

# **The Space and Motion of Large Informatic Systems**

**Visions of Computer Science, 2008**

**Robin Milner, University of Cambridge**

## PARTS OF THE TALK

- What are **Informatic Models**? How do they fit together?
- **Ubiquitous Computing**, and modelling it
- **Space** and **Motion** in large systems
- **Conclusion**

## PARTS OF THE TALK

- What are **Informatic Models**? How do they fit together?
- Ubiquitous Computing, and modelling it
- Space and Motion in large systems
- Conclusion

## An informatic model

Entities in a model explain, or are realised by, entities in the physical world—as in natural science.

ENTITIES

**PROGRAMS**

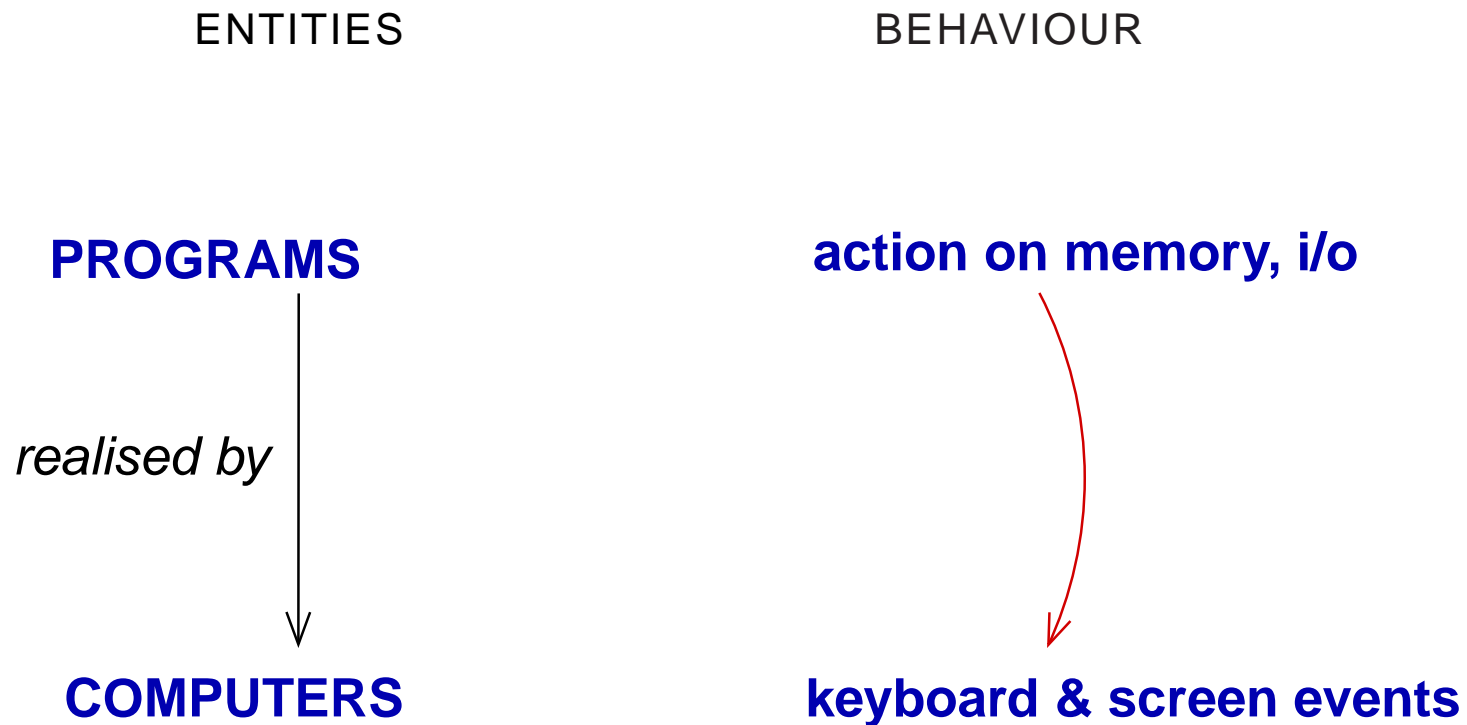
*realised by*



**COMPUTERS**

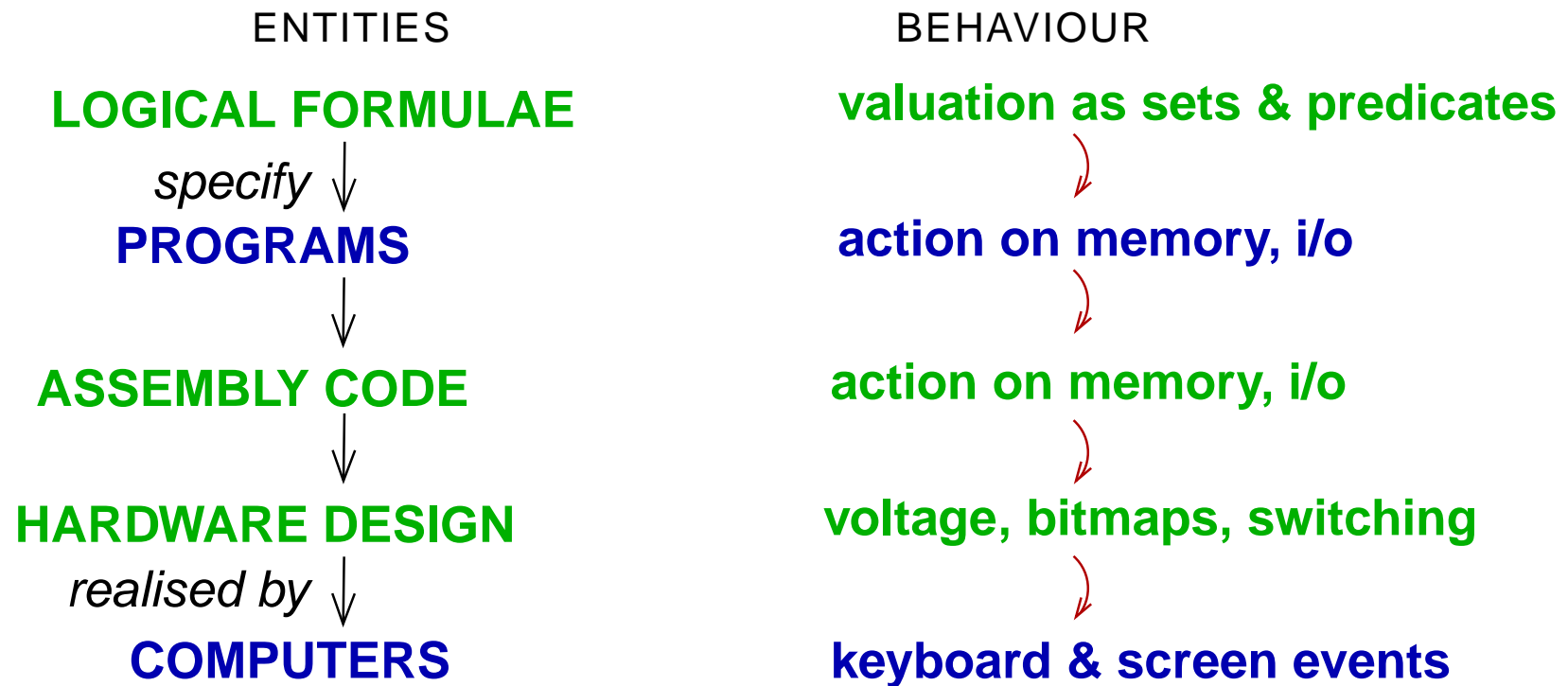
## An informatic model with behaviour

Entities *and behaviour* in a model explain, or are realised by, entities in the physical world—as in natural science.



## Layered informatic models with behaviour

Entities *and behaviour* in a model explain, or are realised by, entities in the physical world *or in a lower model*.

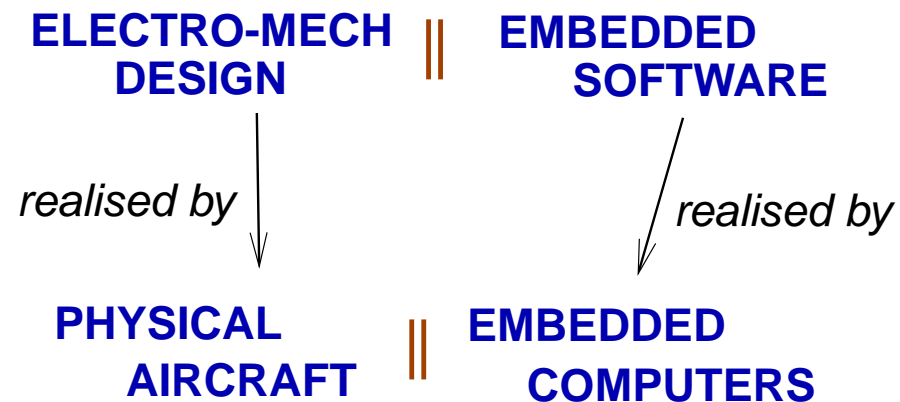


## Combining models

Real systems combine interacting **sub-systems**; we must also combine **partial models**. Thus, combine models of the electro-mechanical and informatic parts of an aircraft:

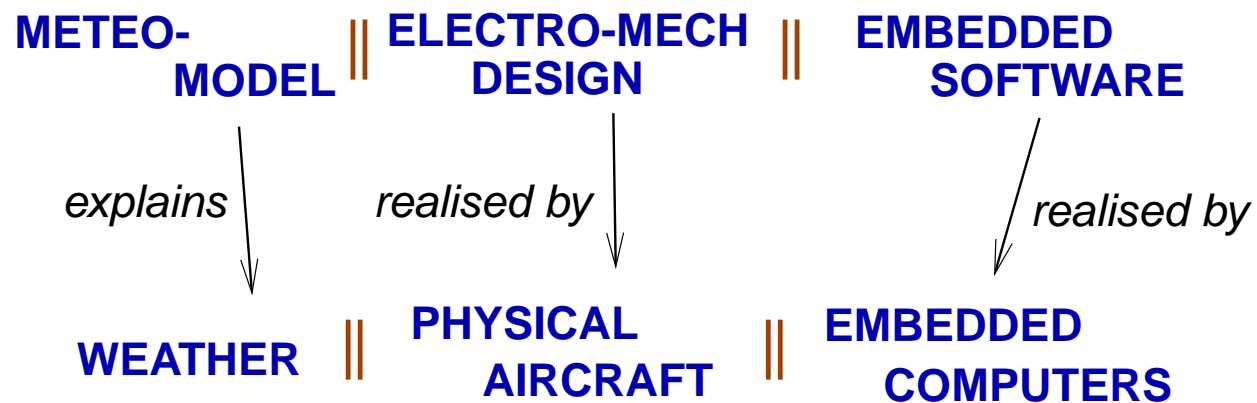
## Combining models

Real systems combine interacting **sub-systems**; we must also combine **partial models**. Thus, combine models of the electro-mechanical and informatic parts of an aircraft:



