QoS Provisioning and Orchestrating Processes within an SOA

Bill Mitchell
IT Innovation Centre,
School of Electronics and Computer Science
University of Southampton
IRMOS overview

- Design, develop and validate a Service Orientated Infrastructure which will allow the adoption of interactive real-time applications, and especially multimedia applications.

Social networking, education

Augmented Reality

Film post production
Intuition: think of distributed Amazon EC2 and S3 with guaranteed workflow and guaranteed realtime QoS

- Guaranteed virtual resource QoS:
  processor, RAM, storage, bandwidth, latency, jitter, ...

- Guaranteed reservations:
  resources will be available during reservation time frame

- QoS is monitored and verifiable during runtime
Questions for modelling and verification

Before developing application into IRMOS service
Estimate performance characteristics

Why bother?
not possible to increase resources or time interval during execution
Questions for modelling and verification

Before developing application

- Time frame reservations required for workflow and components
- Application workflow choreography; verified with respect to
  - resource contention,
  - resource utility,
  - causal dependencies,
  - deadlock,
  - livelock, etc,
Black Box Process Component Performance

- Stochastic correlations from benchmarking and partial knowledge of algorithms.

![Diagram showing input, black box, and output with QoS input and output data estimates](image)
Streaming Real Time Processes

State machine pattern for streaming data
Streaming Real Time Processes

Stochastic interaction events and data
Termination Estimate

- Given stochastic complexity and input data
- If executing at time \( n \) what’s probability of still running at \( n+1 \)?

![Diagram showing data distribution and process data chunk]

- Estimate probability
- QoS input data
- QoS output data

 termination by time \( n+1 \)
Discrete time stochastic automaton

- has standard semantics
- formally verify properties with stochastic model checkers
- can synchronise on stochastic events
rho(n) for known complexity

- Data is Borel measurable space BV
- Data estimates are probability measure p over BV
- Time to execute black box is f(v) for v in BV
- Define
  \[ T(f, t) = \{ v \in BV \mid f(v) > t \} \]
- If executing at time t, probability of still executing at t+d
  \[ \rho(t + d) = \frac{\int_{v \in T(f(t+d))} p(v) dv}{\int_{v \in T(f,t)} p(v) dv} = \frac{p(T(f, t + d))}{p(T(f, t))} \]
  probability of complexity greater than t+d
  probability of complexity greater than t
Estimated Complexity Discrete Time

Time complexity = \( \text{comp}(d) \)

Data size \( d \)

Probability \( q(t,d) \)

\( \Pr( \text{comp}(d) > t \mid \text{input size is } d) = \sum_{t > t_0} q(t,d) \)

\[ \Pr\{d \in BV \mid \text{comp}(d) > t\} = \int_{d \in BV} \left( \sum_{t > t_0} q(t,d) \right) dp \]
Estimated Complexity Discrete Time

Time complexity
\[ = \text{comp}(d) \]

Probability \( q(t,d) \)

Data size \( d \)

\[ \rho(n) = \frac{\int_{d \in BV} \left( \sum_{t>n} q(t,d) \right) dp}{\int_{d \in BV} \left( \sum_{t>n-1} q(t,d) \right) dp} \]
rho(n) for uniform distribution linear complexity

Uniform data distribution

\[
\rho(n) = \frac{u - n}{u - (n - 1)}
\]

For time \( n \leq u \)

\[
1 - \left( \frac{u - n}{u - (n - 1)} \right)
\]
Uniform plus linear termination estimate

Probability of terminating at time $n$

$$= \rho(0) \cdots \rho(n-2)\rho(n-1)(1-\rho(n))$$

$$= \left(\frac{u-1}{u-0}\right)\left(\frac{u-2}{u-1}\right)\left(\frac{u-3}{u-2}\right) \cdots \left(\frac{u-(n-2)}{u-(n-3)}\right)\left(\frac{u-(n-1)}{u-(n-2)}\right)\left(1-\frac{u-n}{u-(n-1)}\right)$$

$$= \left(\frac{u-(n-1)}{u}\right)\left(1-\frac{u-n}{u-(n-1)}\right) = \frac{1}{u}$$
Exact complexity termination estimate

\[
\rho(n) = \frac{\Pr(T(f,n))}{\Pr(T(f,n-1))}
\]

Estimate that automaton terminates at time \( n = \)

\[
\Pr(T(f,0)) \left( \frac{\Pr(T(f,1))}{\Pr(T(f,0))} \right) \left( \frac{\Pr(T(f,2))}{\Pr(T(f,1))} \right) \left( \frac{\Pr(T(f,3))}{\Pr(T(f,2))} \right) \ldots
\]

\[
\ldots \left( \frac{\Pr(T(f,n-2))}{\Pr(T(f,n-3))} \right) \left( \frac{\Pr(T(f,n-1))}{\Pr(T(f,n-2))} \right) \left( 1 - \frac{\Pr(T(f,n))}{\Pr(T(f,n-1))} \right)
\]

\[
= \Pr(f(d) > n-1 \mid d \in BV) - \Pr(f(d) > n \mid d \in BV)
\]

\[
= \Pr(n \geq f(d) > n-1 \mid d \in BV)
\]
Elementary timed workflow

Arbitrary Workflow synchronised with process and clock can deadlock.
Synchronisation and stochastic properties have to be arranged correctly.
PRISM tool semantics require additional idle and termination loops.
Correct Workflow Semantics

