Graph Transformation-based Stochastic Simulation

Paolo Torrini
joint work with Reiko Heckel and Ajab Khan
pt95@mcs.le.ac.uk
University of Leicester
Network reconfiguration

- Graph Transformation Systems as modelling technique
- Stochastic simulation as validation technique
- Application: simulation of system reconfiguration and behaviour
- Present focus: reconfiguration of P2P networks (previous work by Reiko), VoIP applications — esp. Skype (Ajab’s PhD)
- Semantics: stochastic GTS interpreted as stochastic processes (previous work by Reiko et al.)
- Implementation: based on existing GTS tools (VIATRA)
Modelling networks

- Network as a typed graph (SPO): components as nodes, connections as edges

- Transformations in our model: basic behaviour
  - sending packets
  - checking QoS parameters
  - connection/disconnection of clients
  - reconfiguration
  - change of node status, client-supernode and overlay connections to improve QoS
Stochastic approach

- Probabilistic rather than indeterministic actions
- Simulation based on sampling delay values according to given distributions
- Individual processes determined by Random Number Generator
- Rule application associated with an expected delay according to a probability distribution function
Probability distributions

- Expected delay (timer) — random variable associated to probability distribution function

  $$F_T(x) = P\{T \leq x\}$$

- Exponential distribution — determined by a rate, depends only on the present state

- Normal distribution — determined by mean and variance, finer modelling
Representing time

- Each component has a clock — *chronos* attribute
- *chronos* rule to advance time
- Normally distributed
  — exponential distribution would not do

```
<table>
<thead>
<tr>
<th>n:Node</th>
<th>chronos</th>
<th>n:Node</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>= t</td>
<td></td>
</tr>
<tr>
<td></td>
<td>chronos rule</td>
<td></td>
</tr>
<tr>
<td></td>
<td>= t + 1</td>
<td></td>
</tr>
</tbody>
</table>
```
GTS with probability

- Stochastic Graph Transformation Systems: rules associated with exponential probability distributions

- Generalised Stochastic Graph Transformation Systems: rule matches associated with general probability distributions

- Rule matches as equivalence classes to preserve inter-graph identity — restrictions on GSGTS to ensure they are a proper set (numbered graphs)

- Continuous time — timers are independent variables, so probability of two actions taking place at the same time is 0. So we skip parallelism
**Stochastic processes**

- Continuous-time stochastic process — time-indexed family of random variables over states

- Markov process: next state depends on current state only, interevent time is exponentially distributed

- Semi-Markov process: next state depends on time spent in current state, too

- Translation of GSGTS to Generalised Semi-Markov Processes
Stochastic structures

- Generalised Semi-Markov process: generated by a structure based on timers + race condition
- Timers as stochastic clock structure: Stochastic Timed Automata
- Timers set by RNG: Generalised Semi-Markov Scheme
- Timers do not need to be reset at each state change i.e. they do not need to be exponential — so neither interevent time do
GSGTS

- SPO — components may disconnect without warning
- $\mathcal{G} = \langle TG, P, \pi, G_0, F \rangle$
- Type graph, rule names, initial graph
- $F$ maps rule matches to probability distribution functions (cumulative distributions: — max delay value mapped to probability)
\[ P = \langle \begin{array}{l}
\text{States} \\
\text{Events} \\
\text{ActiveStates} : \text{State} \rightarrow \wp(\text{Event}) \\
\text{Transition} : \text{State} \times \text{Event} \rightarrow \text{State} \\
\Delta : \text{Event} \rightarrow (\mathcal{R} \rightarrow [1, 0]) \\
\text{InitialState} : \text{State} \end{array} \rangle \]
from GSGTS to GSMS

\[ \mathcal{P} = \langle \text{ReachGraphs} \]
\[ \text{RuleMatches} \quad (\text{equivalence classes}) \]
\[ \text{EnabledMatches} : \text{ReachGraph} \to \emptyset \text{RuleMatch} \]
\[ \text{GraphTrans} : \]
\[ \text{ReachGraph} \times \text{RuleMatch} \to \text{ReachGraph} \]
\[ F : \text{RuleMatch} \to (\mathcal{R} \to [1, 0]) \]
\[ \text{InitialGraph} : \text{ReachGraph} \quad \rangle \]
GSMS-based simulation

- Simulation of GSMP based on Event Scheduling Scheme algorithm
- Refinement of ESS
  — for GSMP obtained from GSGTS
- Substantial problem: computation of active matches
  — we need to keep track of pre-existing matches
- Incremental pattern-matching
  — implemented in VIATRA (Eclipse plug-in)
- Planned Java implementation using SSJ libraries for RNG
Implementation architecture

- Graph transformation tool (GTT)
- Simulation control (SC)
- Random number generator (RNG)
Scheduling Scheme I

- Initial input:
  - graph transformation system (for GTT)
  - simulation time $t$ (for SC)
  - probability distribution functions (for RNG)

- Initialisation:
  - active matches of the initial graph computed by GTT
  - associated to scheduled delay $d$ (from RNG)
  - scheduled time = $t + d$
  - collected in list ordered by scheduled time

- list ordering implements race condition
Simulation Control

1. First item \((r, m, t')\) removed from list, simulation time updated to \(t'\)

2. Graph update (by GTT)
   — rule \(r\) applied to the current graph at \(m\)
   — new and surviving matches are computed
     (incremental approach should help)

3. New matches are associated to scheduling times
   — RNG calls

4. list ordering
Further work

- Containment relations and spatial aspect — to model domains, firewall restrictions, geographic locations

- *Chronos* rule application overkill: two possible strategies
  — temporal granularity with laziness
  — synchronous approach (global time)

- Comparison with existing simulation tools (such as NS2)

- A. Kahn, P. Torrini, R. Heckel — *Model-based Simulation of VoIP Network Reconfigurations using Graph Transformation Systems*
  Doctoral Symposium ICGT 2008 (post-proceedings, 2009)