GC2 Science for global ubiquitous computing

Marta Kwiatkowska and Vladimiro Sassone

How many computers will you be using, wearing, or have installed in your body in 2020? How many other computers will they be talking to? What will they be saying about you, doing for you or to you? By that time computers will be ubiquitous and globally connected. It is better not to count them individually, but to regard them collectively as a single Global Ubiquitous Computer (GUC). Who will then program the GUC and how? Shall we be in control of it or even understand it?

Let us imagine what life will be like in 2020. Smart homes will be equipped with autonomous, self-aware sensors, which will be capable of gathering information about their physical and digital environment in order to recognize events, identify threats and take appropriate actions. Intelligent spaces will be created by positioning sensors in the environment, all wirelessly connected and capable of monitoring working conditions and access to restricted parts of the buildings by automatically sensing radio tags worn on clothing. Medical treatment will be personalized, based on our genetic make-up and factors such as age, and even delivered directly to our bodies by wearable devices.

Does this seem too futuristic? Not so, if one considers the degree to which our lives are today dependent on the internet, with access to grid and web services, and involve electronic devices that are ever smaller, wireless and mobile. Already there are hundreds of embedded processors in modern cars, wireless ‘motes’ are deployed in environmental monitoring and industrial control, intelligent buildings are a commercial reality and first successes have been reported with ‘medibots’ (nanoscale DNA computers capable of fighting cancer cells in a test tube).

What are the essential features of an infrastructure to support such a scenario? Firstly, the internet already enables *global connectivity*, by means of wired, radio and satellite communications; this implies the huge *scale* and *complexity* of the system, its highly *distributed* nature, *mobility* and continued *evolution*. Secondly, each node on the network, either sensor or device, is capable of *computation*, *communication* and *information processing*, as it shrinks in size to the microscale, possibly nanoscale. Thirdly, the devices are increasingly *self-aware*, space and location conscious, able to interact with the surrounding environment and exhibiting introspective behaviour. The size of the network, and the complexity of the interactions within it, demand that they be capable of *cooperation*, *self-organization*, *self-diagnosis* and *self-repair*. Finally, *trust*, *privacy*, *security* and *dependability* must be assured, as the cost of malfunction or breach of contract can be very high.

The challenges this raises for both computer engineers and software designers are enormous. The risks in the GUC are great: confidential medical records must never be leaked out to unauthorized parties, drive-by-wire sensors must respond timely, and ‘medibots’ must be dispatched only if it can be guaranteed they work correctly. This is in sharp contrast with the frequent failures of current computer systems, which are orders of magnitude simpler than those we envisage for the GUC.

So, how can we engineer a better GUC? Just as engineers rely on *sciences* of the physical world when building bridges, and use toolkits and theories from Physics and Mathematics to model and evaluate their designs, we need to develop a Science for Global Ubiquitous Computing, a fundamental theory describing ubiquitous computational phenomena. This will involve *conceptual*, *mathematical* and *software tools* to inform and support the design process for ubiquitous systems, via models, programming languages, protocols, analysis techniques, verification tools and simulation. For example, the ‘medibots’ have been modelled using automata theory, temporal logic has been used to specify the security of communication protocols and verification tools to detect their flaws, the pi-calculus models *ad hoc* networks, and simulation is often used to analyse system performance. But we shall need many more new concepts and tools. For instance, the theory must provide logics for trust and resource, and tools for the analysis of crypto-protocols and routing algorithms. It must also span several abstraction levels to provide a flexible framework to harness complexity.

How can science make a difference?

Is there a challenge for science at all? The scenarios we envision are, as illustrated above, typical futuristic, technological scenarios. Why cannot they be conquered as purely engineering, 'hands-on' tasks? The difficulty is that, without rigorous analysis of all the possible interactions between system components in the design phase, it is all too easy to build systems that exhibit erroneous behaviour. The recent discovery of security flaws in the 802.11 and Bluetooth® protocols can serve as a pertinent example.

Of course, new technologies are bound to be experimental and, therefore, they run an intrinsic risk of being flawed. There is nothing wrong with that, and it would be naive for us to advocate a future in which progress and market forces wait for scientists to give the green light. On the other hand, the scientific analysis of technological innovations allows us to understand solutions and their limitations, and is irreplaceable in a field like the GUC, which aims at applications with the potential to radically change the way we live. Enormous wealth, both in moral and physical terms, is currently entrusted to the internet, and this can hardly be expected to change in the GUC future.

Such analysis is the purpose of our challenge. It will underpin the design and engineering that is the focus of our sister challenge Scalable Ubiquitous Computing Systems. Science is not design itself; the two are distinct, but sisters. They proceed with different timescales, tools, principles and milestones. But they only make sense as Grand Challenges when coupled together. We describe below two groups of relevant issues where the scientific, foundational approach will make a difference. The first group is concerned with models.

System and software architectures: We need models that inform the design of large software-intensive systems formed by *ad hoc* networks of heterogeneous components. The models must support evolution, adaptive behaviour, loose coupling, autonomy, context-awareness, learning, security and more.

Mobility, self- and context-awareness: We need calculuses and logics to formalize these notions. For example, how will systems attach a semantic meaning to information received from the contexts met while roaming the global network, and how will they validate such information?

Discrete/continuous models: We need to extend the current limited logics and formalisms to provide predictive theories for hybrid systems (e.g. sensor networks) that feature continuous inputs, for example position and temperature, as well as discrete ones, for example room number.

Stochastic models: We need to adapt the current stochastic models (increasingly important for network protocols) to provide compositional probabilistic analysis of the systems and subsystems that make up the GUC.

