A SCIENTIFIC HORIZON FOR COMPUTING

Robin Milner, WCC 2004

A SCIENTIFIC HORIZON FOR COMPUTING

Robin Milner, WCC 2004

- **Grand Challenges**: what and why?

A SCIENTIFIC HORIZON FOR COMPUTING

Robin Milner, WCC 2004

- **Grand Challenges**: what and why?

- One Challenge: **A science for Global Ubiquitous Computing**

A SCIENTIFIC HORIZON FOR COMPUTING

Robin Milner, WCC 2004

- **Grand Challenges**: what and why?
- One Challenge: **A science for Global Ubiquitous Computing**
- **Mounting** this Challenge

A SCIENTIFIC HORIZON FOR COMPUTING

Robin Milner, WCC 2004

- **Grand Challenges**: what and why?
- One Challenge: **A science for Global Ubiquitous Computing**
- **Mounting** this Challenge
- Some **beginnings**

WHAT IS A GRAND CHALLENGE?

-
-
-

WHAT IS A GRAND CHALLENGE?

- A fifteen year project in **basic science** or **engineering** . . .
-
-

WHAT IS A GRAND CHALLENGE?

- A fifteen year project in **basic science** or **engineering** . . .
- . . . with clear **goals**, clear **failure criteria** . . .
-

WHAT IS A GRAND CHALLENGE?

- A fifteen year project in **basic science** or **engineering** . . .
- . . . with clear **goals**, clear **failure criteria** . . .
- and **worldwide participation**.

WHAT IS A GRAND CHALLENGE EXERCISE?

-
-
-
-

WHAT IS A GRAND CHALLENGE **EXERCISE**?

- The community examines and adopts **long-term goals** . . .
-
-
-

WHAT IS A GRAND CHALLENGE **EXERCISE**?

- The community examines and adopts **long-term goals** . . .
- . . . from **within the science**, not outside it.
-
-

WHAT IS A GRAND CHALLENGE **EXERCISE**?

- The community examines and adopts **long-term goals** . . .
- . . . from **within the science**, not outside it.
- Thus to develop and refine a **portfolio of proposals** . . .
-

WHAT IS A GRAND CHALLENGE **EXERCISE**?

- The community examines and adopts **long-term goals** . . .
- . . . from **within the science**, not outside it.
- Thus to develop and refine a **portfolio of proposals** . . .
- . . . showing the public (and funders) **what we aspire to**.

THE UK GRAND CHALLENGE PROPOSALS

1 IVIS: In Vivo ⇔ In Silico

2 Science for Global Ubiquitous Computing

3 Memories for Life

4 Scalable Ubiquitous Computing Systems

5 Architecture for Brain and Mind

6 Dependable Systems Evolution

7 Journeys in Non-Classical Computation

SCIENCE FOR GLOBAL UBIQUITOUS COMPUTING

-
-
-

SCIENCE FOR GLOBAL UBIQUITOUS COMPUTING

- By 2020, a single **Global Ubiquitous Computer (GUC)**
-
-

SCIENCE FOR GLOBAL UBIQUITOUS COMPUTING

- By 2020, a single **Global Ubiquitous Computer (GUC)**
- Part designed, part natural phenomenon
-

SCIENCE FOR GLOBAL UBIQUITOUS COMPUTING

- By 2020, a single **Global Ubiquitous Computer (GUC)**
- Part designed, part natural phenomenon
- Shall we understand it?

UNDERSTANDING and BUILDING

- Underlying both are **modelling kits**
-
-
-

UNDERSTANDING and BUILDING

- Underlying both are **modelling kits**

- **Traditional science and engineering** has
  Differential equations, Laplace transforms, Matrix algebra, . . .
  . . . and they join understanding with building

-

-

UNDERSTANDING and BUILDING

- Underlying both are **modelling kits**

- **Traditional science and engineering** has
  Differential equations, Laplace transforms, Matrix algebra, . . .
  . . . and they join understanding with building

- **Computer science and engineering** has
  Automata, Languages, Relational algebra, Network theories, Logics, Stochastics, Type theory, Process calculi, Semi-structured data, . . .
  . . . but the junction is tenuous. **Why?**

-

UNDERSTANDING and BUILDING

- Underlying both are **modelling kits**

- **Traditional science and engineering** has
  Differential equations, Laplace transforms, Matrix algebra, . . .
  . . . and they join understanding with building

- **Computer science and engineering** has
  Automata, Languages, Relational algebra, Network theories, Logics, Stochastics, Type theory, Process calculi, Semi-structured data, . . .
  . . . but the junction is tenuous. **Why?**

- For **Ubiquity??** Separation will lead to *stagnation or worse.*

JUNCTION BETWEEN SCIENCE AND ENGINEERING IN COMPUTING

TENUOUS: theories only retrofitted. WHY?

JUNCTION BETWEEN SCIENCE AND ENGINEERING IN COMPUTING

TENUOUS: theories only retrofitted. WHY?

- **The Y2000 Problem**: We knew the type theory to avoid it!
  The IEEE 802.11 WEP standard: We knew the analysis to debug it!

BUT

JUNCTION BETWEEN SCIENCE AND ENGINEERING IN COMPUTING

TENUOUS: theories only retrofitted. WHY?

- **The Y2000 Problem**: We knew the type theory to avoid it!
  The IEEE 802.11 WEP standard: We knew the analysis to debug it!

BUT

- Theories lag behind technology:
  This forms the habit not to use them

JUNCTION BETWEEN SCIENCE AND ENGINEERING IN COMPUTING

TENUOUS: theories only retrofitted. WHY?

- **The Y2000 Problem**: We knew the type theory to avoid it!
  The IEEE 802.11 WEP standard: We knew the analysis to debug it!

BUT

- Theories lag behind technology:
  This forms the habit not to use them

- In the software market speed of delivery is paramount:
  Software houses cannot afford time to deploy theories

The Challenge: SCIENCE FOR GLOBAL UBIQUITOUS COMPUTING

- *To develop an informatic science whose* **concepts**, **calculi**, **theories** *and automated* **tools** *allow descriptive and predictive analysis of the GUC at each level of* **abstraction**

-

The challenge: SCIENCE FOR GLOBAL UBIQUITOUS COMPUTING

- *To develop an informatic science whose* **concepts**, **calculi**, **theories** *and automated* **tools** *allow descriptive and predictive analysis of the GUC at each level of* **abstraction**

- *That every* **system** *and* **software** *construction —including languages— for the GUC shall employ only these concepts and calculi, and be* **analysed** *and* **justified** *by these theories and tools.*

www.nesc.ac.uk/esi/events/
Grand_Challenges/proposals/Ubiq.pdf

The Challenge: SCIENCE FOR GLOBAL UBIQUITOUS COMPUTING

- *To develop an informatic science whose* **concepts**, **calculi**, **theories** *and automated* **tools** *allow descriptive and predictive analysis of the GUC at each level of* **abstraction**

- *That every* **system** *and* **software** *construction —including languages— for the GUC shall employ only these concepts and calculi, and be* **analysed** *and* **justified** *by these theories and tools.*

www.nesc.ac.uk/esi/events/Grand_Challenges/proposals/Ubiq.pdf

An *ideal* goal? But no argument limits the degree of possible success!

PLATFORM FOR GUC RESEARCH

Basic notions *Automata; Relational databases; Program logics; Verification; Mathematical semantics; Type theories; . . .*

PLATFORM FOR GUC RESEARCH

Basic notions *Automata; Relational databases; Program logics; Verification; Mathematical semantics; Type theories; . . .*

Concurrent systems *Petri nets; Process calculi; Logics of action; . . .*

PLATFORM FOR GUC RESEARCH

Basic notions *Automata; Relational databases; Program logics; Verification; Mathematical semantics; Type theories; . . .*

Concurrent systems *Petri nets; Process calculi; Logics of action; . . .*

Ubiquity *Mobility (ambients, pi calculus); Security and privacy; Boundaries, resources and trust; Distributed data; Game-theoretic models; Hybrid systems; Stochastics; Model-checking; . . .*

www.cl.cam.ac.uk/users/rm135/plat.pdf

PLATFORM FOR GUC RESEARCH

Basic notions *Automata; Relational databases; Program logics; Verification; Mathematical semantics; Type theories; . . .*

Concurrent systems *Petri nets; Process calculi; Logics of action; . . .*

Ubiquity *Mobility (ambients, pi calculus); Security and privacy; Boundaries, resources and trust; Distributed data; Game-theoretic models; Hybrid systems; Stochastics; Model-checking; . . .*

`www.cl.cam.ac.uk/users/rm135/plat.pdf`

. . . *and* **what more?**

GUC RESEARCH: THREE KINDS OF PROJECT

Experimental Applications: *Collaborate* with designers of specific systems.

GUC RESEARCH: THREE KINDS OF PROJECT

Experimental Applications: *Collaborate* with designers of specific systems. Examples:

- *A sentient building*
- *Healthcare coordinated across a city*
- *Platform for business processes*

GUC RESEARCH: THREE KINDS OF PROJECT

Experimental Applications: *Collaborate* with designers of specific systems. Examples:

- *A sentient building*
- *Healthcare coordinated across a city*
- *Platform for business processes*

Experimental Generic Systems: *Collaborate* with sister engineering Grand Challenge, **Scalable Ubiquitous Computing Systems**.

GUC RESEARCH: THREE KINDS OF PROJECT

Experimental Applications: *Collaborate* with designers of specific systems. Examples:

- *A sentient building*
- *Healthcare coordinated across a city*
- *Platform for business processes*

Experimental Generic Systems: *Collaborate* with sister engineering Grand Challenge, **Scalable Ubiquitous Computing Systems**. Examples:

- *Resource allocation in open distributed environment*
- *Database design respecting provenance*
- *Logic and language for reflectivity*

GUC RESEARCH: THREE KINDS OF PROJECT

Experimental Applications: *Collaborate* with designers of specific systems. Examples:

- *A sentient building*
- *Healthcare coordinated across a city*
- *Platform for business processes*

Experimental Generic Systems: *Collaborate* with sister engineering Grand Challenge, **Scalable Ubiquitous Computing Systems**. Examples:

- *Resource allocation in open distributed environment*
- *Database design respecting provenance*
- *Logic and language for reflectivity*

A Theoretical Hierarchy: . . .

