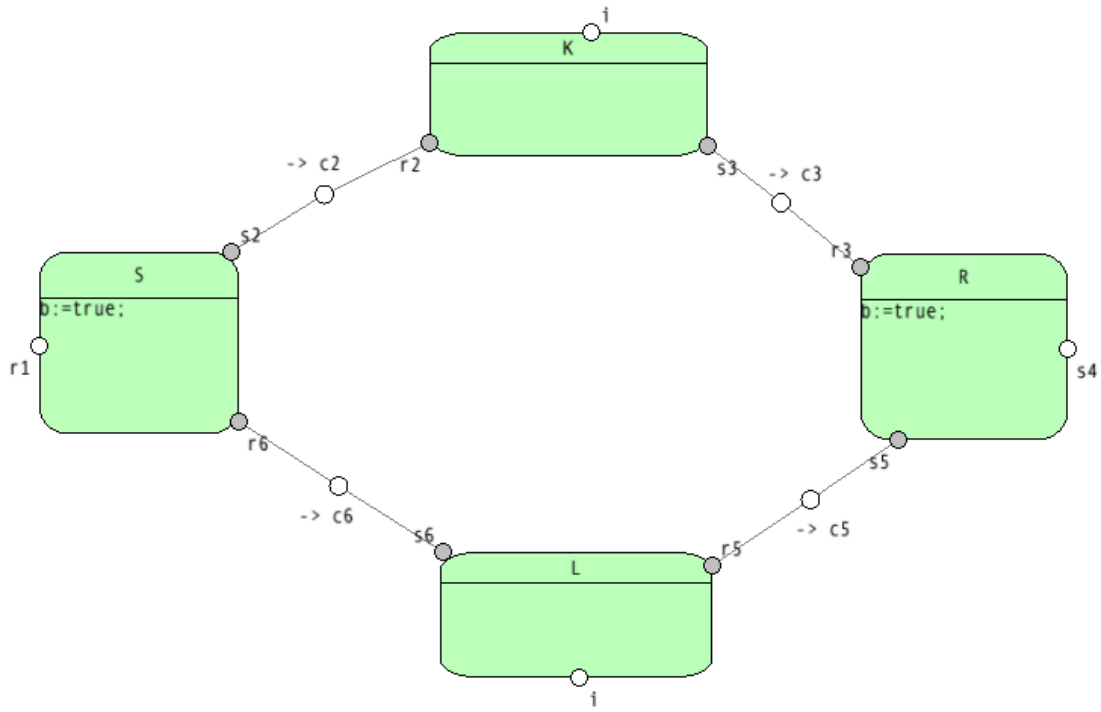
mCRL2 toolset: modelling

The toolset supports two kinds of modelling:

- *Textual*:

  init $\nabla_{\{\mathsf{r1,s4,c2,c3,c5,c6,i}\}}(\Gamma_{\{\mathsf{r2|s2\rightarrow c2,r3|s3\rightarrow c3,r5|s5\rightarrow c5,r6|s6\rightarrow c6}\}}($
  $\mathsf{S}(true) \parallel \mathsf{K} \parallel \mathsf{L} \parallel \mathsf{R}(true)$
  $));$

- *Graphical*:



