Specification and Testing of Banknote Processing Systems with Coloured Petri Nets

Thomas Schnattinger, Andrej Pietschker

Munich, 06/17/2010
30th TAV

Agenda

- Automated testing in the software development process
- Motivation for system specifications with formal methods
- Excursion: Basics of Coloured Petri Nets (CPNs)
- A Coloured Petri Net model for BN processing and jam recovery
- A Coloured Petri Net model as System Under Test (SUT)
- Conclusions
Automated SW-Testing – Where do we Stand?

- Stakeholder Requirements
- System Specifications
- Architecture
- Design
- Implementation
- Product Acceptance Test
- Stable system releases for system testing
- Pre-tested unit integrated and tested at component level using TTCN-3 test automation
- Automated Unit Tests integrated into the build

Covered by automated tests
Automated SW-Testing – Potential Extensions

Test-cases should be applied as early as possible

Test-cases should be available and verified as early as possible

*We need an executable model as a System Under Test (SUT) to apply and verify the test cases already in the first phases of the development process*

System Specifications – Our Current Approach

How do we currently tackle these issues in the specifications?

“Activity diagrams”:

*We need formal, executable high-level descriptions for the specifications, e.g. Petri Nets*
**Excursion: Coloured Petri Nets (CPNs)**

Coloured Petri Net = Petri Net + elaborated type system + arbitrary data manipulation

**Some benefits of Coloured Petri Nets**

- graphical oriented language with a well-defined semantics for design, specification, simulation and verification of systems
- hierarchical descriptions (subnets)
- description of states and actions (data manipulation)
- formal analysis methods
- tool support → **CPN Tools**

---

**Banknote Processing with Coloured Petri Nets**
Banknote Processing with Coloured Petri Nets – Details

Automated Model Testing – Coupling with TTCN-3
The new approach allows for testing specifications and system releases with the same test-cases.
Outlook

**Model extensions**
- Inclusion of timing/performance aspects in the CPN model
- Refinement to the level of single CAN messages

**Tool extensions**
- CPN model support for SW-architecture decisions
- Test case generation from the model
- Code generation from the CPN model
## Conclusions

**CPN require more effort during specifications, but**

- CPN Tools facilitates *explicit, unambiguous* and *complete* descriptions and thus *better specifications*
- The *visual representation* allows an abstract view, focused on the core elements
- The specifications can be used in automatic and interactive simulations
- Coupling with TTCN-3 allows to use the specifications as a *System Under Test* (SUT) in automated testing
- Test cases can be validated as soon as the specifications are available