TURTLE
An Environment for the Modeling and Validation of Protocols and Real-Time Systems

Ludovic Apvrille
ludovic.apvrille@enst.fr
Eurecom, Office 223

Outline

- Context
- TURTLE, A UML profile
- TURTLE’s extensions
  - Extensions for real-time systems
    - At design level
  - Extensions for protocols and distributed systems (TURTLE-P)
    - At analysis level
    - At design level
  - Extensions taking into account UML 2.0 (TURTLE 2.0)
- The TURTLE Toolkit
- Future work
Context

- Design of real-time embedded system is complex
  - Equipments’ heterogeneity
  - Functionalities to offer are more and more complex
- Actual methodologies
  - Are informal (e.g. UML)
    - No formal validation
  - Take into account a limited amount of constraints
    - Real-time constraints
- Formal methods
  - Hardly no industrial use

Propositions

- Idea: let us enrich UML
  - UML operators are informal
  - UML lacks advanced temporal operators such as time intervals
  - UML has no methodology (no validation)
- Proposition: Semi-formal UML-based environment
  - Semantics given by mapping to a Formal Description Technique
- What formal language?
  - Well-defined formal semantics
  - Logical and temporal operators
  - Tools
    - RT-LOTOS / Petri Nets with real-time extensions
    - TURTLE UML profile (Timed UML and RT-LOTOS Environment)
TURTLE: A UML Profile

Chronology of TURTLE

1999
- First definition of operators

2000 - 2001
- Definition of a methodology supporting validation
- Modeling and translation rules
- Translation from TURTLE to RT-LOTOS partially implemented

2002
- New operators (temporal operators, new diagrams)
- Methodological extensions

2003
- First release of the TURTLE toolkit (Ttool)

2004
- TURTLE 2.0
  - UML 2.0-based extensions
Labs and People Involved in TURTLE

- LAAS / CNRS
  - Jean-Pierre Courtiat
- ENSICA
  - Pierre de Saqui-Sannes
- Concordia University
  - Ferhat Khendek
  - Christophe Lohr
- ENST
  - Ludovic Apvrille
- Alcatel Space Industries

References

- Definition of the profile

- Use of the profile
TURTLE : Comparison with UML 1.5

UML 1.5

- Class diagram
  - Parallelism is implicit
  - Associations = documentation

- Behavior diagram
  - Operation calls
  - Delay with pre-determined duration

- Industrial tools
  - Implementation-oriented simulation
  - Sequence diagram based testing

TURTLE

- Extended class diagram
  - Explicit parallelism
  - Explicit association between classes (parallelism, synchronization through gates, etc.)

- Extended activity diagrams
  - Data sending/receiving on gates
  - Advanced temporal operators
    - Time-intervals

- Tools
  - TTool + RTL
  - Generation of reachability graphs

TURTLE Class Diagrams

TClass Stereotype

- Tclass Id
- Attributes
- Gates
- Operations
- Behavior Description

Gate Abstract Type

- Parallel
- Synchro
- Sequence
- Preemption

Composer Abstract Type

- Composition operator
- Association

Gate

InGate

OutGate

TC1

TC2

TURTLE Activity Diagrams

Temporal operators

Synchronization

\[ g \text{x?y} \]

AD

\[ d \quad \text{AD} \]

\[ t \quad \text{AD} \]

\[ \text{dmin} \]

\[ \text{dmax-dmin} \]

AD

AD

AD1

AD2

\[ g @ d \]

A D

Methodology

Translation to RT-LOTOS

Validation (RTL)

Intensive simulation

Reachability analysis

Formal proofs

Design

Class diagram

Activity diagram

Object diagram

Analysis

Use case

Scenarios
Example

Example (Cont.)
Reachability Graph

TURTLE’s Extensions
Extending TURTLE

"Native TURTLE"

- 4 composition operators
- 4 temporal operators

"TURTLE for RT Systems"

- 3 composition operators
- 4 temporal operators

Parallel
Synchro
Sequence
Preemption
Deterministic delay
Non deterministic delay
Time-limited offering
Time capture

"TURTLE for Protocols" TURTLE-P

Invocation
Periodic
Suspend
Suspendable deterministic delay
Suspendable non-deterministic delay
Suspendable time-limited offering
Suspendable Time capture

"TURTLE for Multimedia Documents"

Invocation

BrowserWindow + getpage : Gate

getpage Iport Iurl ?data

Comm + request : Gate

request ?port ?url

{ Browser.getpage = Comm.request }

Page 9
Periodic

Suspend / Resume
Suspend: Example

Suspendable Temporal Operators

- Suspensable delay
- Suspensable non-deterministic delay
- Suspensable time-limited offer
- Suspensable time capture
Suspendable Temporal Operators

Basics of TURTLE-P

Analysis
Use case
MSCs

High-level design
Class diagrams
Activity diagrams

Low-level design
Component diagrams
Deployment diagrams

Consistency
Analysis of logical and temporal behavior

Ttool + RTL
Reachability graph
Simulation traces

TURTLE-P translation algorithms + RTL
Reachability graph
Simulation traces
TURTLE-P Deployment Diagram

Case Study
Low-level Design

Ludovic Apvrille - February 9, 2004. Page 14

Structuring with TURTLE 2.0

Behavior Modeling with TURTLE 2.0

Activity diagrams

Statecharts

TTool: The TURTLE Toolkit
Formal Validation Process with TTool

Simulation or reachability analysis

Code generator

RT-LOTOS specification

Simulation or verification

Result analyzer

Results

TTOOL

RTL

This process can be automatically performed from Ttool; it is hidden to UML users.

Screen Captures of TTool
Future work

- Refinement process at specification and design level
- TURTLE 2.0
- New operators
  - Explicit support of hardware and software components
    - Bus
  - Support of multimedia components
- Formal semantics
  - Petri Nets
- Toolkits
  - TINA
  - Support of TURTLE’s extensions under TTool

Any questions?

- TURTLE’s website:

- TTool’s website: