Incremental Design and Formal Verification with UML/RT in the FUJABA Real-Time Tool Suite

Sven Burmester, Holger Giese, Martin Hirsch, and Daniela Schilling
Software Engineering Group
University of Paderborn
Warburger Str. 100
D-33098 Paderborn, Germany
{burmi|hg|mahirsch|das}@upb.de

Mechatronic systems of the future will be …

- networked,
- hard real-time,
- safety-critical,
- embedded, and
- will contain complex software.

How to ensure correctness?

A shuttle system that builds convoys to optimize the energy consumption: safety-critical maneuvers

I. UML/RT Subset

Pattern (Distance Coordination):
- Model: Statecharts for roles and connector
- Specification: required OCL RT properties
- Components (Shuttles):
- Model: Statecharts for ports (refined roles) and synchronization
- Specification: local OCL constraints

Elements:
- Components
- Ports
- Connectors
- Patterns
- Roles

Verify UML models with Real-Time

Model checking: limited today due to two main obstacles:
1. state explosion ⇒ restricted model size
2. batch style model checking vs. iterative design process

⇒ incremental design and verification with UML/RT (FUJABA Real-Time Tool Suite)

Underlying techniques (Outline):
I. Restrict approach to a UML/RT subset
II. compositional reasoning [ESEC03]
III. mapping of the UML/RT subset to HUppaal
IV. incremental checking of modified submodels
Real-Time Statecharts

RT @ PIM:
- Add Clocks (s. TA)
  - resets
  - guards
  - invariants
- Deadline intervals for each transition

Automatic code synthesis later ensures that the specified real-time properties are guaranteed.

Component Design & Verification

Behavior:
- Port statecharts which refine the related role protocols
- A synchronization statechart which coordinates all realized roles
- Iterations until appropriate component behavior has been found

Model checking:
\[ M_{\text{RearPort}} \parallel M_{\text{FrontPort}} \parallel M_{\text{Synchronization}} = \psi \land \neg \delta \]

with \( \psi = (\text{RearPort.convoy} \Rightarrow \text{CanBrakeFully}) \land (\text{FrontPort.convoy} \Rightarrow \neg \text{CanBrakeFully}) \)

After Composition?

Theorem (in [ESEC03]):
All systems build of verified components that cooperate only via verified patterns via correctly refined pattern roles (= syntactical correct composition) will behave correctly.
Incremental Design and Formal Verification with UML/RT in the FUJABA Real-Time Tool Suite

Holger Giese
University of Paderborn
Software Engineering Group

III. Mapping (RTSC $\Rightarrow$ ExHTA $\Rightarrow$ HTA)

(a) RTSC

(b) ExHTA

(c) HTA

ExHTA $\Rightarrow$ HTA: Priorities

Asynchronous Messages & History
Abstract form Methods …

\[ S_1 \xrightarrow{\text{convUseful}} S_2 \]

\[ S' \xrightarrow{\text{convUseful}=false} S \xrightarrow{\text{convUseful}=true \land \text{convUseful}=true} S' \]

Mapping Process

Compositional reasoning:
- Pattern mapping (closed model):
  - Map role statecharts
  - Map connector
  - Parallel composition
- Component Mapping (open model):
  - Map port statecharts
  - Special treatment of external events (\(\text{\&}\) closed model)
  - Map synchronization statechart
  - Parallel composition

Also other submodels …

Remark: Currently no OCL Mapping \(\Rightarrow\) TCTL is used

IV. Incremental Design & Verification

- Consistency Management detects updates
- Model Checking of modified parts (background)
- Modes: \(\bigcirc\) true, \(\bigotimes\) false, \(\bigotimes\) unsafe (not up-to-date)

Conclusion

- Sufficient UML/RT subset (mechatronic control)
- Compositional Verification with Patterns & Components
- Automatic mapping of UML/RT models to HUPPAAL
- Consistency management for properties \(\Rightarrow\) incremental and iterative design and verification

Current Work

- Other model checking back ends (RAVEN)
- Hybrid UML models (presented on FSE04)

Future Work

- Counter examples visualization