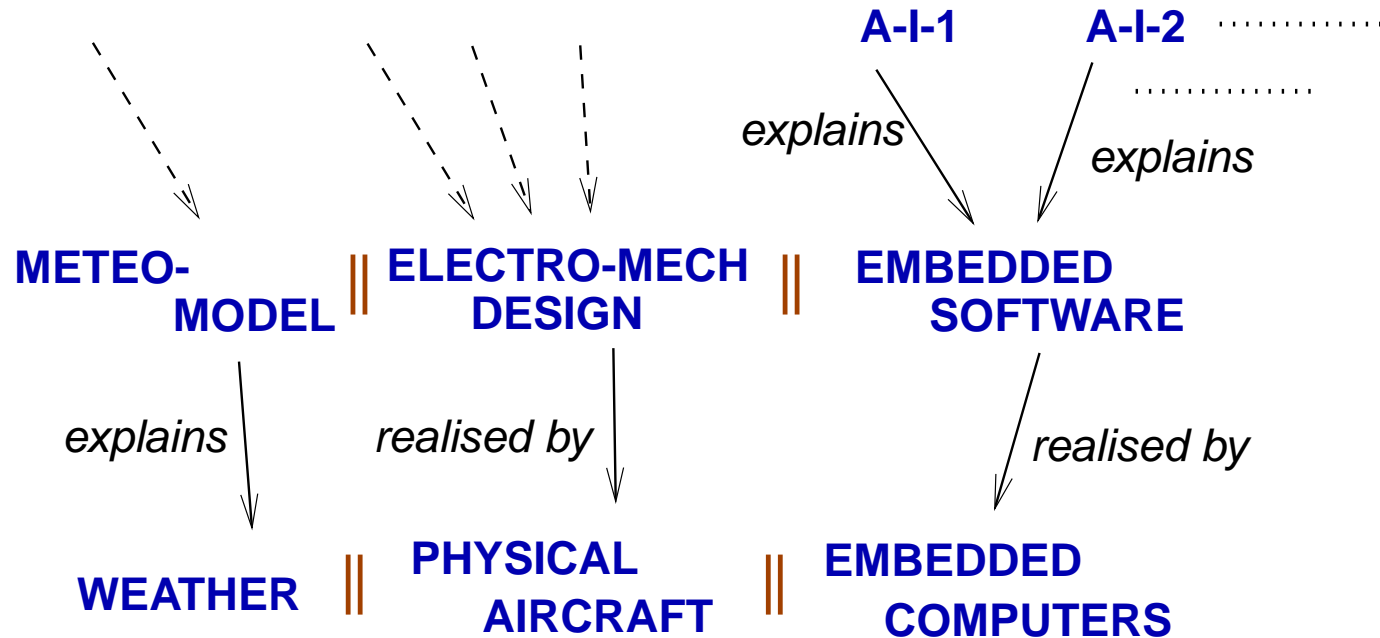
## Combining models

Real systems combine interacting **sub-systems**; we must also combine **partial models**. Also, combine models of artificial and natural systems:



# Combining models

For a program, we may combine different explanatory models. INRIA did this for the **Airbus** using **abstract interpretation**, following successful analysis of the failure of the Ariane-5 rocket:



## Models and their tower

A **model** consists of some *entities*, and their *behaviour*.

EXAMPLE: flowcharts, and how to execute them.

## Models and their tower

A **model** consists of some *entities*, and their *behaviour*.

**EXAMPLE:** flowcharts, and how to execute them.

A **tower** of models is built by *explanation* and *combination* :

Model A **explains** model B if

*A abstracts from or specifies B, or if  
B implements or realises A.*

**EXAMPLE:** a specification logic specifies programs.

## Models and their tower

A **model** consists of some *entities*, and their *behaviour*.

**EXAMPLE:** flowcharts, and how to execute them.

A **tower** of models is built by *explanation* and *combination* :

Model A **explains** model B if

*A abstracts from or specifies B, or if  
B implements or realises A.*

**EXAMPLE:** a specification logic specifies programs.

Model C **combines** models A and B if

*its entities and behaviours combine those of A and B.*

**EXAMPLE:** combine distributed programs with a network model.

## How do we validate an explanation?

### Natural science:

Explanation of reality by a model can only be supported by *observation*. Complete validation impossible (Karl Popper).

## How do we validate an explanation?

### Natural science:

Explanation of reality by a model can only be supported by *observation*. Complete validation impossible (Karl Popper).

### Informatics at lowest level:

Similar (e.g. realisation of circuit diagrams by a computer).

### Informatics at higher levels:

Higher levels abound in the model tower. Can aspire to *complete validation* between precise models.

## How do we validate an explanation?

### Natural science:

Explanation of reality by a model can only be supported by *observation*. Complete validation impossible (Karl Popper).

### Informatics at lowest level:

Similar (e.g. realisation of circuit diagrams by a computer).

### Informatics at higher levels:

Higher levels abound in the model tower. Can aspire to *complete validation* between precise models.

**PROPOSITION: Informatics is an science just to the extent that it aspires to complete validation.**

## Scientific status of the **Tower of Models**

- Useful models, and validations, may well be **informal**
- **Different models** suit different people, including **non-experts**

## Scientific status of the **Tower of Models**

- Useful models, and validations, may well be **informal**
- **Different models** suit different people, including **non-experts**
- **Many instances** of models and validations exist

## Scientific status of the **Tower of Models**

- Useful models, and validations, may well be **informal**
- **Different models** suit different people, including **non-experts**
- **Many instances** of models and validations exist
- Can we derive **languages from models**, not vice-versa?

## PARTS OF THE TALK

- What are Informatic Models? How do they fit together?
- **Ubiquitous Computing**, and modelling it
- Space and Motion in large systems
- Conclusion

# Two visions of Ubiquitous Computing

## Two visions of Ubiquitous Computing

Populations of computing entities will be a significant part of our environment, performing tasks that support us, and we shall be largely unaware of them. (after Mark Weiser, 1994)

## Two visions of Ubiquitous Computing

Populations of computing entities will be a significant part of our environment, performing tasks that support us, and we shall be largely unaware of them. (after Mark Weiser, 1994)

In the next five to ten years the computer will be erased from our consciousness. We will simply not talk about it any longer, we will not read about it, *apart from experts of course.*

*(my emphasis)*

Joseph Weizenbaum (2001)

## Two visions of Ubiquitous Computing

Populations of computing entities will be a significant part of our environment, performing tasks that support us, and we shall be largely unaware of them. (after Mark Weiser, 1994)

In the next five to ten years the computer will be erased from our consciousness. We will simply not talk about it any longer, we will not read about it, *apart from experts of course.*

*(my emphasis)*

Joseph Weizenbaum (2001)

..... and my vision:

Ubiquitous computing will **empower us**, if we **understand it**.

# **Qualities of a ubiquitous computing system (UCS)**

What is new about a UCS?

# Qualities of a ubiquitous computing system (UCS)

What is new about a UCS?

- It will continually make **decisions** hitherto made by us
- It will be **vast**, maybe 100 times today's systems

# Qualities of a ubiquitous computing system (UCS)

What is new about a UCS?

- It will continually make **decisions** hitherto made by us
- It will be **vast**, maybe 100 times today's systems
- It must continually **adapt**, on-line, to new requirements
- Individual UCSs will **interact** with one another

# Qualities of a ubiquitous computing system (UCS)

What is new about a UCS?

- It will continually make **decisions** hitherto made by us
- It will be **vast**, maybe 100 times today's systems
- It must continually **adapt**, on-line, to new requirements
- Individual UCSs will **interact** with one another

**Can traditional software engineering cope?**

## Concepts for Ubicomp

Each ubicomp **domain**, hence each **model**, will involve several concepts.

# Concepts for Ubicomp

Each ubicomp **domain**, hence each **model**, will involve several concepts. Here are a few:

provenance obligations self-management  
locality intentions specification data-protection  
beliefs continuous space authorisation simulation  
encapsulation mobility continuous time role  
compilation policy failure  
delegation reflectivity verification  
stochastics negotiation connectivity  
trust security authenticity

# Managing the conceptual overload



## Managing the conceptual overload

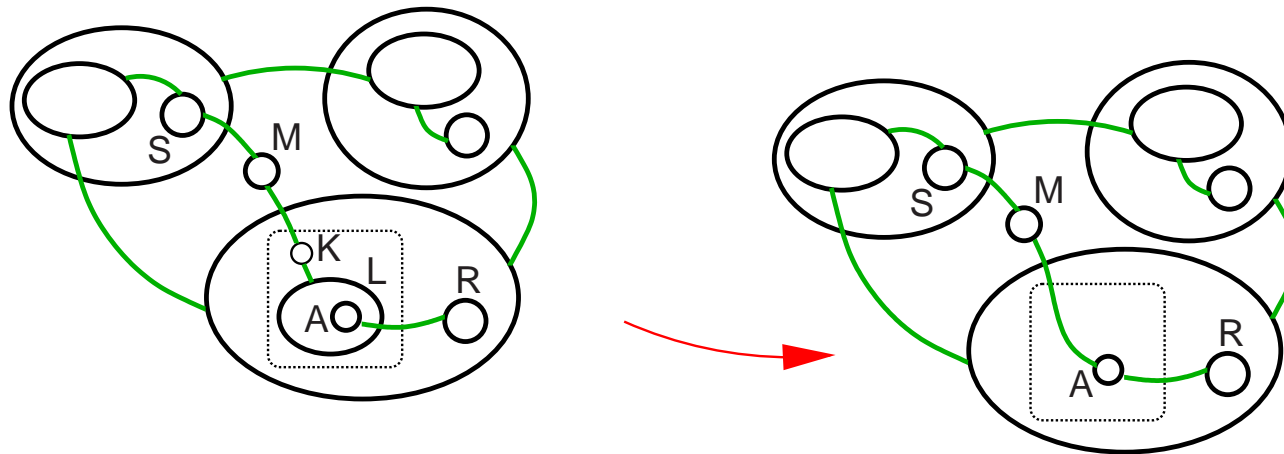


- Define **UAM**, the *Ubiquitous Abstract Machine*, in terms of locality, connectivity, mobility, stochastics.
- Build a *model tower* above **UAM**, layering the concepts.