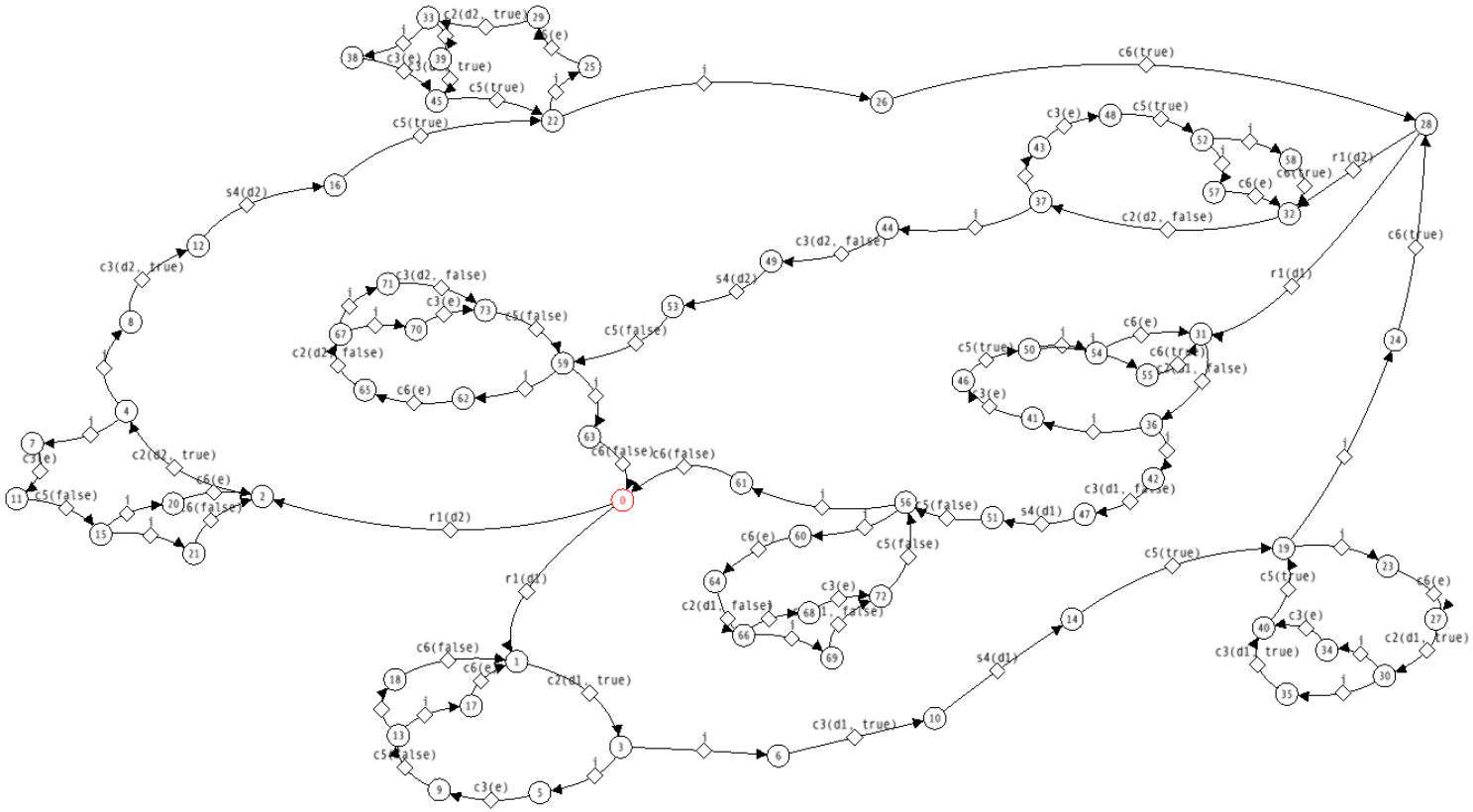
mCRL2 toolset: validation

Validation of models supported by the toolset:

- Manual or semi-automated simulation
- Automated testing using the TorX test tool
- Different types of visualisation

