

# Science for Global Ubiquitous Computing

## a Grand Challenge for two decades' research

Robin Milner  
Cambridge University, Computer Laboratory

How many computers will we be using, wearing, or have installed in our bodies, in 2020? How many other computers will they be talking to? What will they be saying about us, doing for us, or doing to us?

By that time computers will be ubiquitous and globally connected. It is better not to count them, but to regard them collectively as a single Global Universal Computer (GUC). Who then will program the GUC, and how, and for whose benefit? Shall we still be in control of it, or even understand it?

These questions underlie two recently formulated Grand Challenges for computing research. This one is theoretical: what are the models and theories we need to understand the GUC? The other, **Scalable Ubiquitous Computing Systems**, is concerned with engineering: how shall we design and build the GUC? The two are strongly linked, but the attack will be sharper if their goals are set separately.

The GUC is partly *engineering artefacts*, and partly a *natural phenomenon* (i.e. not fully designed). As engineers we build it; as scientists we seek to understand it. The Challenge is to develop theoretical models, to be deployed consistently both in building and in understanding the GUC, at a hierarchy of levels of abstraction. To ease both tasks, these models must be equipped with calculi —both for programming and for reasoning— and corresponding automated tools. A paradigm shift is sought: that computing theories should inform *every aspect* of the creation of such systems, not merely be retrofitted to analyse them. Many engineering disciplines meet this test, but software engineering often fails it. In particular it fails for the GUC, because only recently have we attained some grasp of the range of concepts that it involves.

What are these concepts? The list will grow, but already includes: Non-determinism and concurrency; Space and mobility; Security and Privacy; Boundaries, resources and trust; Distributed data; Game-theoretic semantics; Hybrid systems; Reflectivity; Stochastics; Model-checking.

To address the Challenge, three categories are defined for its sub-projects: (1) experimental applications (exemplars); (2) experimental generic systems; (3) evolving a hierarchy of theoretical models. These are all essential, to ensure that the theoretical Challenge both informs and is informed by practical systems. The first two categories must be collaborative. Possible exemplar applications are 'A Sentient Building', and 'Healthcare Coordinated across a City'. On experimental generic systems we shall collaborate with our sister Grand Challenge, **Scalable Ubiquitous Computing Systems**.

The third category of projects, to evolve the hierarchy of theoretical models, is the only way to render the GUC susceptible to scientific explanation. Why a hierarchy? Consider just one example: at one level of the hierarchy we must explain the GUC in terms of notions like *trust*. We have to develop a *calculus*

or *logic* of trust, which codifies when and why one agent (of billions) may trust another that seeks to interact with it. But the behaviour which conforms to this trust model will be *enacted* at a lower level in terms of purely operational concepts. Our theory must demonstrate that this enactment is sound. Notions like *security* and *reflectivity* (the ability of a system to monitor and adapt its own performance) demand similar treatment.

So the hierarchy of models lies at the heart of the proposed research. There is already have a precedent for this kind of work in calculi, logics and tools already developed for concurrent interactive systems, so we are not starting from scratch. Success in meeting our Challenge may be measured by how far such theories are not merely developed but applied in practice.