Jobbers, sentient buildings and lions

A short walk into Robin Milner's tower

Jean Krivine
PPS lab. CNRS & Univ. Paris Diderot
Concurrency

- PhD «Reversible process algebra» at INRIA (sup. J JL, 2006)
- Transactional systems, distributed transactions, self assembly
- Formal methods in systems biology (Danos, Fontana)
- Bigraph theory, stochastic semantics (Milner)
Meeting a great scientist
Someone to employ, please?
Research in bigraphs
Robin Milner, July 2006

Bigraphs are a model that represents, roughly speaking, a synthesis of the mobile linkage of the π-calculus with the movement of regions in Mobile Ambients. The hope is that they describe realistic systems in which both software and hardware (people or sensors) move about. Think of an intelligent building, or a body sensor network monitoring your state of health. One would like to use such a model both for description (specification) and for programming.

An example
Here is an example, describing a simple interaction discipline that models sentient buildings — buildings whose infrastructure of sensors and computers assists the performance of human occupants.
Someone to employ, please?

An example

Here is an example, describing a simple interaction discipline that models sentient buildings—buildings whose infrastructure of sensors and computers assists the performance of human occupants.

The jobshop
Someone to employ, please?

An example

Here is an example, describing a simple interaction discipline that models sentient buildings—buildings whose infrastructure of sensors and computers assists the performance of human occupants.

The jobshop

Bigraphs are rigorous mathematical entities, but amenable to this kind of graphical representation.
Research in bigraphs

Robin Milner, July 2006

Bigraphs are a model that represents, roughly speaking, a synthesis of the mobile linkage of the π-calculus with the movement of regions in Mobile Ambients. The hope is that they describe realistic systems in which both software and hardware (people or sensors) move about. Think of an intelligent building, or a body sensor network monitoring your state of health. One would like to use such a model both for description (specification) and for programming.

An example

Here is an example, describing a simple interaction discipline that models sentient buildings — buildings whose infrastructure of sensors and computers assists the performance of human occupants.

Bigraphs are rigorous mathematical entities, but amenable to this kind of graphical presentation used here. Each of the upper and lower diagrams is a bigraph representing the state (much simplified) of two buildings and their occupants. A reconfiguration changes one into the other.

Buildings are designated $B$. Each building here contains two rooms $R$, and each room is equipped with a sensory computer $C$. Moreover, the infrastructure of each building links all its computers, allowing them to communicate. Finally, agents $A$ perform the appropriate actions.
Robin Milner
Robin Milner

Comprehension axiom

Robin always understands what you say
Robin Milner

Comprehension axiom

Robin always understands what you say

Explains the gradient of interest:
Robin Milner

Comprehension axiom

Robin always understands what you say

«I don’t understand»

Explains the gradient of interest:
Robin Milner

Comprehension axiom
Robin always understands what you say

«I don’t understand»
... why we are still talking about this

Explains the gradient of interest:
Robin Milner

Comprehension axiom
Robin always understands what you say

Explains the gradient of interest:

«I don’t understand»
... why we are still talking about this

«This is interesting»
Robin Milner

Comprehension axiom

Robin always understands what you say

Explains the gradient of interest:

«I don’t understand»
... why we are still talking about this

«That’s fascinating»

«This is interesting»
Robin Milner

Comprehension axiom
Robin always understands what you say

Explains the gradient of interest:

«I don’t understand»
... why we are still talking about this

«That’s fascinating»

«This is interesting»

Eyebrow scratching
Contribution

- Stochastic semantics for bigraphs (w Angelo Troina, Turin Univ.)
- BRS generators and application to systems biology
- (Beginning of an) abstract machine

Small contribution, but learned a lot...
A tower against the jungle
Someone to employ, please?

