

SCIENCE FOR GLOBAL UBIQUITOUS COMPUTING

Robin Milner, UbiNet 2003

-
-
-

SCIENCE FOR GLOBAL UBIQUITOUS COMPUTING

Robin Milner, UbiNet 2003

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

-

-

SCIENCE FOR GLOBAL UBIQUITOUS COMPUTING

Robin Milner, UbiNet 2003

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

SCIENCE FOR GLOBAL UBIQUITOUS COMPUTING

Robin Milner, UbiNet 2003

- 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*.

A GRAND CHALLENGE FOR 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***



A GRAND CHALLENGE FOR 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](http://www.nesc.ac.uk/esi/events/Grand_Challenges/proposals/Ubiq.pdf)

A GRAND CHALLENGE FOR 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](http://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

Why do we need models at many abstraction levels?



A THEORETICAL HIERARCHY

Why do we need models at many abstraction levels?

Theoretical goals for the Grand Challenge:



A THEORETICAL HIERARCHY

Why do we need models at many abstraction levels?

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

Why do we need models at many abstraction levels?

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, \dots, M_n lying below M*