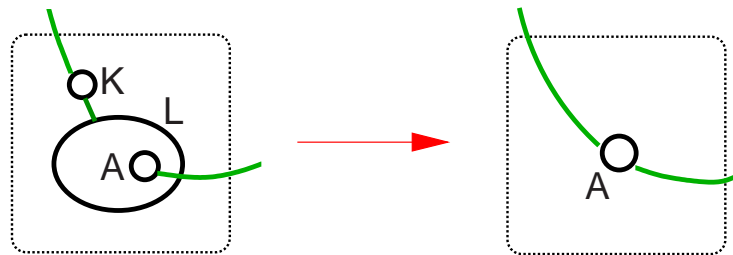
## PARTS OF THE TALK

- What are Informatic Models? How do they fit together?
- Ubiquitous Computing, and modelling it
- **Space and Motion** in large systems
- Conclusion

## A fanciful system, seen as a **bigraph**

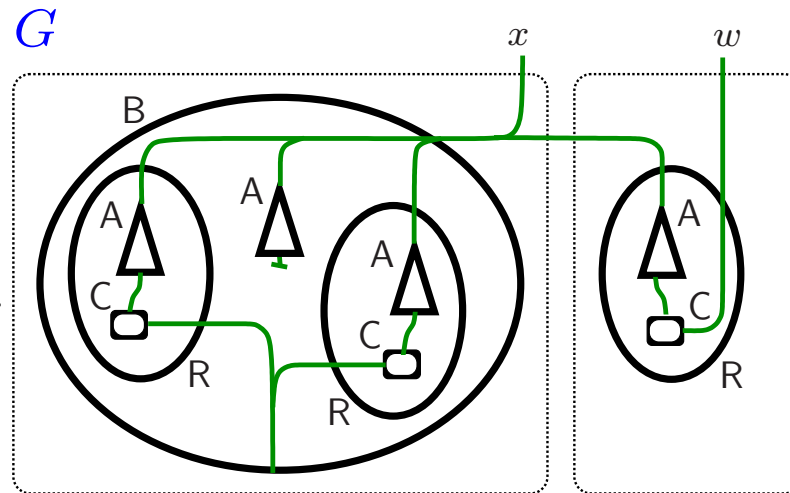


Reaction rule:



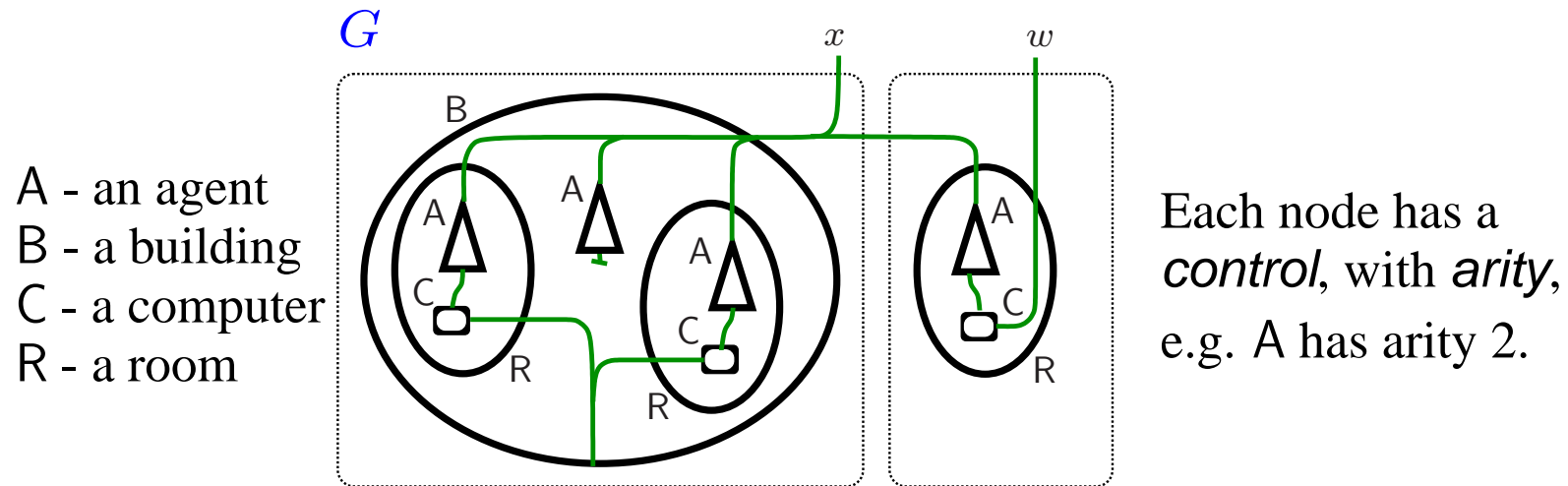
## A built environment $G$

A - an agent  
B - a building  
C - a computer  
R - a room



Each node has a *control*, with *arity*, e.g. A has arity 2.

## A built environment $G$

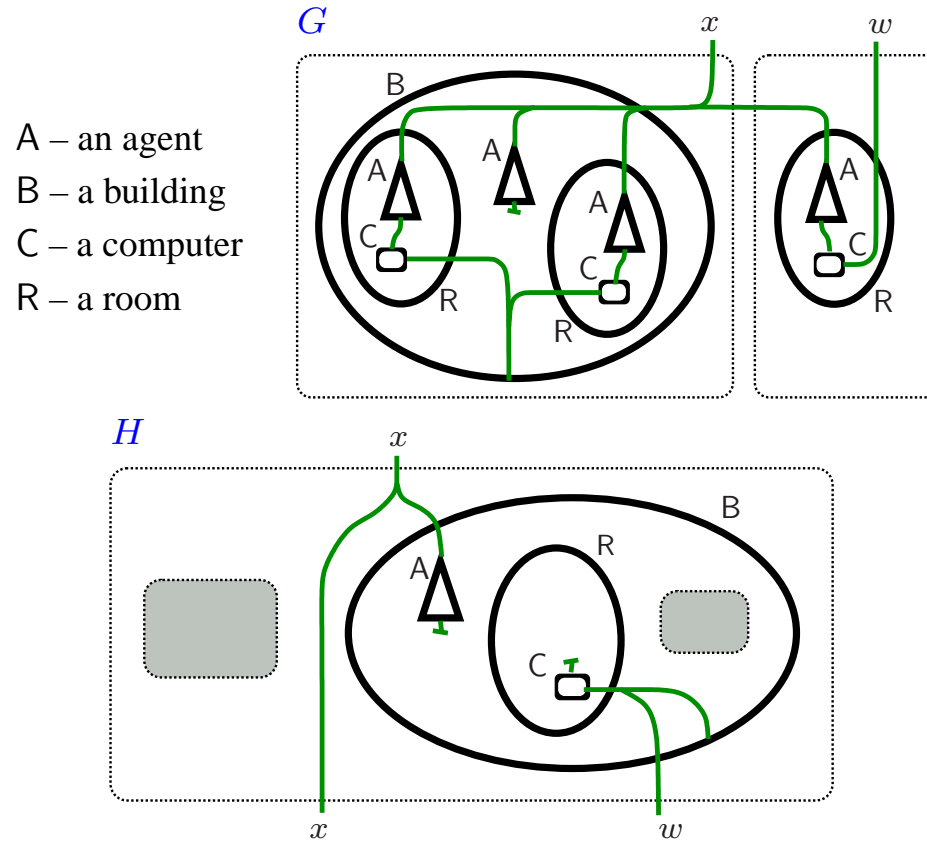


$$G = /z B_z.(\text{Roomfull}_{xz} \mid /y A_{xy} \mid \text{Roomfull}_{xz}) \parallel \text{Roomfull}_{xw}$$

where  $\text{Roomfull}_{xz} \stackrel{\text{def}}{=} R.y (A_{xy} \mid C_{yz})$ .

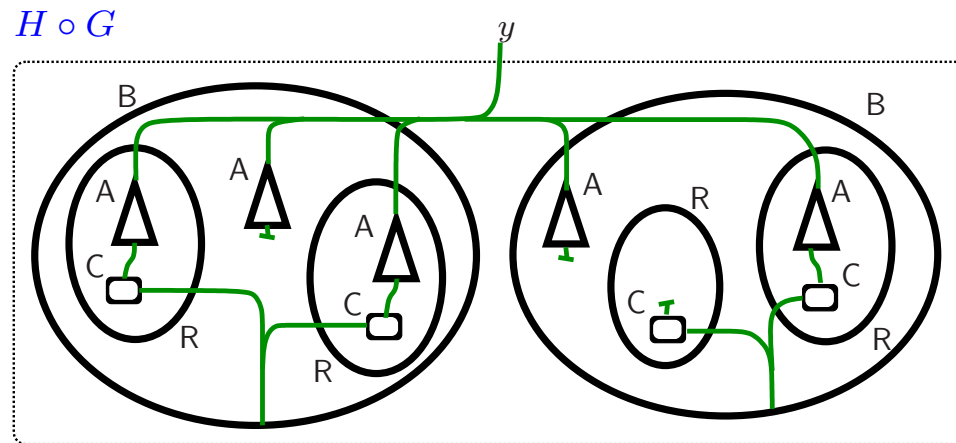
The *signature*  $\mathcal{K} = \{A : 2, B : 1 \dots\}$  gives controls with arities.

..... and a host  $H$  for  $G$



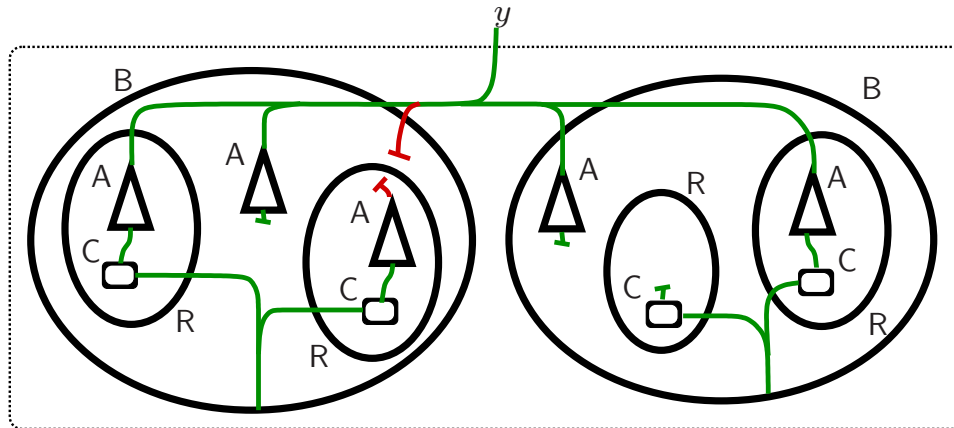
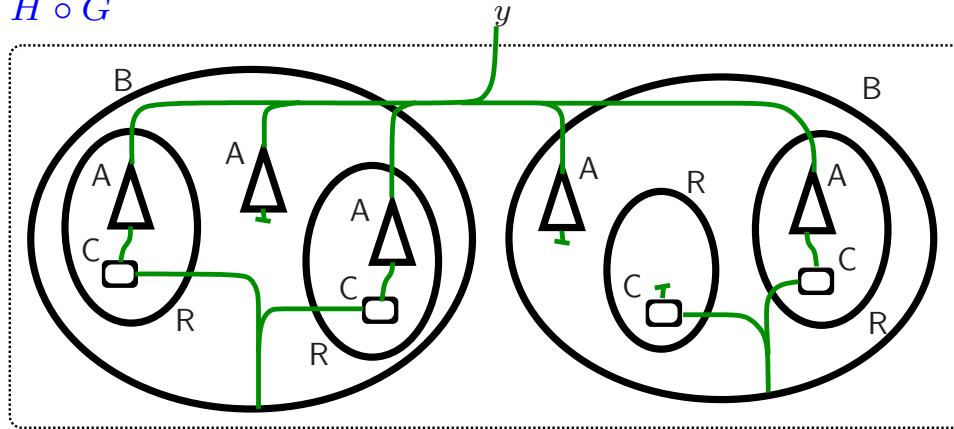
$$H = id_1 | id_x | /w B_w . (/y A_{xy} | R . /y C_{yw} | id_w | id_1) .$$