Finally, bigraphs represent the abstract as well as the concrete. For example, there is a BRS representing the $\pi$-calculus and another representing Mobile Ambients. By combining the abstract with the concrete we can, for example, describe both the physical and the informatic activity in a building.
Consider also a model of humans interacting with a computer; the model of the human components may involve human attributes such as belief or sensation, as distinct from the way the computer is described. These two examples show the need not only to combine informatic models, but to combine them with others that are not informatic.
In Milner’s Tower

Communication

Concurrency

Mobility

Space and motion
In Milner’s Tower

Communication

Mobility

Concurrency

Distributed computation

Space and motion

combine

describes
In Milner’s Tower

CCS

Communication

Concurrency

combine

describes

Mobility

Distributed computation

Space and motion
In Milner’s Tower

Communication

Mobility

Concurrency

Distributed computation

Space and motion
In Milner’s Tower

Communication

Mobility

Space and motion

Concurrency

Distributed computation

Distributed systems

CCS describes

Combine

Describes
In Milner’s Tower

Pi-Calculus \rightarrow CCS

Communication \leftrightarrow Concurrency

Mobility \leftrightarrow Distributed computation

Space and motion \leftrightarrow Distributed systems
In Milner’s Tower

- Pi-Calculus
- CCS

Communication
- Concurrency
- Mobility
- Distributed computation

Distributed systems

Space and motion
In Milner’s Tower

Pi-Calculus -> CCS

Communication combines Concurrency

Mobility

Distributed computation

Space and motion

Distributed systems

(Discrete) Complex systems
In Milner’s Tower

- Pi-Calculus
- Bigraphs
- Space and motion
- (Discrete) Complex systems

---

- Communication
- Concurrency
- Mobility
- Distributed computation
- Distributed systems

---

- CCS
- describes
- combine
- describes
In Milner’s Tower

- Pi-Calculus
- Bigraphs
- Space and motion
- Mobility
- Distributed systems
- (Discrete) Complex systems
- Communication
- Concurrency
- Distributed computation

Describes

Combines

CCS

Describes
Populating the tower
Populating the tower

\[ K_{\bar{x}}(\Box) : \langle \emptyset, 1 \rangle \to \langle \{x_1, \ldots, x_n\}, 1 \rangle \]
Populating the tower

\[ K_{\overline{x}}(\square) : \langle \emptyset, 1 \rangle \rightarrow \langle \{ x_1, \ldots, x_n \}, 1 \rangle \]

\[ \overline{y} : \langle \{ y_1, \ldots, y_n \}, 0 \rangle \rightarrow \langle \emptyset, 1 \rangle \]

\[ x \overline{y} : \langle \{ y_1, \ldots, y_n \}, 0 \rangle \rightarrow \langle \{ x \}, 1 \rangle \]
Populating the tower

\[ K_{\bar{x}}(\square) : \langle \emptyset, 1 \rangle \rightarrow \langle \{ x_1, \ldots, x_n \}, 1 \rangle \]

\[ \bar{y} : \langle \{ y_1, \ldots, y_n \}, 0 \rangle \rightarrow \langle \emptyset, 1 \rangle \]

\[ x \bar{y} : \langle \{ y_1, \ldots, y_n \}, 0 \rangle \rightarrow \langle \{ x \}, 1 \rangle \]

\[ \text{merge}_n : \langle \emptyset, n \rangle \rightarrow \langle \emptyset, 1 \rangle \]
Populating the tower

\[
K_x(\square) \mid L_x
\]
Populating the tower

\[ K_x(□) \parallel L_x \]
Populating the tower

Syntax

Bigraph toolbox

Bigraphs are lego pieces...

BRS

Rule set

that can be assembled to form a reactive system...

