Modelling interfaces in distributed systems: some first steps

David Pym
UCL and Alan Turing Institute
London
Modelling distributed systems: basic concepts

• Basic concepts of distributed systems
  – Location: the basic architecture
  – Resource: created, consumed, moved by the processes
  – Process: the services that the system provides

• Situated in
  – Environment: structure not modelled, just events

• These may be composed partially of other models of interest, so need composition

• Mathematically, seek to employ minimal viable structure

• Concerned here with practical modelling, with motivations from security policy
Modelling distributed systems: basic mathematical set-up

• Location
  – Topological structure: e.g., directed graphs

• Resource
  – Combinatorial structure: e.g., partial monoids, possibly ordered (cf. the logic Bi’s resource semantics, which gives rise to Separation Logic)

• Process
  – Synchronous structure (for modelling purposes): e.g., SCCS + integration with resources

• Environment
  – Stochastic representation: events are incident upon a model system from outside
Modelling distributed systems: basic mathematical set-up

- Basic operational judgement:
  \[ L, R, E \xrightarrow{a} L', R', E' \]

- Some rules (omitting locations for brevity):
  \[
  \begin{align*}
  \mu(a, R) &= R' \\
  R, a : E \xrightarrow{a} R', E \\
  S, F \xrightarrow{b} S', F' \\
  R \otimes S, E \times F \xrightarrow{ab} R' \otimes S', E' \times F'
  \end{align*}
  \]

- A bunch of laws for \( \mu, \otimes, \) and \( \oplus \)
- Resource-process equivalence is bisimulation, written \( \sim \)
- Cf. Concurrent Separation Logic
A (bunched) modal logic

\( \phi ::= p | \bot | \top | \phi \lor \phi | \phi \land \phi | \phi \rightarrow \phi \\
| \langle a \rangle \phi | [a] \phi \\
| I | \phi \ast \phi | \phi \rightarrow \ast \phi \\
| \langle a \rangle \nu \phi | [a] \nu \phi \)

In a given model, a truth-functional judgement: \( R, E \models \phi \)

\( R, E \models \phi_1 \land \phi_2 \) iff \( R, E \models \phi_1 \) and \( R, E \models \phi_2 \)

\( R, E \models \langle a \rangle \phi \) iff for some \( R, E \xrightarrow{a} R', E', R', E' \models \phi \)

\( R, E \models \phi_1 \ast \phi_2 \) iff for some \( R_1 \otimes R_2 = R \) and \( E_1 \times E_2 \sim E, R_1, E_1 \models \phi_1 \) and \( R_2, E_2 \models \phi_2 \)

\( R, E \models \langle a \rangle \nu \phi \) iff for some \( S, S' \) s.t. \( R \otimes S, E \xrightarrow{a} R' \otimes S', E', R' \otimes S', E' \models \phi \)

Other similar things, some choices for the last one
Basic meta-theory

• Logical (declarative) equivalence:
  
  \[ R_1, E_1 \equiv R_2, E_2 \text{ iff for all } \phi, R_1, E_1 \models \phi \text{ iff } R_2, E_2 \models \phi \]

• Bisimulation (operational) equivalence:
  
  \[ R_1, E_1 \sim R_2, E_2 \]

• Soundness and completeness (Hennessy-Milner-van Bentham equivalence):
  
  for all \( R_1, E_1 \),  \( R_1, E_1 \sim R_2, E_2 \text{ iff } R_1, E_1 \equiv R_2, E_2 \)
Basic meta-theory

• Hennessy-Milner completeness is not as straightforward as might perhaps be imagined
• In basic resource semantics, based on ordered monoids of resource elements, it holds only for fragments of the modal logic
• Multiplicative implication and multiplicative modalities problematic
• Need the combinatorial structure of $\oplus$ and $\otimes$ to track evolutions of + and x
• Several papers (MSCS, TCS, JLC, others): http://www.cs.ucl.ac.uk/staff/D.Pym/recent.htm
Building models