A THEORETICAL HIERARCHY

Theoretical goals for the Grand Challenge:

-
-

A THEORETICAL HIERARCHY

Theoretical goals for the Grand Challenge:

- *To express theories for the GUC as a hierarchy of* **models** *and* **languages**, *assigning each concept (*e.g. *trust) to a certain level in the hierarchy*

-

A THEORETICAL HIERARCHY

Theoretical goals for the Grand Challenge:

- *To express theories for the GUC as a hierarchy of* **models** *and* **languages**, *assigning each concept (*e.g. *trust) to a certain level in the hierarchy*

- *To define, for each model M, how a system description in M may be* **realised** *or* **implemented** *in models/languages $M_1, \ldots, M_n$ lying below M*

A THEORETICAL HIERARCHY

Theoretical goals for the Grand Challenge:

- *To express theories for the GUC as a hierarchy of* **models** *and* **languages**, *assigning each concept (*e.g. *trust) to a certain level in the hierarchy*

- *To define, for each model M, how a system description in M may be* **realised** *or* **implemented** *in models/languages $M_1, \ldots, M_n$ lying below M*

Why do we need models at many abstraction levels?

LEVELS OF MODELLING

Higher levels: **logical, descriptional, specificational**

LEVELS OF MODELLING

Higher levels: **logical, descriptional, specificational**

- **security and authentication requirements; logic of trust; beliefs, intentions; reflectivity requirements; failure strategy; probability limits on performance/failure; . . .** *many higher levels*

LEVELS OF MODELLING

Higher levels: **logical, descriptional, specificational**

- **security and authentication requirements; logic of trust; beliefs, intentions; reflectivity requirements; failure strategy; probability limits on performance/failure; . . .** *many higher levels*

Lower levels: **structural dynamics, coding**

LEVELS OF MODELLING

Higher levels: **logical, descriptional, specificational**

- **security and authentication requirements; logic of trust; beliefs, intentions; reflectivity requirements; failure strategy; probability limits on performance/failure; . . .** *many higher levels*

Lower levels: **structural dynamics, coding**

- **locality refinement; programming; routing; assembly code: . . .**

LEVELS OF MODELLING

Higher levels: **logical, descriptional, specificational**

- **security and authentication requirements; logic of trust; beliefs, intentions; reflectivity requirements; failure strategy; probability limits on performance/failure; . . .** *many higher levels*

Lower levels: **structural dynamics, coding**

- **locality refinement; programming; routing; assembly code: . . .**

- *many lower levels* – e.g. **higher-level language** compiled to **code**, **action-at-distance** realised by **explicit message routing**

THINGS TO THINK ABOUT . . .

THINGS TO THINK ABOUT . . . TODAY!

A BEGINNING: STRUCTURAL DYNAMICS

- GUC systems reconfigure both their *topography* and their *connectivity*, both physical and virtual.

A BEGINNING: STRUCTURAL DYNAMICS

- GUC systems reconfigure both their *topography* and their *connectivity*, both physical and virtual.

- Mobile processes can be modelled by **pi-calculus** and by **mobile ambients** . . .

A BEGINNING: STRUCTURAL DYNAMICS

- GUC systems reconfigure both their *topography* and their *connectivity*, both physical and virtual.

- Mobile processes can be modelled by **pi-calculus** and by **mobile ambients** . . .

- . . . so try using **bigraphs**, which generalise these.

A BEGINNING: STRUCTURAL DYNAMICS

- GUC systems reconfigure both their *topography* and their *connectivity*, both physical and virtual.

- Mobile processes can be modelled by **pi-calculus** and by **mobile ambients** . . .

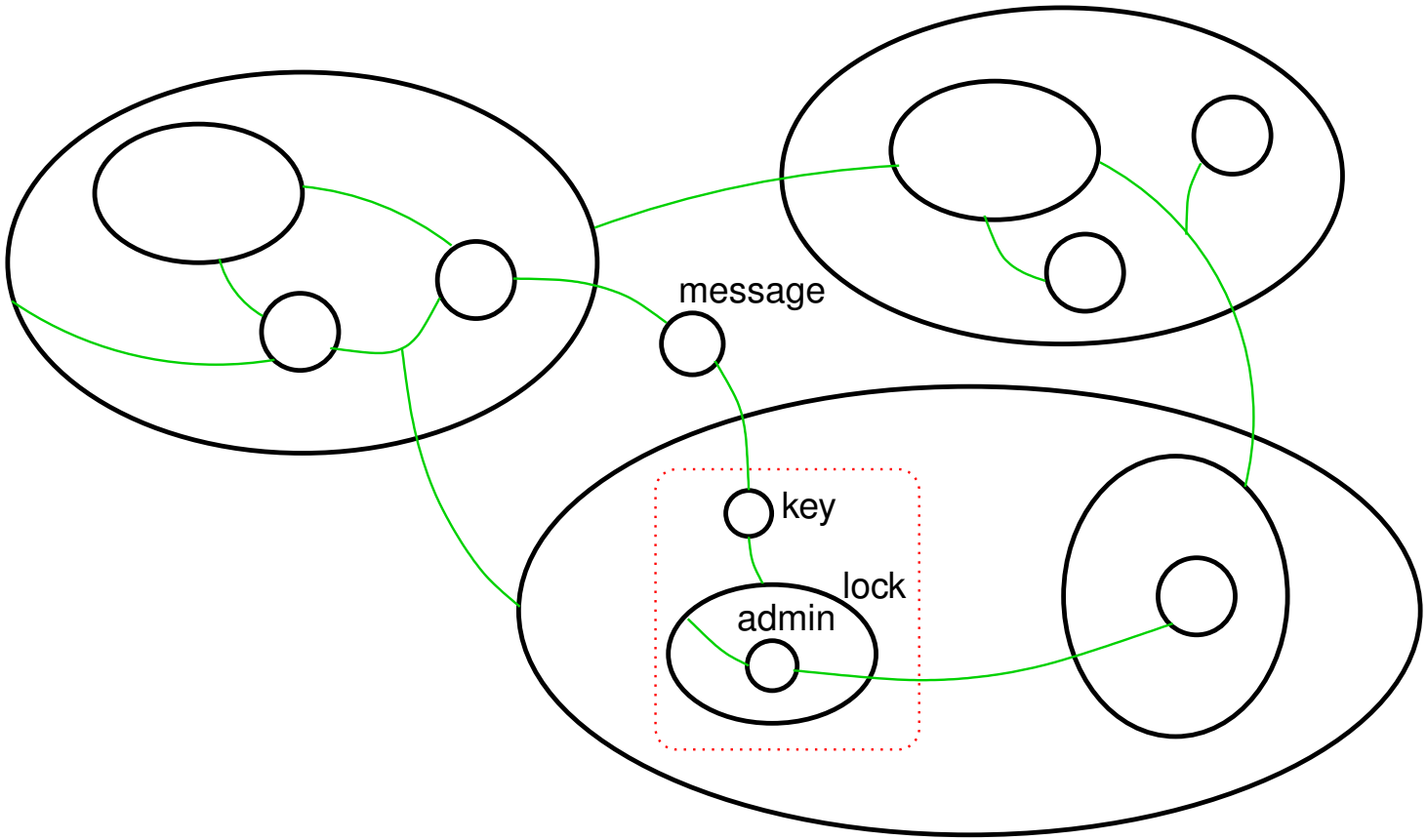
- . . . so try using **bigraphs**, which generalise these.