Instance

Initial bigraph

and applied on a particular initial bigraph.
Modeling the tea coffee machine... and the drinker
A simple tower: Hunting deers

Kids behaviors  Game of lions & deers
A simple tower:

Hunting deers

Kids behaviors

combine

describes

Game of lions & deers

Kids playing lions and deers
A simple tower: Hunting deers

bigraphical reactive systems

describes

Kids behaviors

specifies

Game of lions & deers

combine

Kids playing lions and deers

describes
A simple tower: Hunting deers

- bigraphical reactive systems
  - describes Kids behaviors
  - specifies Game of lions & deers
    - combine Kids playing lions and deers

- Game of lions & deers
  - describes Kids playing lions and deers
A simple tower: Hunting deers

A child/lion enters the game at spot \([xy]\)
(similarly for a deer)

A virtual lion moves to another spot
(similarly for a deer)

A child/lion becomes alert to a deer in its locale

\[/c (\text{child}_c \parallel (\text{lion}_{ac} \mid \text{deer}_b)) \longrightarrow /c (\text{childalert}_c \parallel (\text{lionalert}_{ac} \mid \text{deerseen}_{bc}))\]
Such an analysis can often be based upon a specification in logical model, perhaps at a formal level; the logical sentences echo those to characterise the desired behaviour of the embedded software (the program model). This software model has to be combined—as remarked earlier—with an electro-mechanical model of the plane, as well as a model of the plane's environment. Thus we arrive at a tower like that shown in Figure 4; of course this is only a simplification. The method chosen for analysis, based upon abstract interpretation, can be seen as a refinement of the logic-based approach. An abstract interpretation of a program is a simplification of the program, omitting certain details and making certain approximations, with the aim of rendering detailed analysis feasible. Such abstraction is essential in situations where the state-space is very large; but, to be sound, it must not conceal any undesirable behaviour.

Thus, instead of choosing a fixed specification, one may choose a number of abstractions specifically to match those aspects of behavior that are essential. In the case of the Airbus, by a careful choice of different abstractions, the analysis required to validate the embedded programs was reduced to the extent that, with the assistance of programmed tools, it could be completed.

The Airbus example illustrates that explanations and their validation can be customised within the framework of a tower of models. It also illustrates the importance of programmed analytical tools.

This concludes our examples. It is a good moment to answer a possible criticism of our notions of model and explanation. The criticism is that whenever a model is defined, the meaning of its entities—which are often symbolic—has to be expressed in some formalism; thus a model does no more than translate between formalisms, and our search for real meaning leads to an infinite regress.

Our answer is in two parts. First, every model-designer clearly has some meaning in mind. A programming language, or a logic, or a process calculus, or a graphical representation is never just a class of symbolic entities; its intended behavior is always described, even if informally. Thus it is clearly inadequate to call such a class a model.
Aircraft realised by Aircraft designs explained by electro-mechanical design.

Abstract interpretation explains embedded programs.

Figure 4: A simplified model tower for aircraft construction.
Where should we be?

Semantician

Bigraph toolbox

Set of generators

Expert of M

Describes the «laws» of M

Programming by adding context to generators

GUI
Where should we be?

Semantician

Bigraph toolbox

Set of generators

Expert of M

Describes the «laws» of M

Programming by adding context to generators

GUI

eyebrow scratching...
OK let me show you...
Can we provide a language that biologists can use describing these facts?
Can we provide a language that biologists can use describing these facts?
Can we provide a language that biologists can use describing these facts?
Laws

Can we provide a language that biologists can use describing these facts?
Can we provide a language that biologists can use describing these facts?
Can we provide a language that biologists can use describing these facts?
Can we provide a language *that biologists can use* describing these facts?
Fig. 1: Generators for $C_0$. 

We have introduced so far a simple calculus that rewrites proteins structured as connected domains. Proteins can be connected to each other in complex formation or new domains can be fused to proteins in protein synthesis or severed in transmembrane proteins can be cleaved to emit signals in the intercellular medium. This calculus is fairly abstract in the sense that two proteins may only differ in the number of domains they have and in the number of sites these domains possess. It is clear that we lack means of naming molecular components such as domain names (SH wed, Tyrosine, PWWP etc.) or protein names (SOS, EGF, IGF, p53 etc.). Before performing a bigger increment in expressiveness, when we introduce compartments in Section y of p, we would like to briefly introduce a way to deal with names as a particular type of context in which unamed proteins can be embedded. The intent is to provide a way to define molecular reactions as refinements of the generators we have just presented, in keeping with the biological intuition that information about molecular objects is always partial and that more context could reveal more about the nature of a molecule. In particular we have the ontology problem in mind that several names can denote the same protein or genes.