# The complete system $H \circ G$



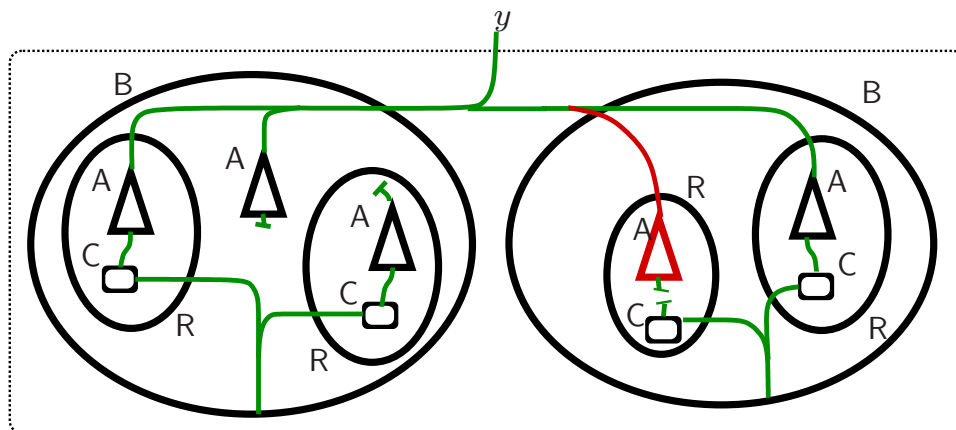
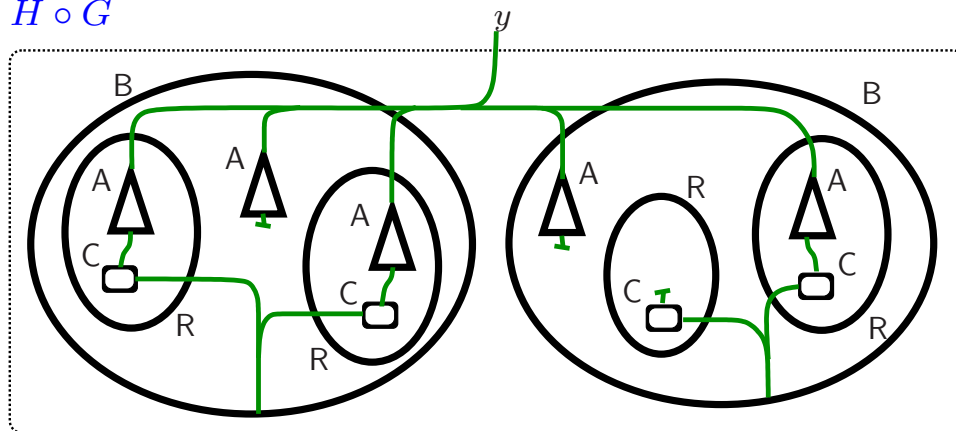
..... and after **one** reaction

$H \circ G$



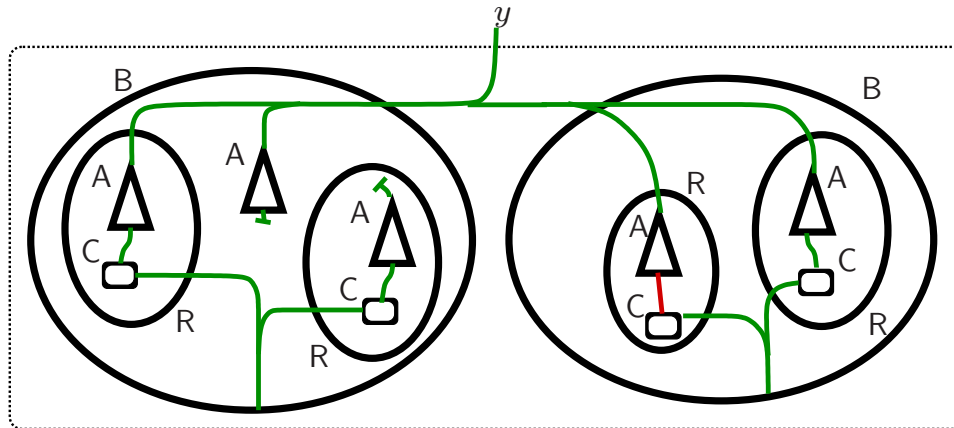
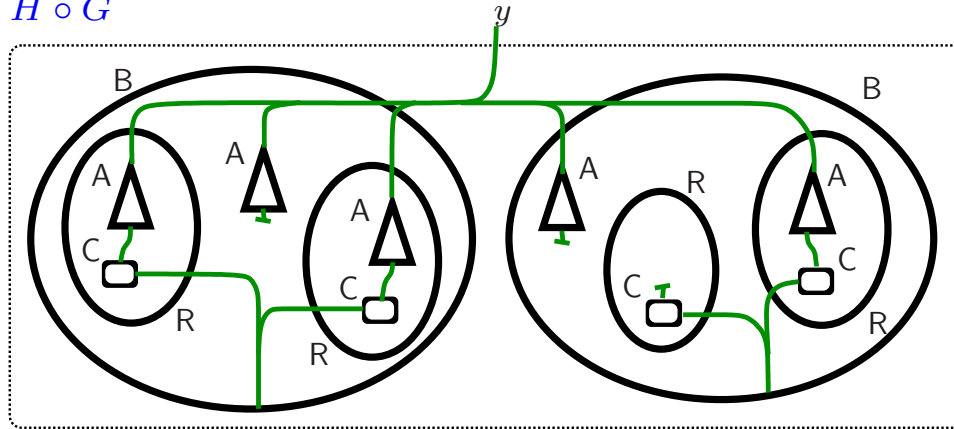
.....and after **two** reactions