- Then extend to a *stochastic* model with *continuous time and space* . . .

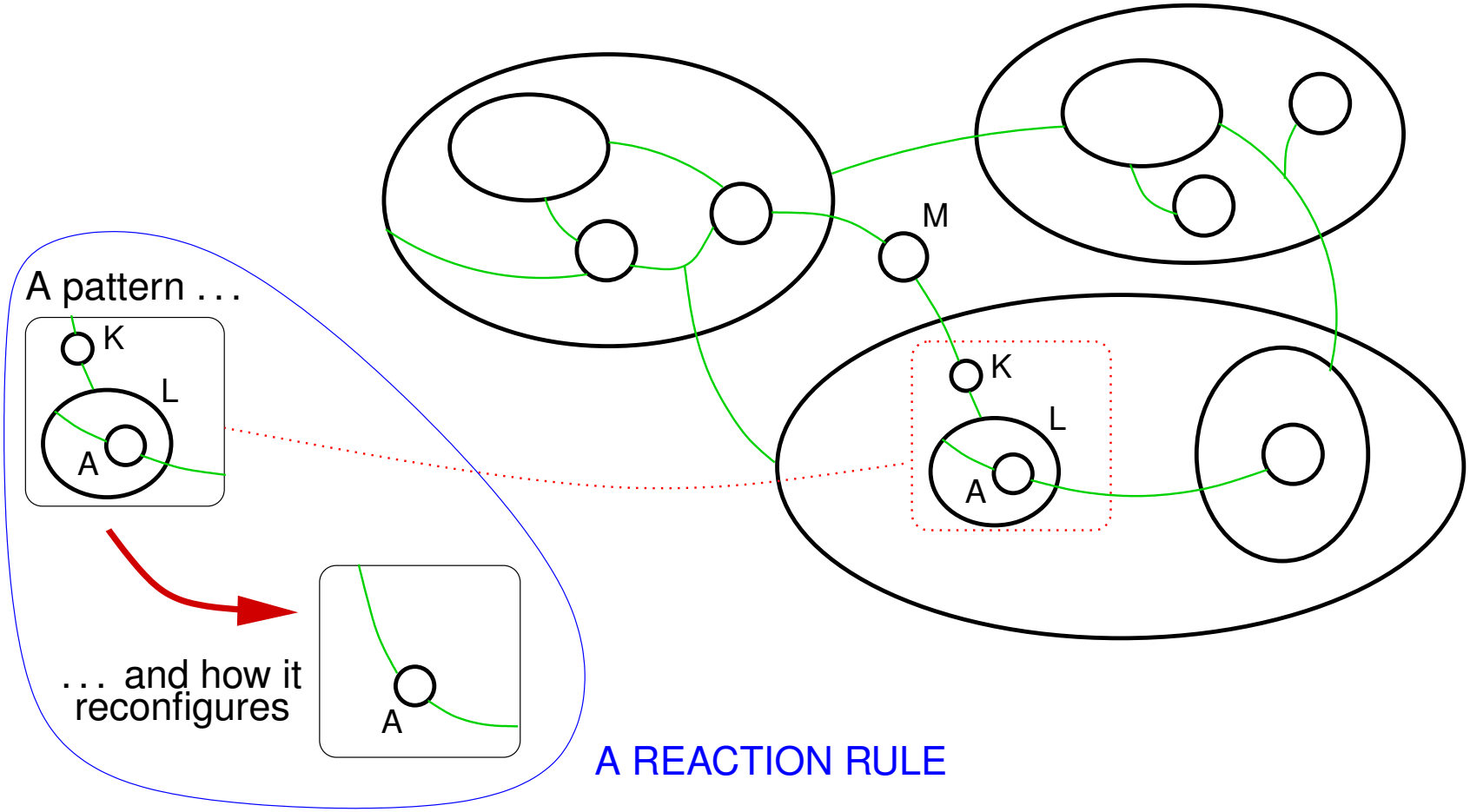
A BEGINNING: STRUCTURAL DYNAMICS

- GUC systems reconfigure both their *topography* and their *connectivity*, both physical and virtual.
- Mobile processes can be modelled by **pi-calculus** and by **mobile ambients** . . .
- . . . so try using **bigraphs**, which generalise these.
- Then extend to a *stochastic* model with *continuous time and space* . . .
- . . . both for *modelling* and for *programming*.

