Composing RT Objects: A Case for Petri Nets and Girard’s Linear Logic

François Taïani, Mario Paludetto (LAAS-CNRS, Toulouse, F), Jerôme Delatour (ESEO, Angers, F)
Plan

• Introduction: Time, Objects and Composition
• Objects and Petri Nets. A good Match?
• Using a Mutant Logic to the Rescue
• Example
• Conclusion and Outlook
Time, Objects and Composition

- Object = “Divide and Conquer” Unit
  (cross-project: reuse, cross-company: COTS)
- Integration Pitfall $\Rightarrow$ Contract Based Development
- Real Time OO Systems $\Rightarrow$ Time Sensitive Contracts
- (Time Sensitive) Reuse Question: “Does that Object match my (Time Sensitive) Needs?"

Checking of Temporal Interoperability
Object / Environment needed.
Objects and Petri Nets

• Petri Net = Good for Control, Bad for Data
• Petri Net in Object:
  – Abstraction of Actual Behavior
  – Orders Object Needs / Services naturally
    (Exclusion, Precedence, …) (Logical Time)
  – Timed Transitions (Quantitative Time)
  – Token flows ≈ Threads

François Taïani
(LAAS-CNRS)

ISORC 2001
May 2 - 4, Magdeburg
Using a Mutant Logic to the Rescue (I)

“... in which truths get lost once they are used.”

[Girard90]

“A at time $\alpha$ with C at time $\gamma$”

$A(\alpha), C(\gamma)$

“(A and C) produces B, in duration $\delta$”

$[A \otimes C \rightarrow B](\delta)$
Using a Mutant Logic to the Rescue (II)

• A Rewriting Step:

\[ A(\alpha), C(\gamma) ; [A \otimes C \rightarrow B](\delta) \]

\[ B(\max(\alpha, \gamma) + \delta) ; \langle\text{empty list}\rangle \]
Using a Mutant Logic to the Rescue (III)

• Girard’s Linear Logic $\approx$

  PN Unfolding with $(\max, +)$ Analysis

• Decisive Features:
  – Symbolic Duration Times
  – Multi-Instantiation of the same Structure ($\approx$Threads)
  – (Finite) Loops
  – No State Explosion due to Concurrency
Using a Mutant Logic to the Rescue (IV)

- Multi-Instantiation of the same Structure

\[
A(0), A(0) ; 2 \times [A \to B](\delta) \\
\sim \succ B(\delta), A(0) ; [A \to B](\delta) \\
\sim \succ B(\delta), B(\delta) ; .
\]

\[\Rightarrow \text{WCET} = \delta\]

\[
A(0) , A(0) , C(0) ; 2 \times [A \otimes C \to B \otimes C](\delta) \\
\sim \succ A(0) , C(\delta) , B(\delta) ; [A \otimes C \to B \otimes C](\delta) \\
\sim \succ B(2\delta) , C(2\delta) , B(\delta) ; .
\]

\[\Rightarrow \text{WCET} = 2\delta\]
Using a Mutant Logic to the Rescue (V)

- No State Explosion due to Concurrency
  - Collection of Partial States
  - No Assumptions about possible “Simultaneity”
    ... until explicit Synchronization needed
  ↓
  - Plain Application of Rules yields Correct Result
  - Case Distinction only at “Conflict Points”
Example: Problem

\{ \text{g.executionTime} < \delta \}
Example: Results

\[ t_1 = \nu + \max\left( 2\mu + \kappa + \max(\iota, \pi + \alpha + \kappa), \nu + \lambda + \max(\mu + \max(\iota, \pi + \alpha + \kappa), \pi + \lambda) \right) \]

\[ \Rightarrow \{ \text{g.executionTime} < \delta \} \text{ fulfilled as soon as } \]

\[ \text{WCET}_{\text{model}} = \max(t_1, t_2, t_3, t_4, t_5, t_6) < \delta \]

= Conflict (i.e. Choice) Point

= Plain Rewriting Sequence:

- something ; something
- \( \rightarrow \) something ; something
- \( \rightarrow \) something ; something
- \( \rightarrow \) something ; something
Conclusion and Outlook

• Temporal Constraint at one Object Interface → Temporal Requirements on whole System

• Further Developments:
  – Multiple Constraints? Choice of Context?
  – Scalability Study on Larger Examples
  – Tool Development (started)
  – Connection to Concrete Distributed OO Frameworks.
Annex: Partial Orders

- Interleaving: $t; s$ and $s; t$
- Girard's LL:
  
  If no Conflicts:
  
  \[
  \rightarrow \text{All rewriting Sequences are equivalent.}
  \]
  
  \[
  \rightarrow \text{Only one does it all:}
  \]
  
  \[
  A(\alpha), C(\gamma); T, S
  \]
  
  \[
  \rightarrow B(\alpha+\tau), C(\gamma); S
  \]
  
  \[
  \rightarrow B(\alpha+\tau), D(\gamma+\varsigma); .
  \]
Annex: Potential Conflict Point

Question 1: Fire $t$ ?, or wait for possible $C$ ?
Question 2: Can a $C$ be produced without $t$ being fired?

→ Solution: Fire $t$, set flag to study causality $t \rightarrow C$, and identify actual conflict afterwards, if any.