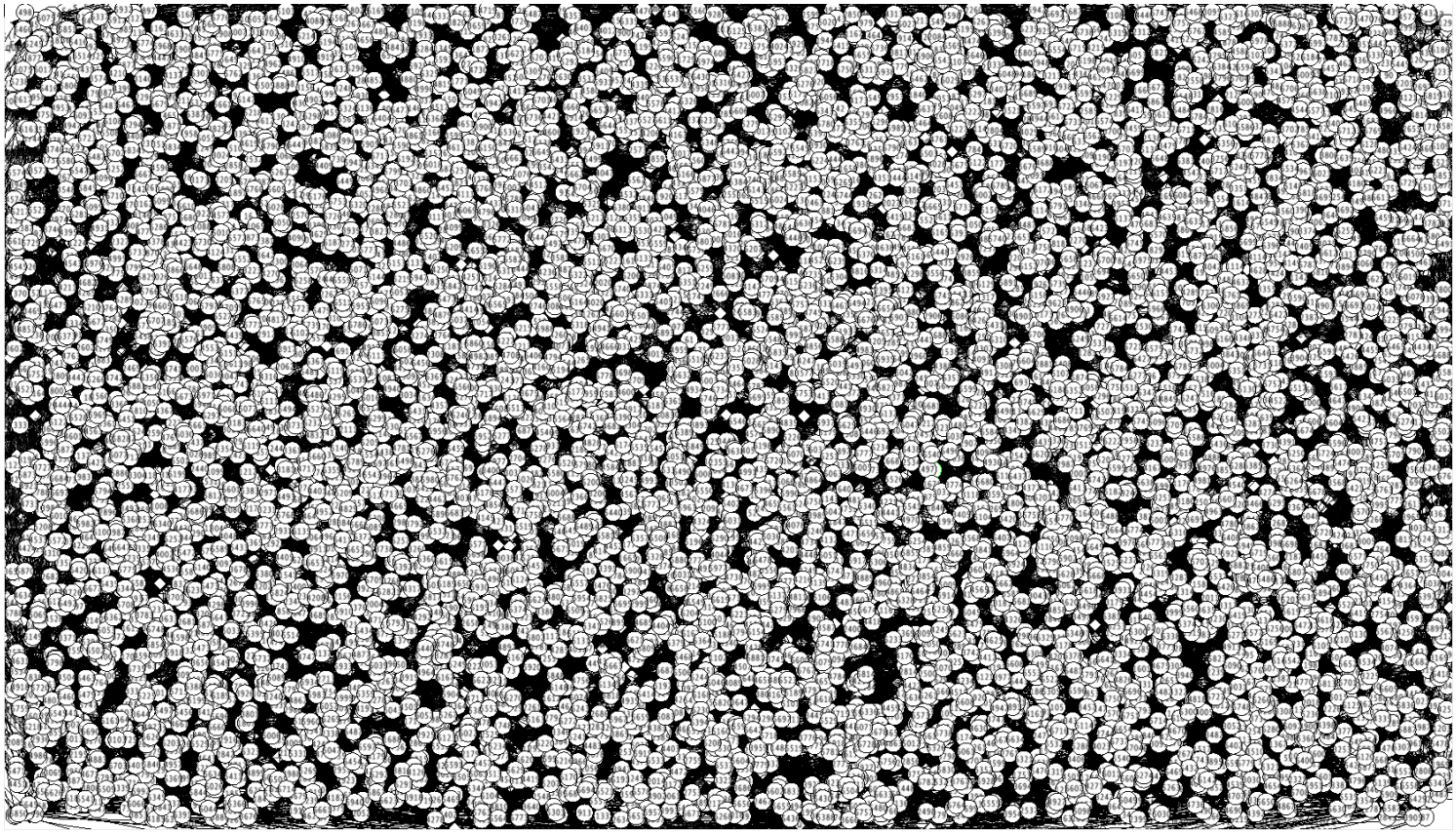
mCRL2 toolset: visualisation

Visualisation as a directed graph using *automatic positioning*:



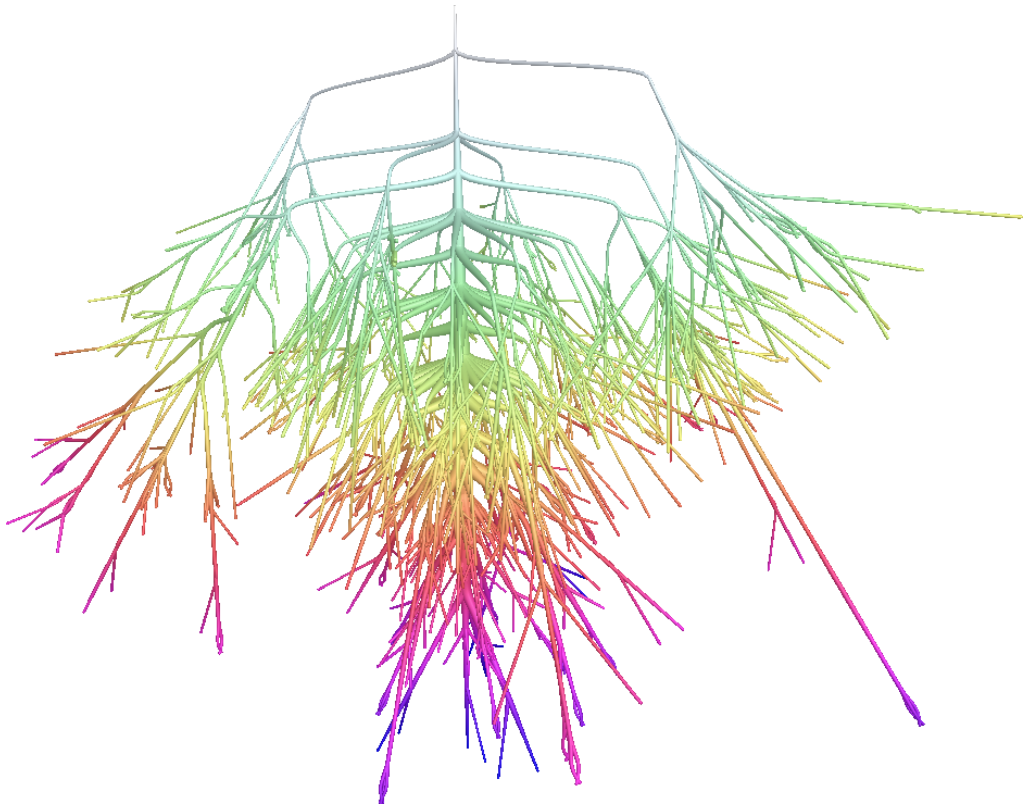
mCRL2 toolset: visualisation

Visualisation as a directed graph is limited to *small models*:



mCRL2 toolset: visualisation

Visualisation as a 3D tree of clusters of states:



mCRL2 toolset: verification

Toolset support for automated verification
of requirements on the complete model:

- Occurrences of *deadlocks*
- Occurrences of specific *actions*
- *Equivalence* of models
- *Formula checking*:
  - express requirements as *temporal logic formulas*
  - check these formulas on the model

Industrial case studies

Selection of industrial case studies with the mCRL2 toolset:

- Error prevention:
  - Control software of a *humanoid robot*
  - Automated *parking garage*
- Error detection:
  - *Load balancer* for document processors
  - I^2C Linux *driver*

Industrial case studies

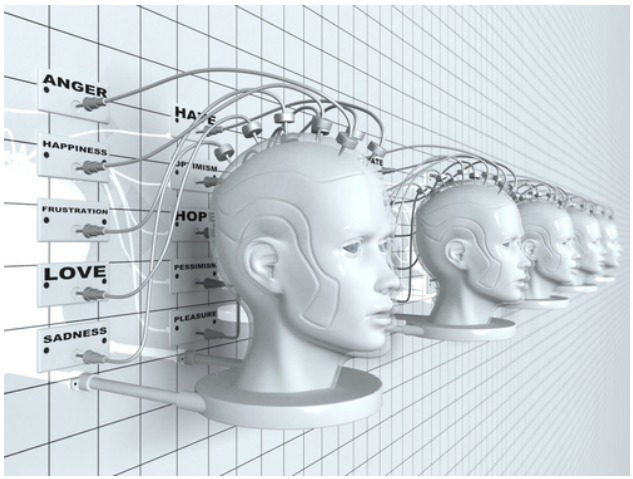
Control software of a humanoid robot

Humanoid robot:

- Standalone and external control
- Architectural design of the control software in *UML*:
  - structure as *component diagrams*
  - behaviour as *state charts*

Analysis:

- UML $\rightarrow$ mCRL2
- Verification:
  - Check for deadlocks
  - Corrected UML models
- Size of the correct model:
  - 6.792 states
  - 29.242 transitions



Industrial case studies

Automated parking garage

An automated parking garage:




Industrial case studies

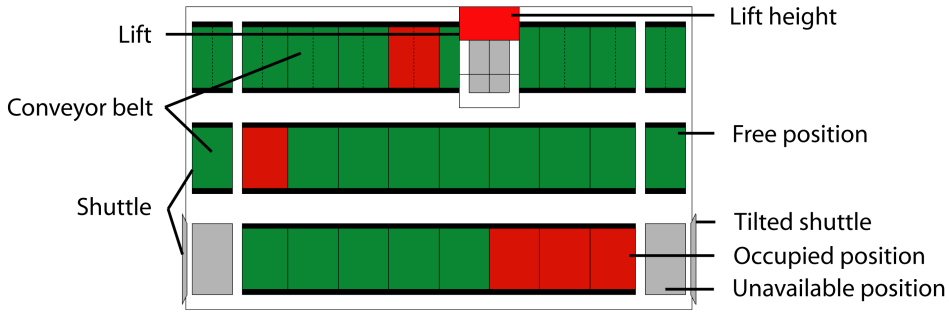
Automated parking garage (2)

Verified design of an automated parking garage:

- Design of the control software
- Verification of the safety layer of this design

Analysis:

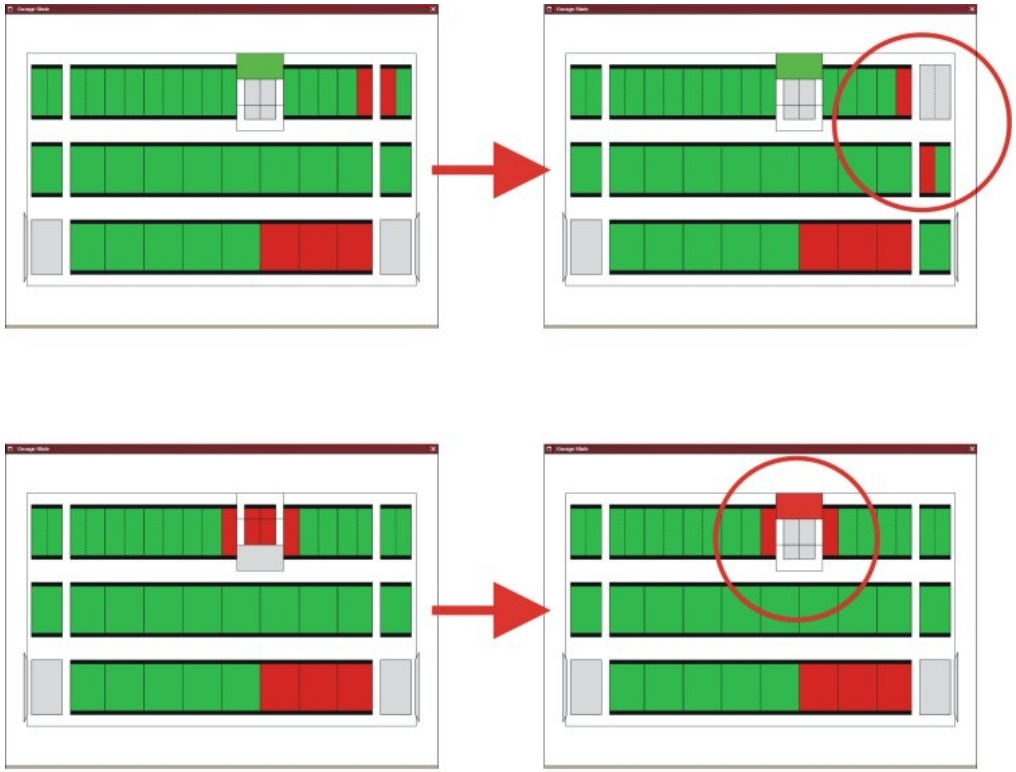
- Model: 991 lines of mCRL2
- Verification: augmented with *error* actions (217 lines)
- Size: 3,3 million states and 98 million transitions
- Simulation using custom made visualisation plugin



Industrial case studies

Automated parking garage (3)

Design errors found using visualisation plugin:



Industrial case studies

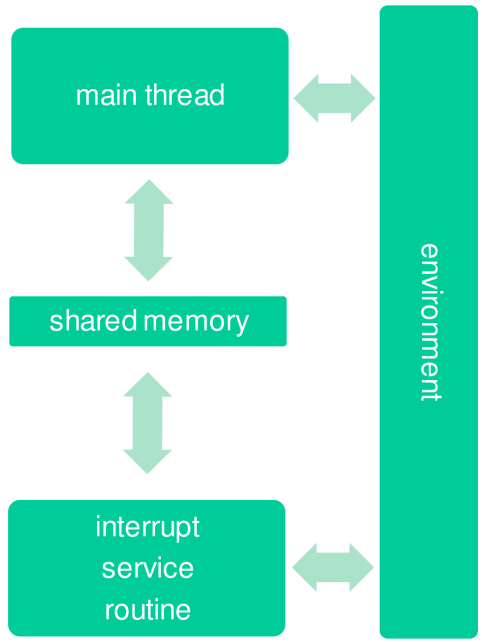
I^2C Linux driver

I^2C Linux driver:

- I^2C: bi-directional 2-wire bus for inter-IC communication
- Linux driver for specific I^2C device

Analysis:

- Focus: shared memory access
- C source code $\rightarrow$ mCRL2
- Verified mutual exclusion
- Size:
  - 62 million states
  - 102 million transitions
- Actual errors found: 2
- Added multi-threading support



Industrial case studies

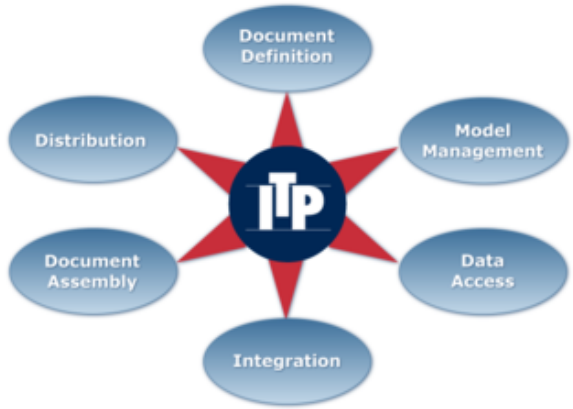
Load balancer for document processors

Intelligent Text Processing (ITP):

- Distribution of print jobs over document processors
- 7.500 lines of C code

Analysis:

- Focus: load balancing
- C source code $\longrightarrow$ mCRL2
- Verification: formula checking
- Size:
  - 1,9 billion states
  - 38,9 billion transitions
- Actual errors found: 6
- LaQuSo certification



Thank you for your attention

More information can be found on `mcrl2.org`.