A TYPICAL BIGRAPH

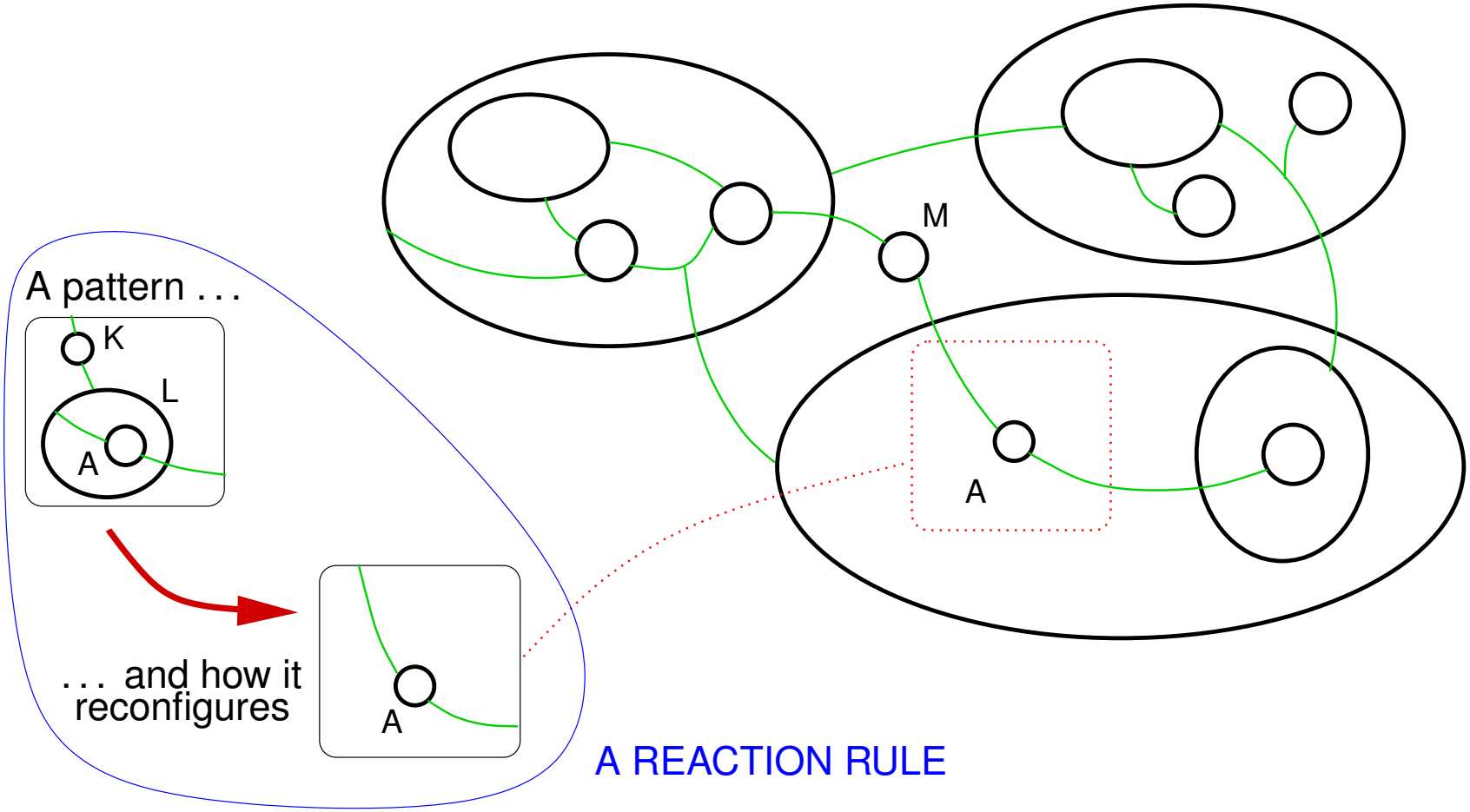


HOW A SYSTEM MAY RECONFIGURE . . . . . .