- For any DAG workflow with consistent time frames
- For any black box process with given stochastic complexity and data
- There is a timed non-stochastic workflow automaton that
  - is deadlock free in composition with processes and clock
  - imposes correct causal dependencies between processes
- Extends to cyclic graphs by extending timed finite automata for black box processes
- Workflow can also be extended to allow for failure, cancellation and restarts among processes.
module Activity

[tick] (x=idle) -> (x'= idle);
[startA] (x=idle) -> (x' = go);
[] (x=go) -> rho_iterate0:(x'=add_one_to_count)
         + (1 - rho_iterate0):(x'=terminate);
[] (x=add_one_to_count) ->
         (x'=exec)&(count0'=incrmnt_count0);
[run] (x=exec) -> (x'=return);
[tick] (x=return) -> (x'=go);
[end] (x=terminate) -> (x'=stop);
[tick] (x=stop) -> (x'=stop);
endmodule

formula rho_itr0 = rho0*(1- count0/interval0) +
                   rho1*count0/interval0;
formula rho_iterate0 =
    (((count0 >= 0) & (count0 <= interval0)) ? rho_itr0 : rho1;
Colour Correction Conceptual Model

No idea what these will be yet, it is extremely abstract

- Process component(s)
- footage
- CC software tool
- real-time streaming
- Post House 1

Conceptual level so we don’t know
- What type of buffering exists
- How data is transcoded for data link
- How data link is shared
- How data link is managed

But we can make a model that gives predictions!
Stochastic Data

The probability of having to CC amount outside range 30 to 40 is approximately 20%.

Black box complexity is linear. If bandwidth is 'too low' for a continuous period (say 1 minute) rewind is necessary.
QoS model of streaming data link

[<<probability>> down a bit] / subtract a bit

[<<probability>> up] / add a bit

[<<probability>> no change]
Predictions

If bandwidth OK 100% of time

If bandwidth OK for 98% of time
Predictions

![Graph showing predictions with different rho_b values](image)
Timed stochastic process algebra

- **Process terms**
  - \([t_0 < t < t_1] \rightarrow P\) if clock is in given range then do \(P\)
  - \([? t < n : r=r'] P\) if clock \(t\) is less than \(n\) set \(r\) to \(r'\) then do \(P\)
  - \(\rho_1 . P_1 + \ldots + \rho_n . P_n\) there is probability \(\rho_i\) that will do \(P_i\)
  - \(e \bullet P\) first do \(e\) next do \(P\)
  - \(P + Q\) do exactly one of \(P\) or \(Q\) depending on which is enabled
  - \(P \parallel Q\) interleave \(P\) and \(Q\)
  - \(P \parallel_s Q\) synchronous interleaving on event set \(S\)
Timed Black Box Process Term

\[ Pr(a) = \Omega \cdot Pr(a) + start_a \cdot run(a) + cancel_a \cdot fail_a \cdot 0 \]

\[ run(a) = [t_0^a \leq t \leq t_1^a] \rightarrow (\rho_a \Omega \cdot run(a) + (1 - \rho_a) \cdot done_a \cdot 0) \]
\[ + ([t > t_1^a] \rightarrow fail_a \cdot 0) \]
\[ + (cancel_a \cdot fail_a \cdot 0) \]

\[ Clock_0 = [: t = 0]Clock \]

\[ Clock = [: t = t + 1]\Omega \cdot Clock \]
Iterative Black Box Processes

\[
\Pr(a) = \Omega \cdot \Pr(a) + \text{start}_a \cdot \text{getrs}(a) + \text{cancel}_a \cdot \text{fail}_a \cdot 0
\]

\[
\text{getrs}(a) = ([r_a = f] \rightarrow \Omega \cdot \text{getrs}(a)) + ([r_a = t] \rightarrow [r_a = f] r_n_a \cdot \text{run}(a))
\]

\[
\text{run}(a) = [t_0^a \leq t \leq t_1^a] \rightarrow (\rho_a \Omega \cdot \text{run}(a) + (1 - \rho_a) \cdot [r_a = t] \text{done}_a \cdot \text{iter}(a))
\]

\[
+ ([t > t_1^a] \rightarrow [r_a = t] \text{fail}_a \cdot 0)
\]

\[
+ (\text{cancel}_a \cdot [r_a = t] \text{fail}_a \cdot 0)
\]

\[
\text{iter}(a) = (exp(a) \mapsto \Pr(a)) \parallel s_y \prod_{a' \in \text{nxt}(a, G)} (exp(a') \mapsto \text{start}_{a'} \cdot 0)
\]