$H \circ G$

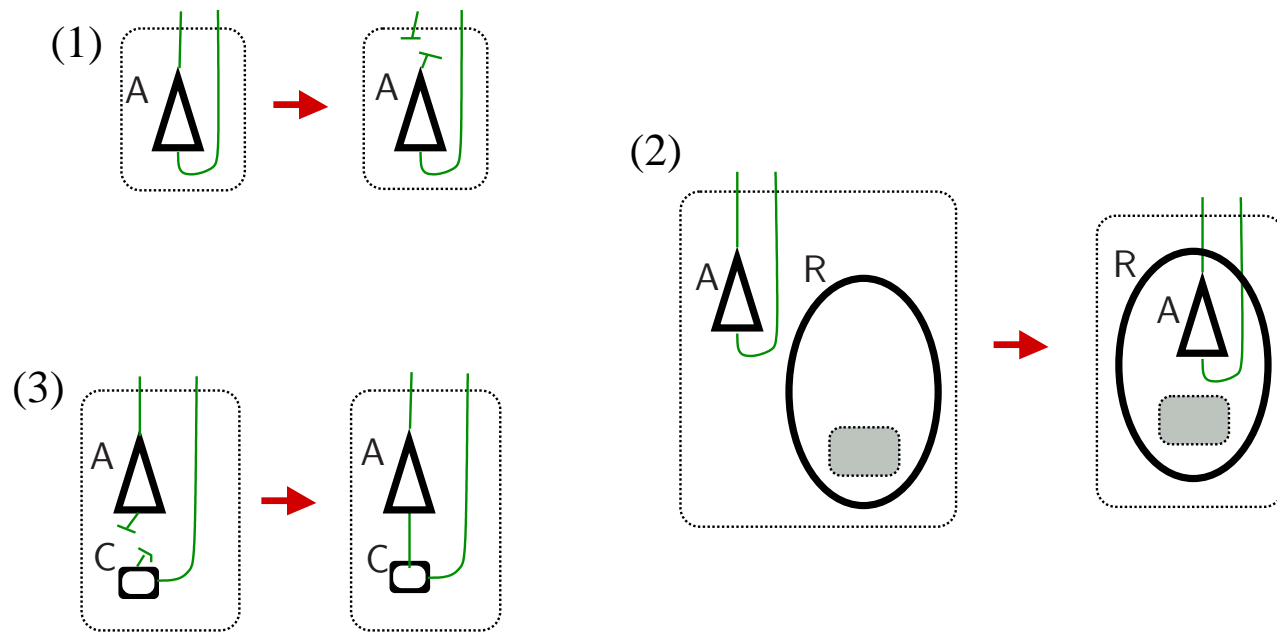


.....and after **three** reactions

$H \circ G$

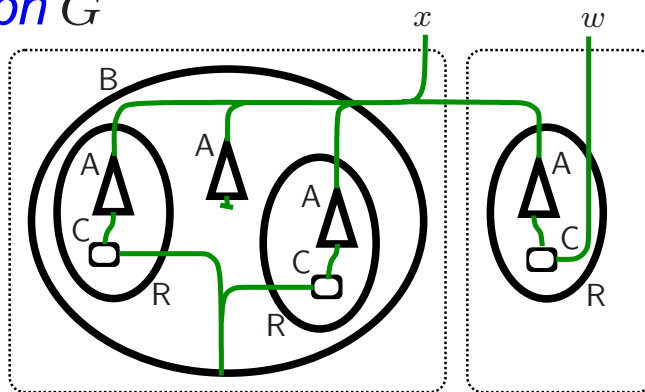


## Three possible reaction rules

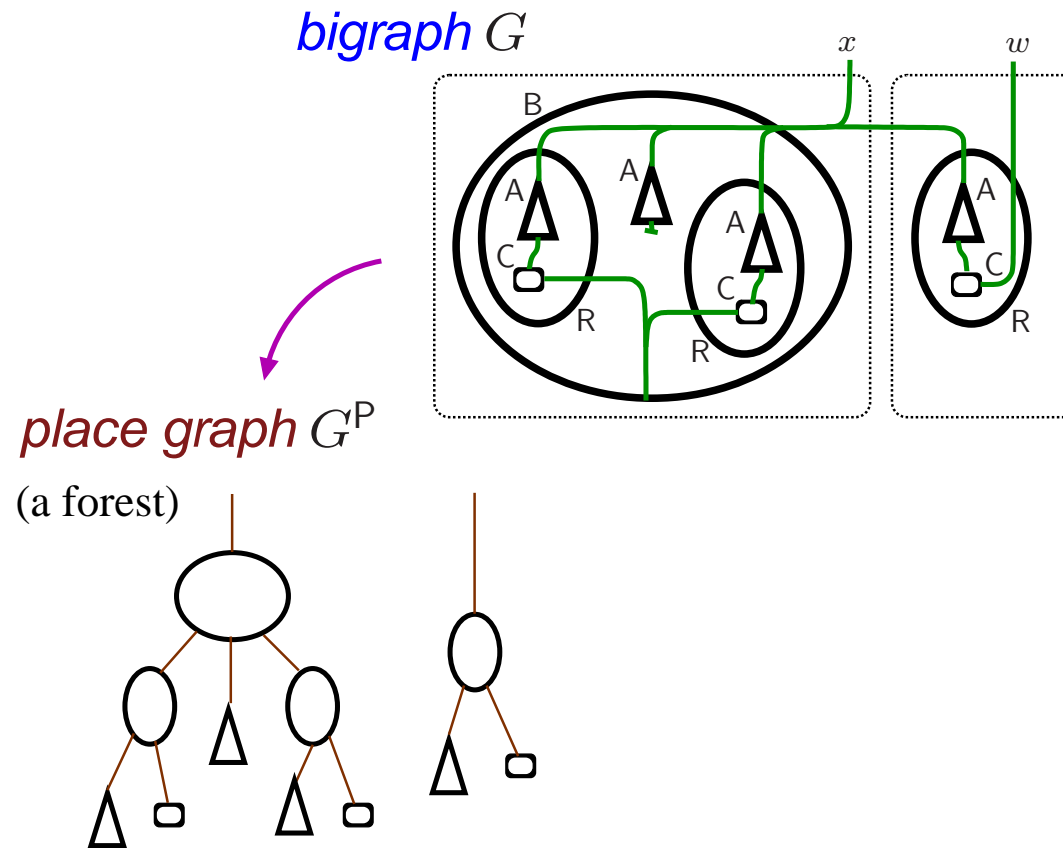


# The 'bi-' structure of a bigraph

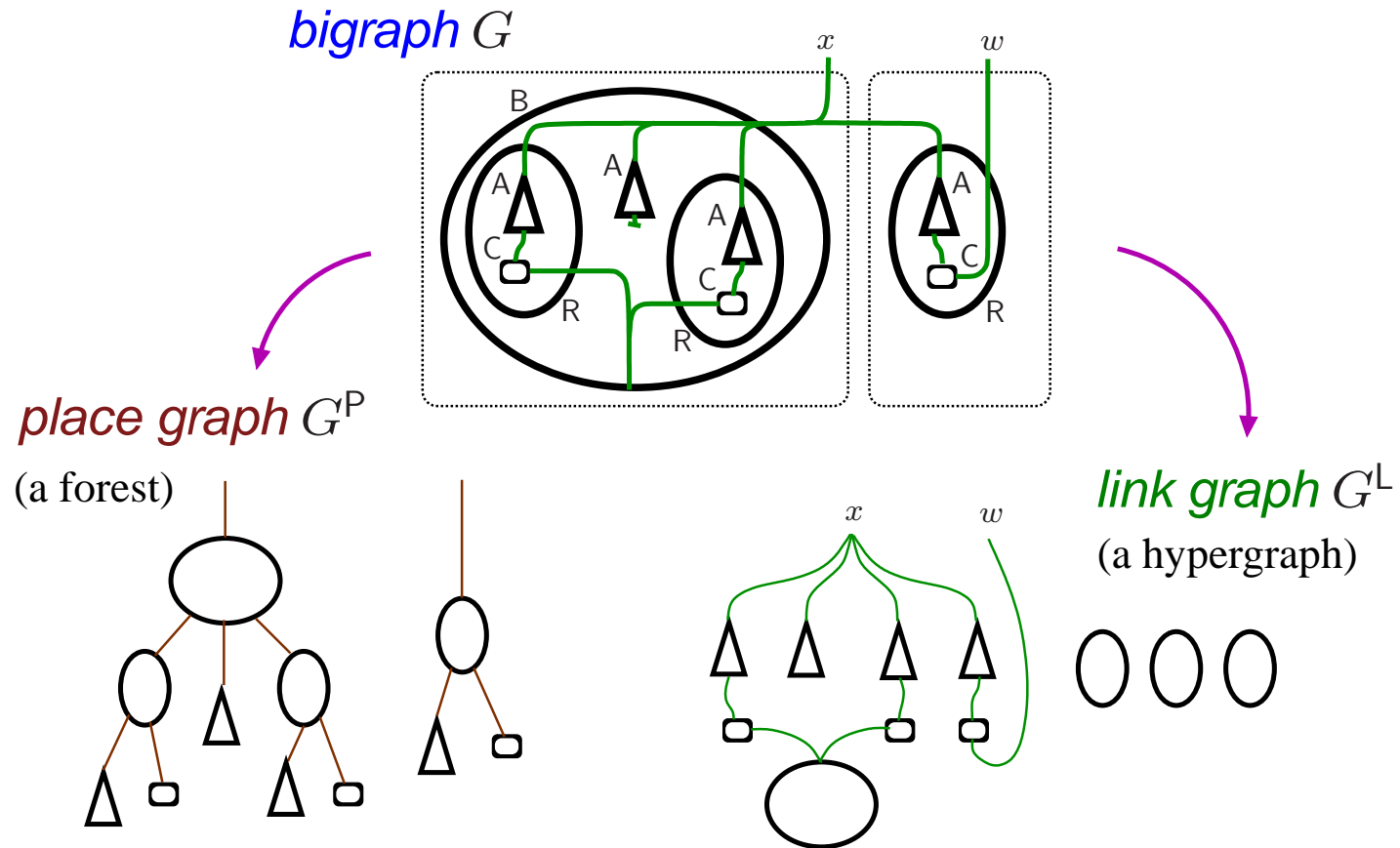
*bigraph*  $G$



# The 'bi-' structure of a bigraph



# The 'bi-' structure of a bigraph



## The variety of bigraphical models

- A *bigraphical reactive system (BRS)*  $\mathbf{BG}(\Sigma, \mathcal{R})$  is defined by a *sorting*  $\Sigma$  and a *reaction regime*  $\mathcal{R}$  (reaction rules). .

## The variety of bigraphical models

- A *bigraphical reactive system (BRS)*  $\text{BG}(\Sigma, \mathcal{R})$  is defined by a *sorting*  $\Sigma$  and a *reaction regime*  $\mathcal{R}$  (reaction rules). .
- *Process calculi* (CCS, CSP,  $\pi$ -calculus, Petri nets, Mobile Ambients) are represented faithfully by BRSs.
- *Transition systems* and *behavioural theory* (e.g. bisimilarity) for these calculi are derived uniformly from reaction regimes.

## The variety of bigraphical models

- A *bigraphical reactive system (BRS)*  $\text{BG}(\Sigma, \mathcal{R})$  is defined by a *sorting*  $\Sigma$  and a *reaction regime*  $\mathcal{R}$  (reaction rules). .
- *Process calculi* (CCS, CSP,  $\pi$ -calculus, Petri nets, Mobile Ambients) are represented faithfully by BRSs.
- *Transition systems* and *behavioural theory* (e.g. bisimilarity) for these calculi are derived uniformly from reaction regimes.

We now outline the maths of bigraphs.

Then we sketch BRSs for a **reflective building**, a **process calculus**, and a **biological phenomenon**.

# The mathematics of bigraphs

- Each BRS is based on a *symmetric partial monoidal (spm) category*, plus dynamics.
- The *static algebra* of BRSs is *completely axiomatised*.

# The mathematics of bigraphs

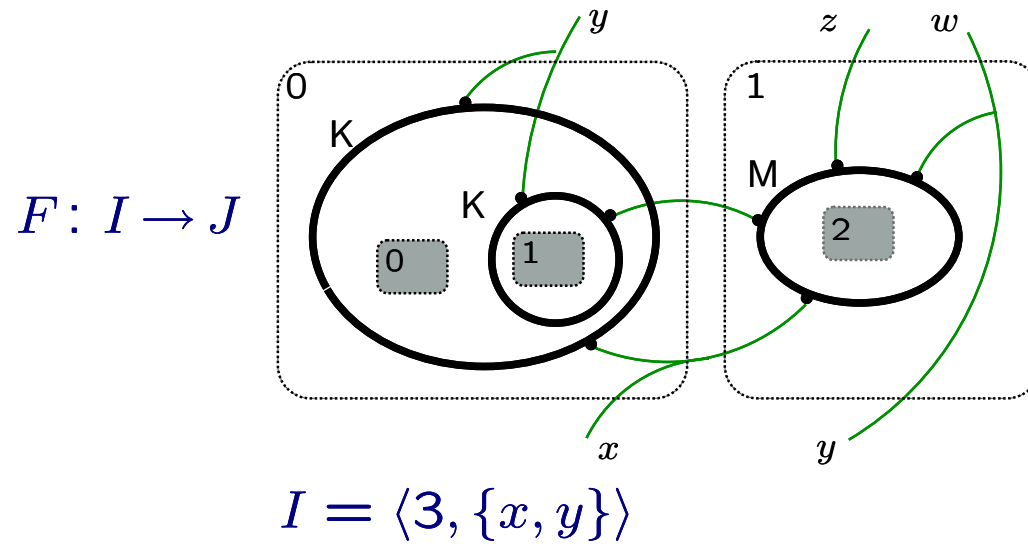
- Each BRS is based on a *symmetric partial monoidal (spm) category*, plus dynamics.
- The *static algebra* of BRSs is *completely axiomatised*.
- The *dynamics* of BRSs involves graph matching, formally defined. Hence *bigraphical programming language (BPL)* under development at the ITU, Copenhagen.
- The uniform dynamical theory of BRSs is based on a categorical notion, *relative pushouts*.