A REACTION RULE

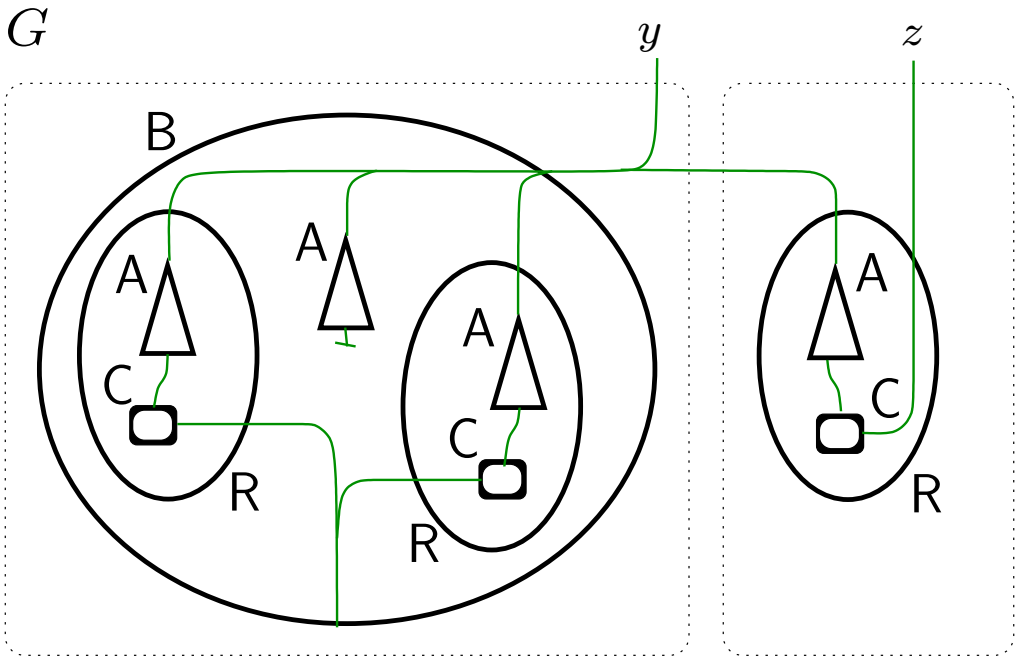
. . . . . AND THE NEW CONFIGURATION



INTERACTIONS IN A BUILT ENVIRONMENT (1)

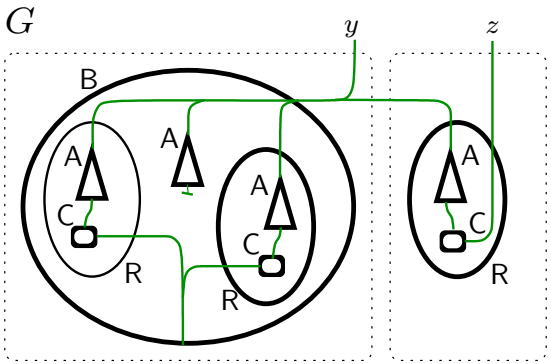
A bigraph G with two regions, representing a conference call

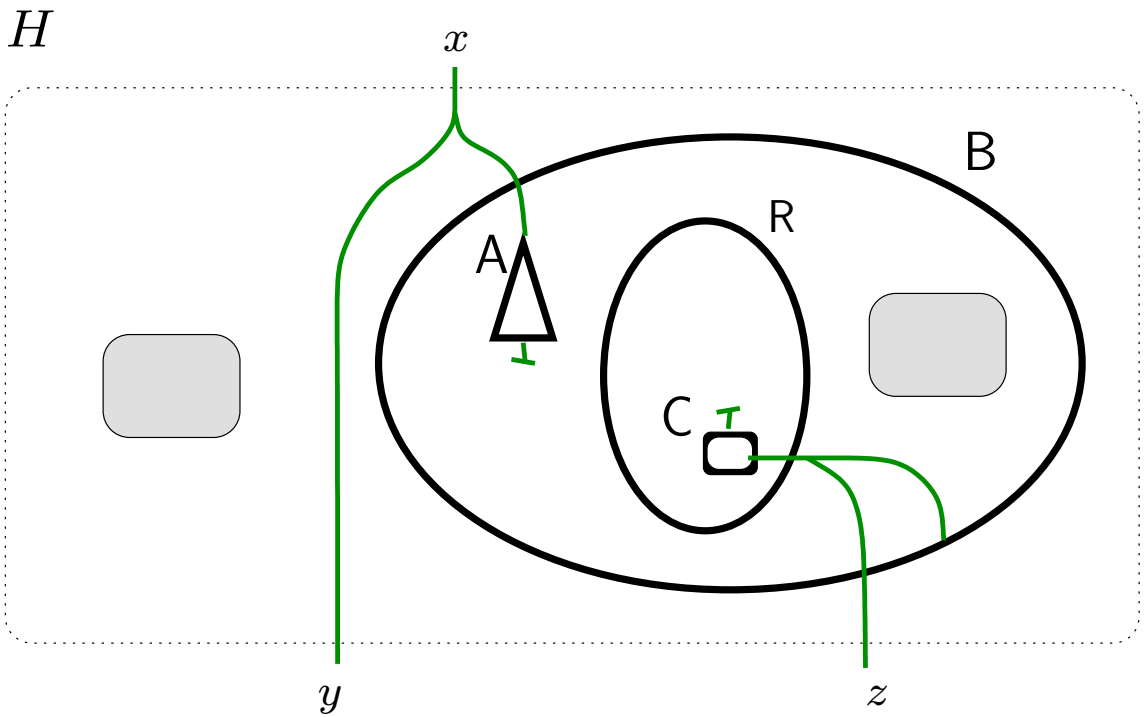
- B BUILDING
- R ROOM
- A AGENT
- C COMPUTER



INTERACTIONS IN A BUILT ENVIRONMENT (2)