• Classical mathematical modelling approach using these tools

• Early versions deployed with Hewlett-Packard and its customers, and more recently in projects in the GCHQ RISCS

• Currently aiming for policy modelling apps in the Turing Institute; lots of big industry partners

• Several papers at http://www.cs.ucl.ac.uk/staff/D.Pym/recent.htm

• julia code at https://github.com/tristanc/SysModels
Aside: building models

- Approach is essentially scale-free
- Abstraction level therefore chosen to fit problem
- Predictions explored using simulations
- Model checking also possible (though much less developed at this point)
- The map is not the territory (Alfred Korzybski)
- Time-value of models
Example: security modelling
Interfaces: basic concepts

• Mediate composition of models
• Build on the structure of distributed systems models, quite pragmatically
• In practice, must reflect
  – the locations involved,
  – the resources involved, and
  – processes/actions crossing the boundaries
• Note that models are being substituted for environment
Interfaces: sketch of basic mathematical set-up

- Implement the distributed systems model:
  - Location graph labelled with resources
  - Explicitly identify actions with associated locations in interfaces
- Each model comes with a specified set of interfaces, specifying input/output locations, with associated actions
- Decent basic algebraic properties: commutative, associative composition of models with compatible interfaces
Interfaces: sketch of basic mathematical set-up

- Implement models as tuples
  \[ M = (G(V[R], E), A, P, L, I) \]

- Here
  - Graph with resource-labelled vertices
  - Sets of actions, processes, and located actions
  - A set \( I \) of interfaces

- An interface \( I \in I \) on a model is a tuple of (disjoint) input and output locations and located actions \((In, Out, L)\)
Example: security modelling
Interfaces: the frame property

• Supports compositional reasoning: $M_1 I_1 |_{I_2} M_2$

• The Frame Rule (think of Hoare’s program logic and CSL):

\[
\frac{\{\phi\} (M \xrightarrow{a} M') \{\psi\} \quad \{\psi \ast \chi\} (M | N \xrightarrow{a} M' | N) \{\psi \ast \chi\}}{\{\phi \ast \chi\} (M | N \xrightarrow{a} M' | N) \{\psi \ast \chi\}}
\]

$N \models \chi$, where $N \not\xrightarrow{a}$

• Side-condition restricts evolution to part of model not in the interface

• Correctness reasoning can then be restricted to the interfaces themselves

• This gives local reasoning about models in their global context; that is, compositionality
Example: security modelling
Next steps

• Refine definition of interface, useful abstractions
• Some underpinning logical theory
• The Frame Rule in theory and practice; cf. (Concurrent) Separation Logic’s theory and implementation of local reasoning: *abduction* important here?
• Applications to big-scale systems
  – Networking
  – Distributed databases and their consistency
  – Supply chains
• Deliver tools for reasoning about big-scale systems
• Small-scale systems: weak memory
Thank you
Modelling distributed systems: basic mathematical set-up

• Other key combinators
  – Hiding
    \[
    \frac{R \circ S, E \xrightarrow{a} R' \circ S', E'}{R, \nu S.E \xrightarrow{\nu S.a} R', \nu S'.E'}
    \]
    \[\mu(\nu S.a, R) = R'\]
  – Generalizes restriction (build a term model for resources; partial monoid of actions)

• Sequential composition

• Fixed points
A (bunched) modal logic

- Other logical operators
  - Additive and multiplicative quantifiers (over actions)
  \[ R, E \models \exists \nu x. \phi \iff \text{there exist } S, F, \text{ and } a \text{ s.t. } R, E \sim R, \nu S.F \]
  \[ \text{and } R \circ S, F \models \phi[a/x] \]

- Systematic logical treatment in recent joint work with Galmiche, Courtault, and Kimmel

- Applications in access control
  - Roles: \( E \propto F \)
  - Corresponding (via simulation) ‘says’ modality: \( \{E\} \phi \)
References


More references


• The julia package used for creating system models may be obtained from GitHub: https://github.com/tristanc/SysModels