# The mathematics of bigraphs

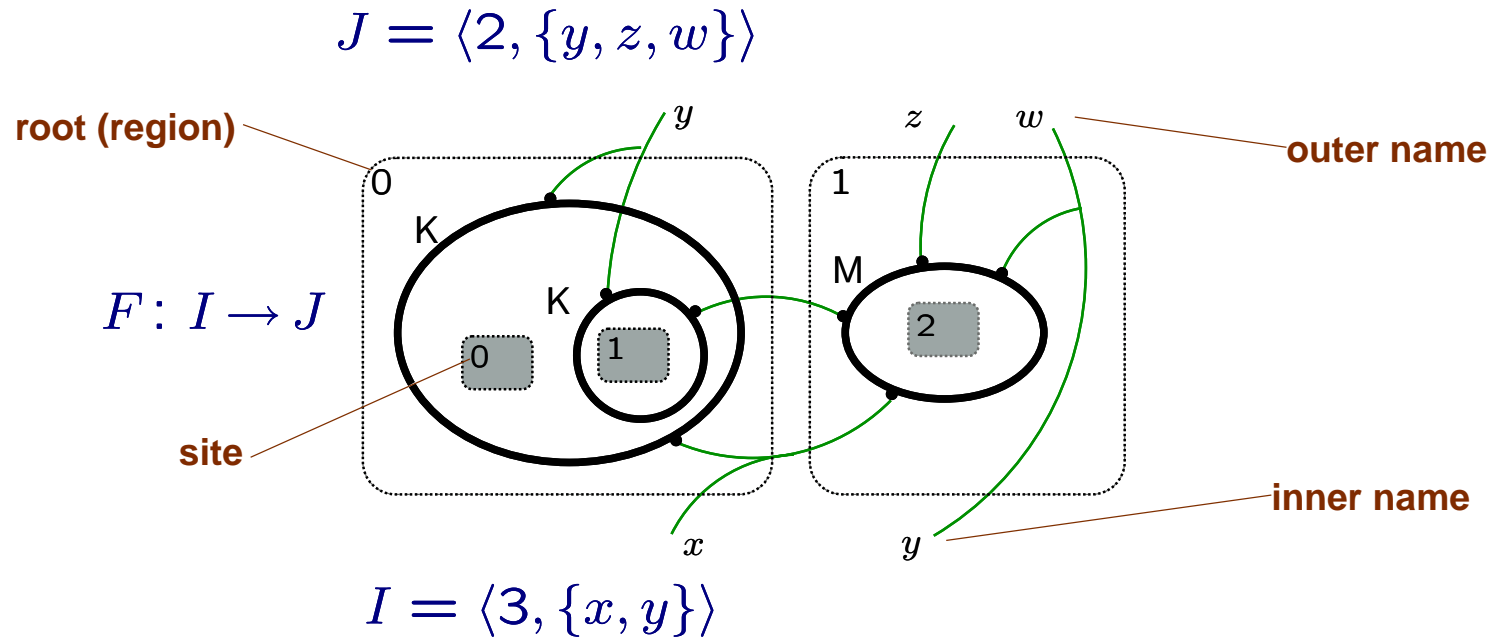
- Each BRS is based on a *symmetric partial monoidal (spm) category*, plus dynamics.
- The *static algebra* of BRSs is *completely axiomatised*.
- The *dynamics* of BRSs involves graph matching, formally defined. Hence *bigraphical programming language (BPL)* under development at the ITU, Copenhagen.
- The uniform dynamical theory of BRSs is based on a categorical notion, *relative pushouts*.
- *Stochastic behaviour* is uniformly derived.

## Bigraph algebra: their interfaces and operations

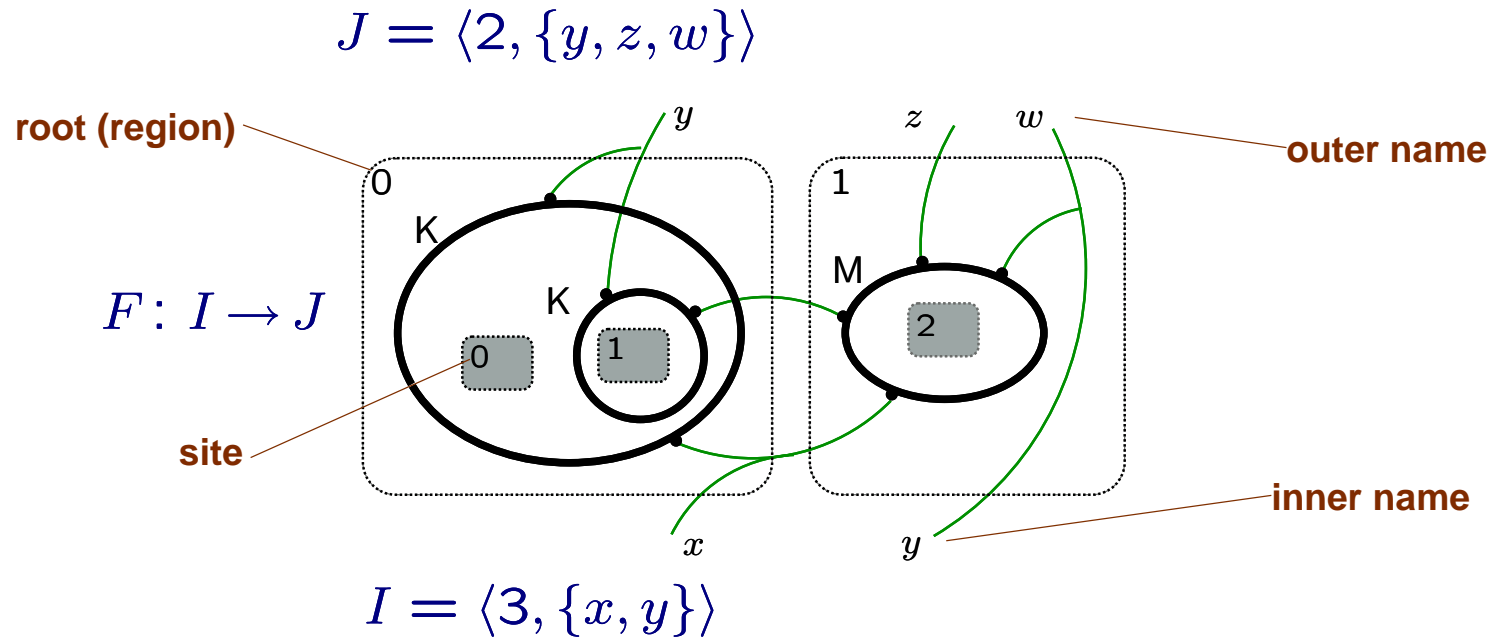
$$J = \langle 2, \{y, z, w\} \rangle$$



# Bigraph algebra: their interfaces and operations



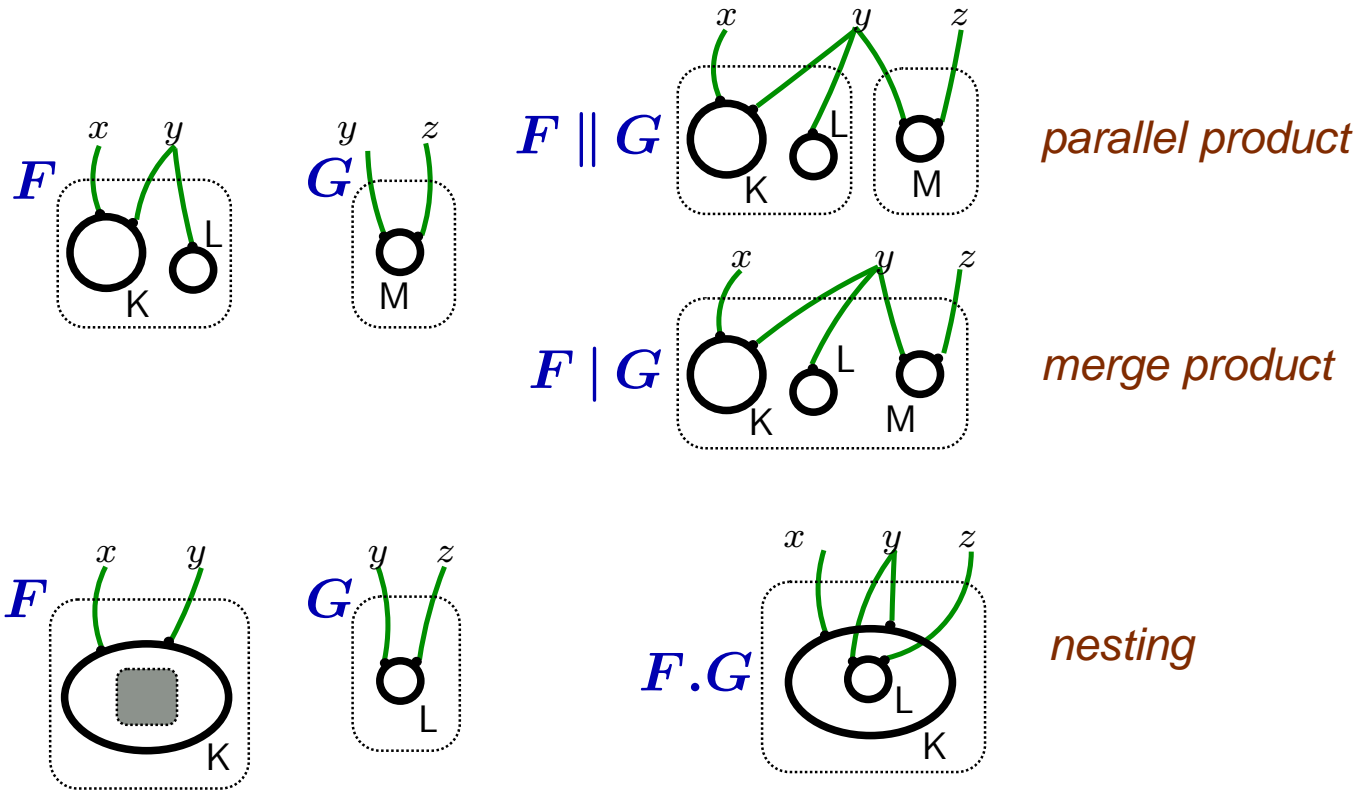
## Bigraph algebra: their interfaces and operations



**Composition:** Place  $F: I \rightarrow J$  inside  $G: J \rightarrow K$   
to yield  $G \circ F: I \rightarrow K$ .

**Product:** Place  $F: I \rightarrow J$  alongside  $G: H \rightarrow K$   
to yield  $F \otimes G: I \otimes H \rightarrow J \otimes K$ .

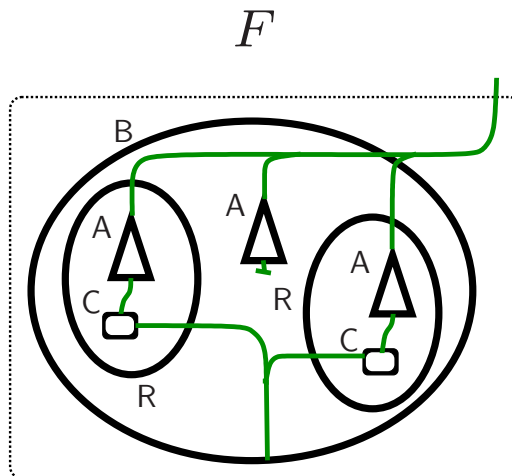
## Derived operations: product and nesting



These operations are *elementary* for process calculi.  
 Illuminating that they are *derived* in the categorical framework.