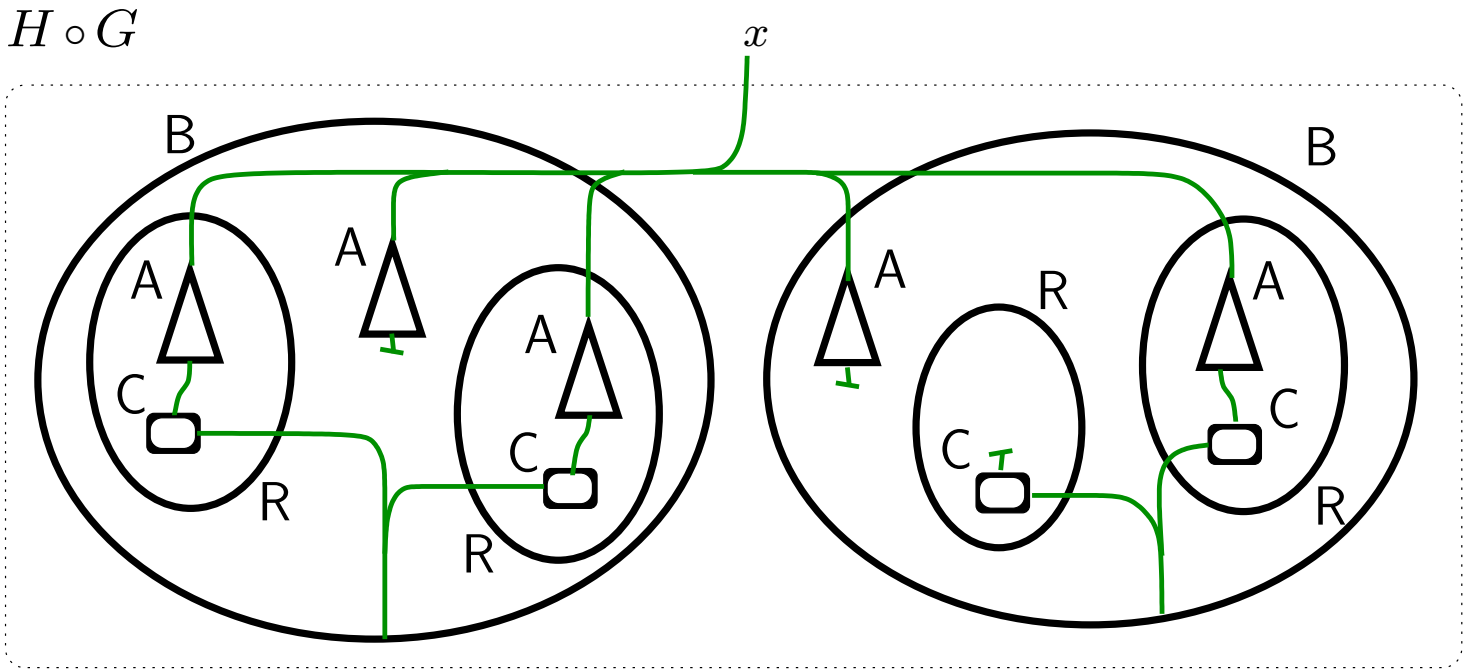
A host environment H, which G may inhabit





INTERACTIONS IN A BUILT ENVIRONMENT (3)

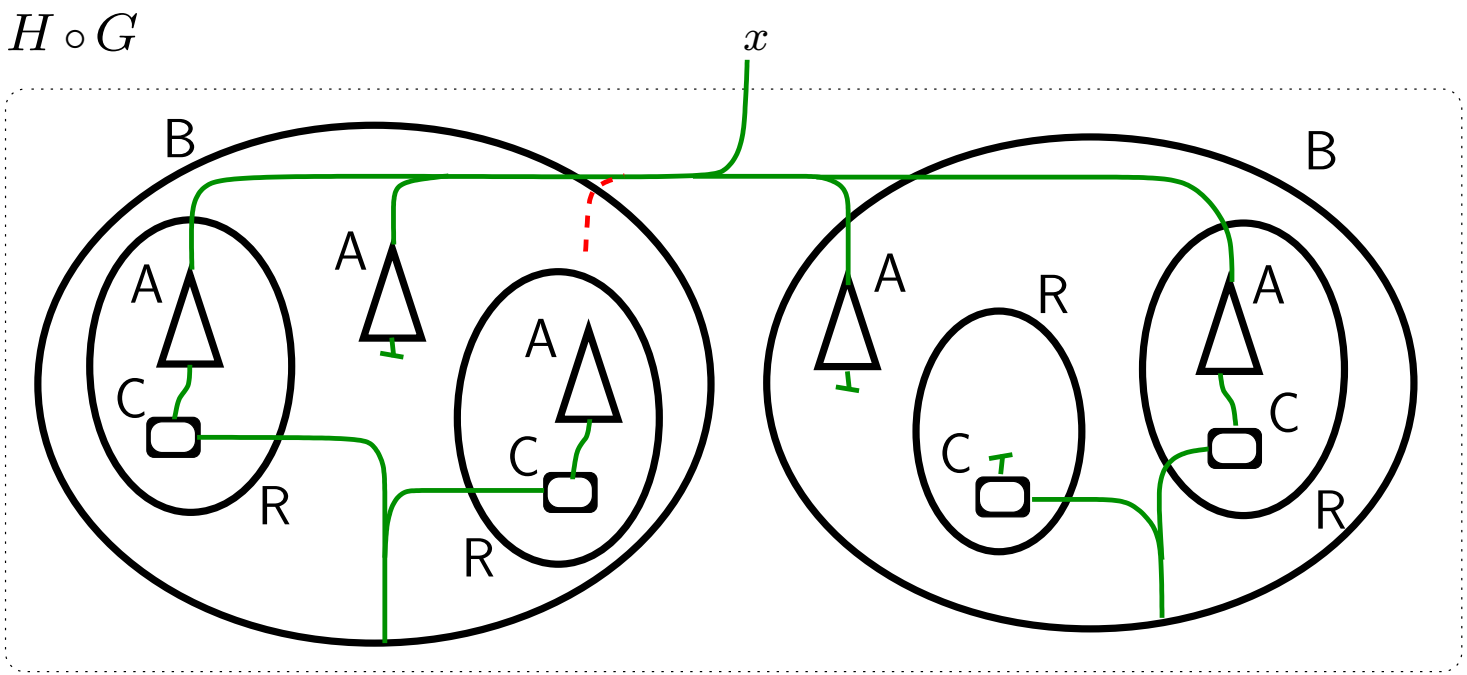
The larger environment, $H \circ G$.