3.1 Terms

Consider a new set of names $M$ that is pairwise disjoint from $B$ and $S$. Terms of $C_1$ are essentially those of $C_0$ where domains have an extra meta-name $m$, $m \in M$ that will point to new types of terms called info-terms (think of protein or domain names). The grammar of $C_1$ is:

$$D, D \rightarrow D^{a_m}_{x_1 \ldots x_k} \left( \begin{array}{c} y_1 \ldots y_q \end{array} \right) a_j D^{b_i}_{y_1 \ldots y_q}$$

$$I, J, \ldots \rightarrow \text{Info}^{m}_{a}$$

$$T, S \rightarrow t | D | I | \ldots \left( \begin{array}{c} t | S \end{array} \right)^{m} | T \downarrow v$$

for $v \in \lambda S \rightarrow \lambda B \rightarrow \lambda M$. 

Generators for PPI
Generators for membrane

<table>
<thead>
<tr>
<th>Informal</th>
<th>Left hand side</th>
<th>Right hand side</th>
<th>Informal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
<td></td>
</tr>
</tbody>
</table>

\[
D^a(x_1, \ldots, x_k) \parallel D^b(y_1, \ldots, y_q) \rightarrow D^a(x_1, \ldots, x_k) \parallel D^a(y_1, \ldots, y_q)
\]

\[
0 \parallel 0 \rightarrow (\text{channel}_m \parallel C(\text{channel}_m)) \backslash m
\]

\[
(\text{channel}_m \parallel C(\text{channel}_m, X)) \backslash m \rightarrow X \parallel 0
\]
Generators for membrane

Informal | Left hand side | Right hand side | Informal
---|---|---|---

**Informal**

**Left hand side**

**Right hand side**

**Informal**

1. **fuse**
   
   \[ D^a(x_1, \ldots, x_k) \parallel D^b(y_1, \ldots, y_q) \parallel b \rightarrow D^a(x_1, \ldots, x_k) \parallel D^a(y_1, \ldots, y_q) \]

2. **pinch**
   
   \[ 0 \parallel 0 \rightarrow (\text{channel}_m \parallel C(\text{channel}_m)) \parallel m \]

3. **merge**
   
   \[ (\text{channel}_m \parallel C(\text{channel}_m, X)) \parallel m \rightarrow X \parallel 0 \]
Systems biologist names the biological entities

which become particular instances of the entities of the generators

\[
\langle P : 0, R : 2, R' : 2, \text{coat} : 1, \text{brane} : 0, S : 1, G : 1 \rangle
\]
Refinements of the PPI generators
Refinements of the PPI generators

Projective view of the membrane .... «fascinating!»
Budding

Refinements of the membrane generators
Space and motion
Diffusion is a consequence of the «diffuse» generator
Space and motion
Conclusion

Communication

Mobility

Distributed computation

Concurrency

Distributed systems

Space and motion

(Discrete) Complex systems

Combined

Describes

Pi-Calculus

Bigraphs

CCS
Conclusion

Higher, Deeper, broader
continuum
recursion
Stronger foundations
combination
morphisms
Abstract machine

Communication
Mobility
Distributed systems
(Discrete) Complex systems
Distributed computation
Concurrency
Space and motion

Pi-Calculus
CCS
describes

Bigraphs
Combine
Parameterized

(recursive) morphisms
The bigraph model is not canonical — variants and alternatives can be imagined — but it has at least enough power and flexibility to serve as a case study for a theory to underpin future systems engineering.
There is a lot of work to do, and we are hiring!

The jobshop

Thanks!