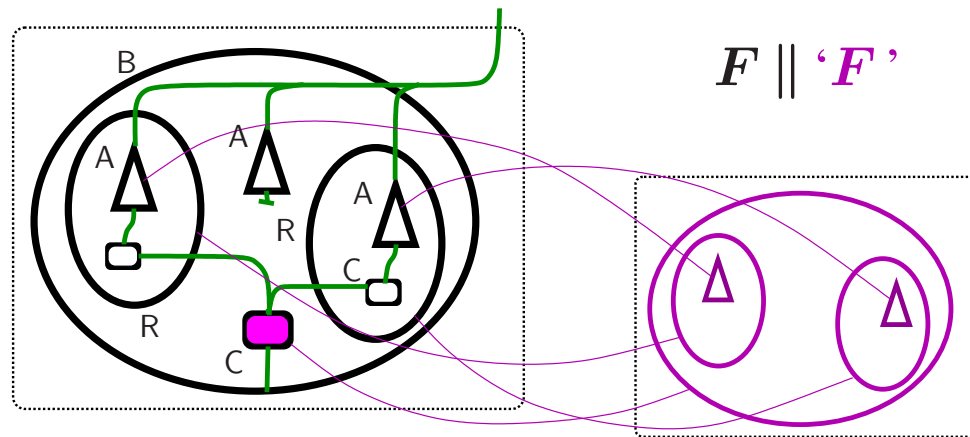
## Reflective building (0)

A building may keep a partial record of its occupancy.



# Reflective building (1)

A building may keep a partial record of its occupancy.



So it has a central computer that 'holds' the record.

The record could be any data structure, accessible to the real occupants via the building's network.

## Finite CCS

$$\text{SYNTAX} \left\{ \begin{array}{l} \mu ::= \bar{x} \mid x \quad \text{actions} \\ P ::= A \mid \nu x P \mid P \mid P \quad \text{processes} \\ A ::= \mathbf{0} \mid \mu.P \mid A + A \quad \text{alternations} \end{array} \right.$$

The BRS for CCS has controls **send**, **get** and **alt**. It has one sort for processes, one for alternations.

## Finite CCS

$$\text{SYNTAX} \left\{ \begin{array}{l} \mu ::= \bar{x} \mid x \quad \text{actions} \\ P ::= A \mid \nu x P \mid P \mid P \quad \text{processes} \\ A ::= \mathbf{0} \mid \mu.P \mid A + A \quad \text{alternations} \end{array} \right.$$

The BRS for CCS has controls **send**, **get** and **alt**. It has one sort for processes, one for alternations.

Maps  $\mathcal{P}_X[\cdot]$  and  $\mathcal{A}_X[\cdot]$  translate CCS entities with names  $\subseteq X$  to bigraphs of the right sort:

$$\begin{array}{l} \mathcal{P}_X[\nu x P] = /x \mathcal{P}_{x \uplus X}[P] \\ \mathcal{P}_X[P \mid Q] = \mathcal{P}_X[P] \mid \mathcal{P}_X[Q] \\ \mathcal{P}_X[A] = \text{alt. } \mathcal{A}_X[A] . \end{array} \left| \begin{array}{l} \mathcal{A}_X[\mathbf{0}] = X \mid 1 \\ \mathcal{A}_X[\bar{x}.P] = \text{send}_x . \mathcal{P}_X[P] \\ \mathcal{A}_X[x.P] = \text{get}_x . \mathcal{P}_X[P] \\ \mathcal{A}_X[A + B] = \mathcal{A}_X[A] \mid \mathcal{A}_X[B] . \end{array} \right.$$

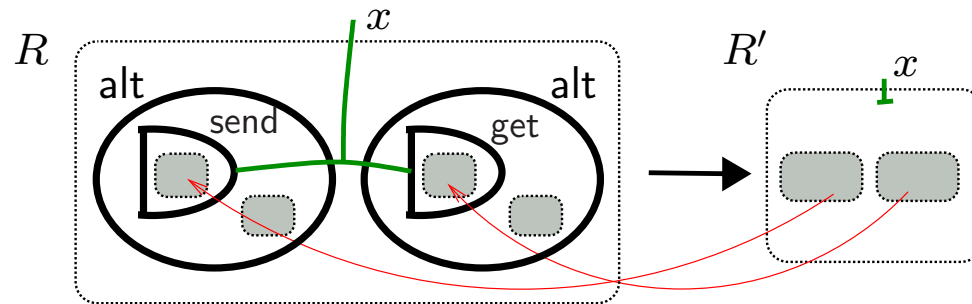
## Reaction in CCS bigraphs

Reaction in CCS:  $(\bar{x}.P_1 + A_1) | (x.P_2 + A_2) \longrightarrow P | Q$

## Reaction in CCS bigraphs

Reaction in CCS:  $(\bar{x}.P_1 + A_1) | (x.P_2 + A_2) \longrightarrow P | Q$

This is encoded in bigraphs by the rule:

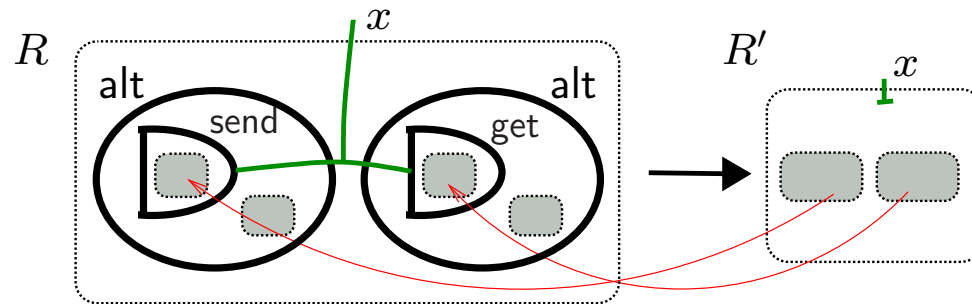


The red arrows show which parameters are retained. The rule generates a reaction relation  $\longrightarrow$  between CCS bigraphs.

## Reaction in CCS bigraphs

Reaction in CCS:  $(\bar{x}.P_1 + A_1) | (x.P_2 + A_2) \longrightarrow P | Q$

This is encoded in bigraphs by the rule:



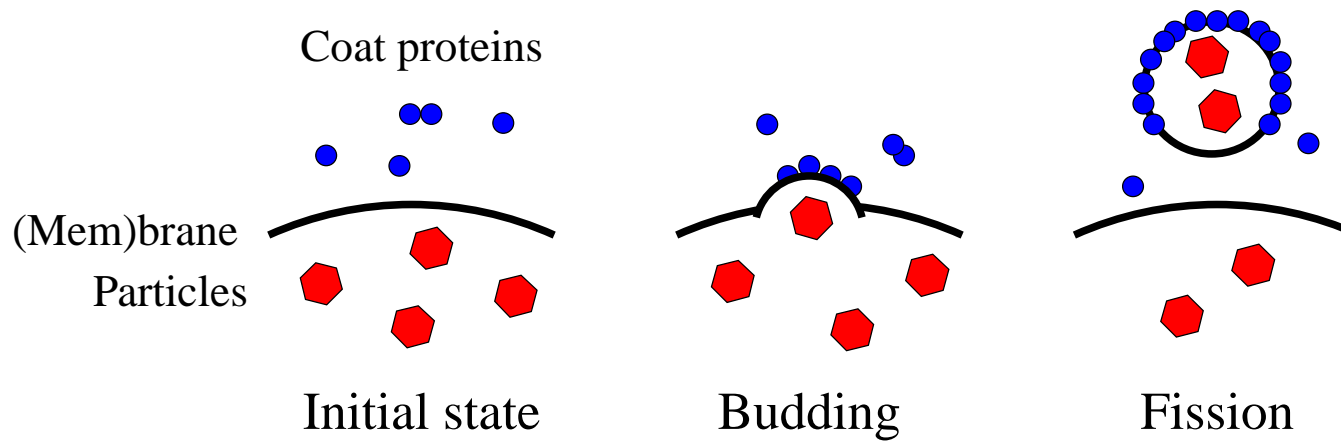
The red arrows show which parameters are retained. The rule generates a reaction relation  $\longrightarrow$  between CCS bigraphs.

**THEOREM** The bigraph model *explains* CCS:  
 $P \longrightarrow P'$  in CCS iff  $\mathcal{P}_X[P] \longrightarrow \mathcal{P}_X[P']$  in bigraphs.

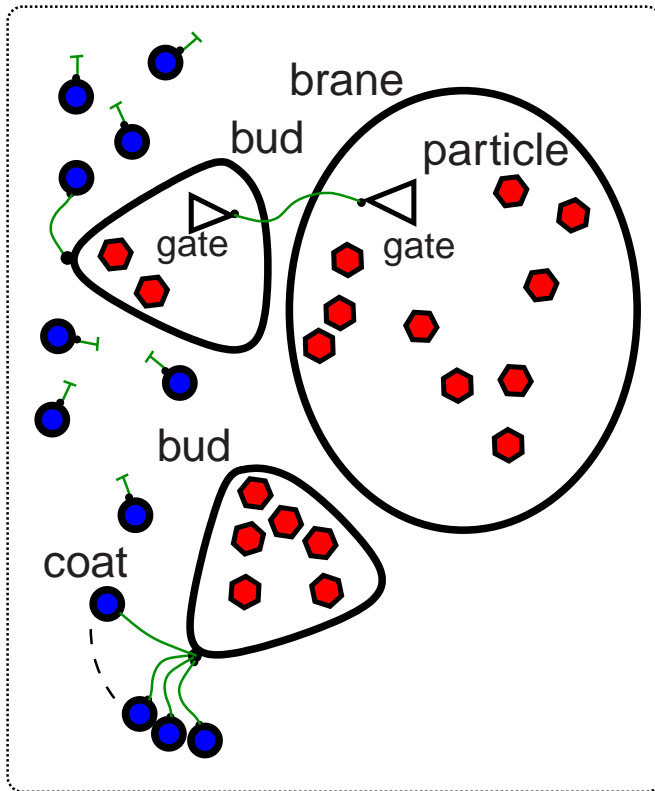
# Stochastic dynamics

*joint work with Jean Krivine and Angelo Troina*

For example, **membrane budding**:



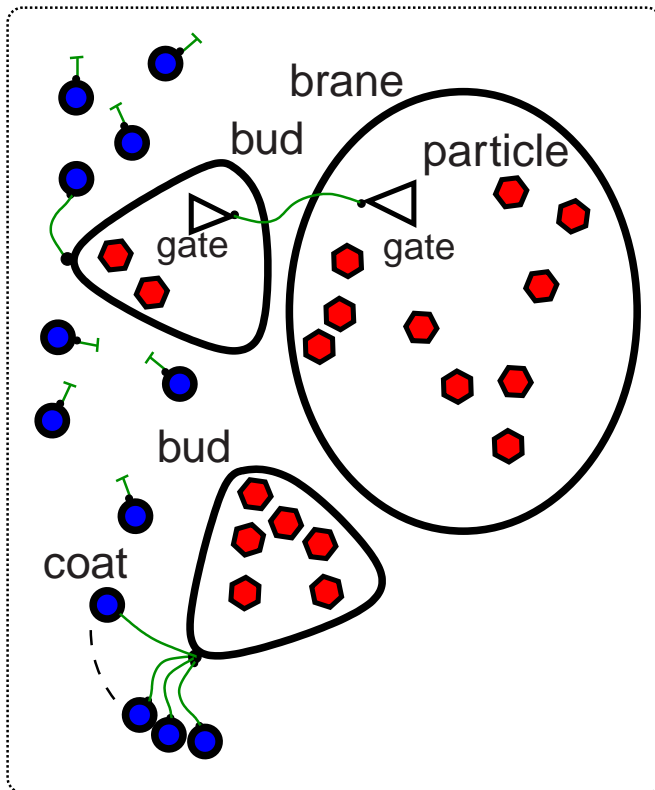
## A membrane-bud system



*The controls are:*

brane, bud, coat, particle, gate

## A membrane-bud system



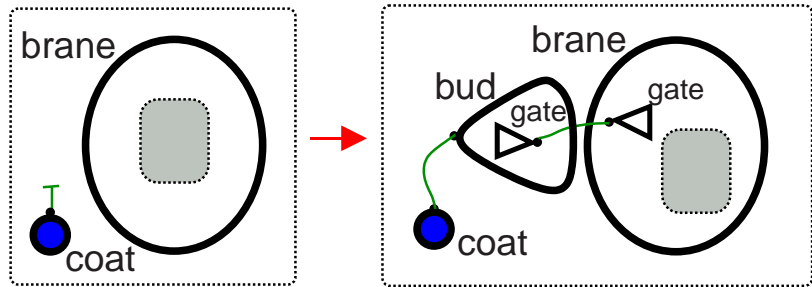
*The controls are:*

brane, bud, coat, particle, gate

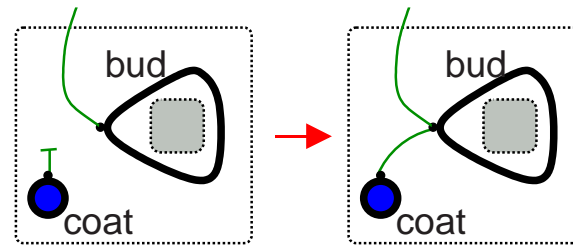
*The **sorting** dictates:*

- a particle, coat protein or gate has no children
- children of a bud or brane are particles or gates

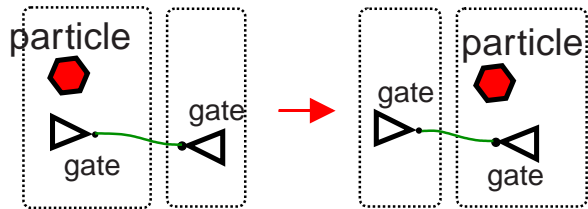
## Reaction rules for budding, with stochastic rates



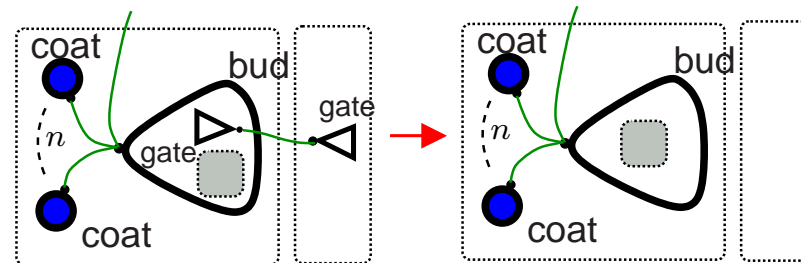
bud formation



coating



particle migration



bud fission

## Stochastics: the rates of reactions

Assign a **rate**  $\rho_i$  to each reaction rule  $R_i \rightarrow R'_i$

## Stochastics: the rates of reactions

Assign a **rate**  $\rho_i$  to each reaction rule  $R_i \rightarrow R'_i$

The rate of a particular **reaction**  $g \rightarrow g'$  is given by

$$\sum_i \rho_i \cdot n_i$$

where  $n_i$  is the number of different ways that the  $i^{\text{th}}$  rule can give rise to the reaction  $g \rightarrow g'$ .

## Stochastics: the rates of reactions

Assign a **rate**  $\rho_i$  to each reaction rule  $R_i \rightarrow R'_i$

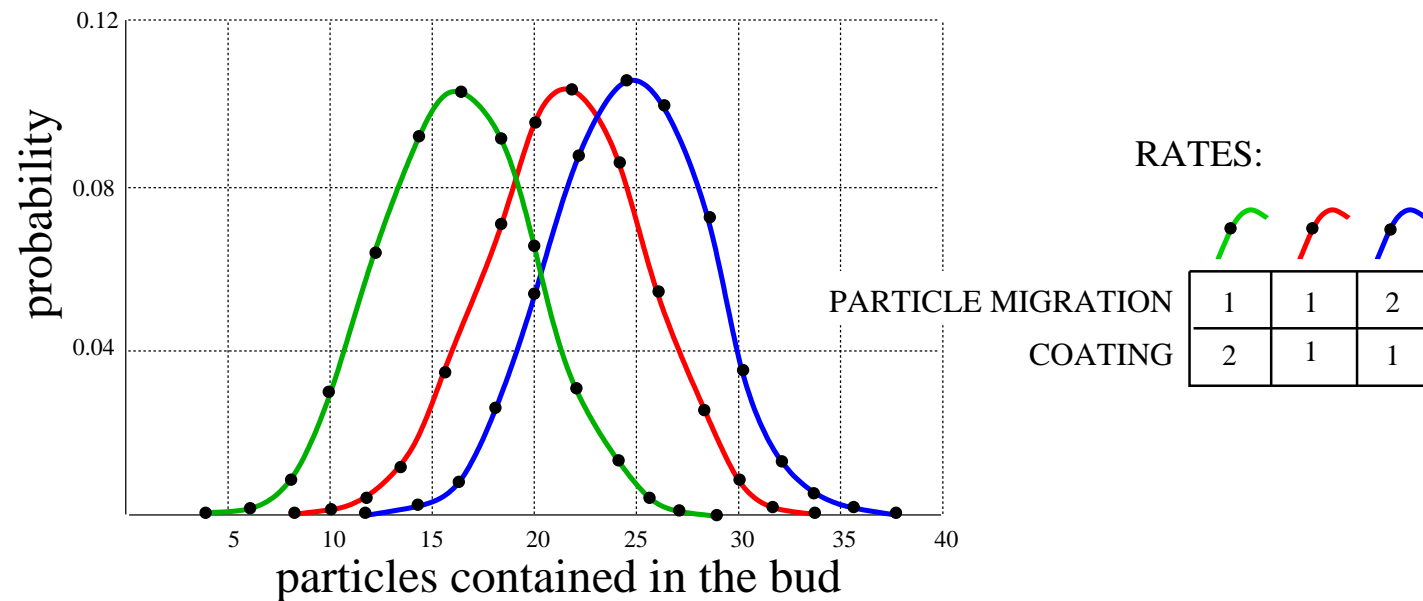
The rate of a particular **reaction**  $g \rightarrow g'$  is given by

$$\sum_i \rho_i \cdot n_i$$

where  $n_i$  is the number of different ways that the  $i^{\text{th}}$  rule can give rise to the reaction  $g \rightarrow g'$ .

The rate of a **labelled transition**  $a \xrightarrow{L} a'$  in a process calculus can be *derived* from rate of its underlying reaction.

## A simulation of budding, using PRISM



As the rate of particle migration increases, relative to the coating rate, the expected number of particles in a bud increases.

This number has a normal distribution of constant width.

## Are bigraphs enough for a **Ubiquitous Abstract Machine**?

- They represent a space for **reconfigurable** entities
- They admit **self-awareness**, hence self-management

## Are bigraphs enough for a **Ubiquitous Abstract Machine**?

- They represent a space for **reconfigurable** entities
- They admit **self-awareness**, hence self-management
- They embed many **process calculi** and recover their theory
- They can provide models for **natural science**

## Are bigraphs enough for a **Ubiquitous Abstract Machine**?

- They represent a space for **reconfigurable** entities
- They admit **self-awareness**, hence self-management
- They embed many **process calculi** and recover their theory
- They can provide models for **natural science**
- They yield a **programming** language

## PARTS OF THE TALK

- What are Informatic Models? How do they fit together?
- Ubiquitous Computing, and modelling it
- Space and Motion in large systems
- **Conclusion**

# What's the point of a Grand Challenge in informatics?

**To make applications that startle the world?**

(e.g. beating a grandmaster at chess)

# **What's the point of a Grand Challenge in informatics?**

**To make applications that startle the world?**

(e.g. beating a grandmaster at chess)

**OR**

**To organise the principles for an engineering science?**

# What's the point of a Grand Challenge in informatics?

To make applications that startle the world?

(e.g. beating a grandmaster at chess)

**OR**

To organise the principles for an engineering science?

**The first alone may (or may not) spin off science**

**The two together will embed computing  
in our scientific culture**

....oooo0000OOOO0000oooo....



## Acknowledgements, References

Thanks to: Ole Jensen and Jamey Leifer for helping bigraphs to get going, and Jean Bezivin, Michael Jackson and Jeff Kramer for discussions on models.

M. (2006) **Ubiquitous Computing: shall we understand it?**  
The Computer Journal 49, pp383–389.

M. (2009) **The Space and Motion of Communicating Agents.**  
Cambridge University Press (to appear). Read-only draft at  
[www.cl.cam.ac.uk/~rm135/Bigraphs-draft.pdf](http://www.cl.cam.ac.uk/~rm135/Bigraphs-draft.pdf).

L. Birkedal *et al* (2004) **Bigraphical programming languages.** Laboratory for Context-Dependent Mobile Communication, IT University, Copenhagen.  
[www.itu.dk/research/bp1/](http://www.itu.dk/research/bp1/).

J. Krivine, M. and A. Troina (2008) **Stochastic Bigraphs.**  
Proc. 4th Conf on *Math. Foundations of Programming Systems*,  
Electronic Notes in Theoretical Computer Science.