INTERACTIONS IN A BUILT ENVIRONMENT (3)

The larger environment, $H \circ G$.

One agent leaves the call!

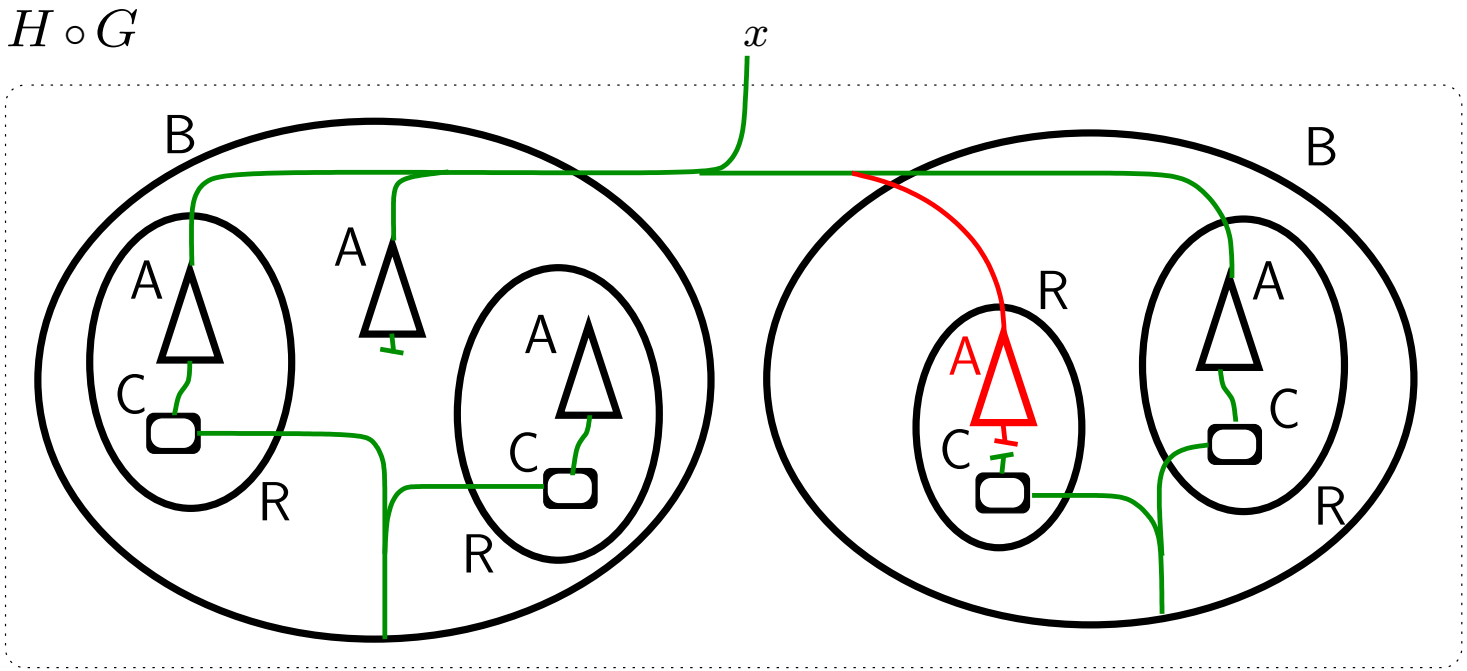


INTERACTIONS IN A BUILT ENVIRONMENT (3)

The larger environment, $H \circ G$.

One agent leaves the call!

Another moves into a room!



CONCLUSION

- The **challenge** is to devise computational *theories* for GUC system alongside the *engineering* of those systems. . .

AND

- . . . the **sub-challenge** is to establish dialogue between the *theorists* and the *engineers*.

FACTS on the GUC proposal

www.nesc.ac.uk/esi/events/Grand_Challenges/

Two linked Grand Challenge proposals. The website contains their proposals, and runs a discussion group. Please join and participate!

- **Science for the Global Ubiquitous Computer**
  moderated by Marta Kwiatkowska and Vladi Sassone.

- **Scalable Ubiquitous Computing Systems**
  moderated by Jon Crowcroft.

- Both these GCs subscribe to **UbiNet**, a UK network on all aspects of ubiquitous computing, at **www-dse.doc.ic.ac.uk/Projects/UbiNet/** .

- Also linked to an FET initiative GC2, **Building the Case for Global Computing**, for Framework 6 of the EC. **www.cogs.susx.ac.uk/users/vs/gc2/gc2.pdf** .