Real-time systems modelling with UML state machines and coloured Petri nets
Journées nationales du GdR Informatique Mathématiques 2015
Mohamed Mahdi BENMOUSSA
LIPN, University Paris 13

Input: description of real-time system in natural language
- Real-time systems: complex and error-prone (high cost)
- Goal: early bug detection
- Case study: example of a toll gate

Specification method
- Structured method (extends [CR09])
- Goal: Avoid common errors
- Precise analysis and modelling of the system

Step 1
- Construction of class diagram
- Scenario of the system execution using Sequence diagram
- Definition of state observer and events

Step 2
- State observers: used to describe the state of the system
- Events: System change from one state to another by reacting to the event
- Update the state observer/event table in the next steps of the method

Step 3
- Invariants: Boolean expression to define a condition on the state observer
- Condition: the situation where the system react to the event
- Reaction: the changing applied to the system after reacting to the event

Step 4
- Table of states: possible and impossible states
- The use of state observers and invariants
- Used to construct the UML state machine model (with events and their condition/reaction)

Translation
- Work presented in [ABC14b]
- Described using a high-level algorithm
- Takes into account: hierarchy, transitions, concurrency, fork, join, inter-level transitions, composite states, variables, history pseudo-states, behaviours
- New implementation of an ad-hoc tool (previous implementation in [ABC14a])

Perspectives
- Extension of the syntactic elements set
- Integration of time in the specification method
- Integration of time in the translation of UML state machines
- Implementation of the translation using an automated support
- Proof of the equivalence between the UML state machine model and coloured Petri nets model using an external semantics (e.g., [LLACSWD13])

References