Cognition and interaction models: We need to model the cognitive aspects of the GUC, whereby software devices learn during their lifetime, and learn how best to interact with humans inside a smart home.

Knowledge, trust, security and privacy: Migrating devices will acquire information on which they will base action, including interaction with potentially dangerous environments. We need to build models for the acquisition, distribution, management and sharing of such knowledge, and in particular how trust may be based upon it.

The second group of issues is concerned with languages and tools.

Programming language design: We need to isolate language features appropriate to the GUC, especially for complex disciplines of interaction. New data structures, for example based on XML, are of equal importance.

Ubiquitous data: We need to understand how best to embed semi-structured data in programming languages and applications. This entails new type systems (for safety) and theories to underpin the notions of 'certified origin' and 'relevance' of data items.

Protocol design: We need to develop decentralized protocols for information exchange in *ad hoc* wireless and sensor networks. These will likely contain elements of randomization to break the symmetry between the nodes, achieve scalability and improve performance under changing conditions.

Algorithms for coordination, cooperation and autonomy: Interactions in the GUC invite a vision of autonomous, mistrustful, selfish components, with no notion of common purpose and a rather faint notion of common welfare, running in some sort of equilibrium. We need to discover the algorithms to underpin all this.

Software technology and design support tools: We need to upgrade and advance the models which underlie our current design technologies. They will involve new type systems, new forms of static analysis (including topographical analysis) and new validation/testing suites equipped for the GUC.

Verification techniques and technology: A marked success of computation theory has been the acceptance in industry of verification software tools, chiefly model checkers and automated theorem provers. We need to further advance research in this field so that every safety or business critical system can be fully verified before deployment.

Related activities

There are already a number of international and UK activities and projects that can provide support for the research outlined above. The EPSRC Network UK-UbiNet provides a forum for a broad spectrum of activities, from appliance design, through user experiences, to middleware and theory. It has already held two workshops and plans a summer school in the near future. The EU-FET Global Computing initiative is a Framework Sixth programme focusing on the foundations of the GUC. In the UK, the Next Wave initiative of the DTI focuses on products and technology transfer for ubiquitous computing; Mobile Bristol[9] aims at developing user experiences with digital devices. The Equator project[10] is an Interdisciplinary Research Collaboration which focuses on interleaving physical and digital interaction. In the USA, Berkeley's Smart Dust originated the concept of 'motes', cubic millimetre miniature computing devices, which are now being developed at Intel[11] and Crossbow. In a recently founded UK–US partnership, the Cambridge-MIT Institute (CMI), continues the research begun by project Oxygen.[12]

Science for Global Ubiquitous Computing is related to several other Grand Challenges discussed at the Conference in Newcastle (March 2004). The closest is Scalable Ubiquitous Computing Systems, a true sister challenge which shares our domain of interest. Indeed, the two held joint meetings in Newcastle and intend to collaborate in future, thus realizing what Christopher Strachey once famously said:

> *It has long been my personal view that the separation of practical and theoretical work is artificial and injurious. Much of the practical work done in computing ... is unsound and clumsy because the people who do it have not any clear understanding of the fundamental design principles of their work. Most of the abstract mathematical and theoretical work is sterile because it has no point of contact with real computing.*

There are also strong and useful relationships with at least three of the other challenges: Dependable Systems Evolution, since the Global Ubiquitous Computer is highly dynamic and dependability is an essential requirement; *In vivo–In silico*, which shares with our challenge the aim of predictive modelling for system dynamics and composition; and Memories for Life which aims to develop devices and recordings to become part of the physical GUC. We also note that the new National Science Foundation's Science of Design programme[13] for internet and large-scale distributed systems, the infrastructure that underlies the GUC, addresses issues very similar to ours.

First steps and ultimate goals

We would like to begin by defining a 'road map': a conjectured path or group of paths leading to the goals. The discussions at the Newcastle Conference led us to the conclusion that it is too early for this,

[9] http://www.mobilebristol.co.uk/flash.html

[10] http://machen.mrl.nott.ac.uk/home.html

[11] http://www.intel.com/research/exploratory/motes.htm

[12] http://oxygen.lcs.mit.edu

[13] http://www.nsf.gov/pubs/2004/nsf04552/nsf04552.htm

largely because the GUC is an artefact that is still developing, and in unpredictable directions. Instead, we decided to focus on initial collaborative experiments, which may be called foothill projects, to help us establish a case for such a road map. The projects arising from the panel discussions include:

Assuring privacy, authentication and identity of medical records: An exemplar that enables anytime/anywhere accessibility to medical records to doctors or patients in a provably secure manner will be developed. It will be based on sound models of trust- and privacy-ensuring technologies, and will consider human and digital communications of various kinds. Collaboration between security experts, researchers in human–computer interfaces, and the medical and legal professions will be needed.

Rigorous process models for web services: We are increasingly dependent on web services for information, but are often frustrated with their unreliability and untimely responses. A collaboration between researchers in process calculuses and middleware will address rigorous foundations of web services, and will identify key coordination primitives for the GUC software architecture.

Communications infrastructure protocols: As the GUC continues to evolve, we will see more and more reliance on sensor and wireless networks in addition to the conventional internet (TCP/IP) protocols. New communication and routing protocols will have to be designed, implemented, verified and analysed. This project will involve networking experts, middleware developers and verification researchers.

We aim to devise a *science* of the GUC: a set of concepts and theories that underpin and play a driving role in the design and engineering of the GUC and its components, which will assist in the modelling, analysis, diagnosis, evaluation and validation of the design. Our goals are:

> *to develop a coherent informatic science whose concepts, calculuses, theories and automated tools allow descriptive and predictive analysis of the GUC at many levels of abstraction; and that every system and software construction, including languages, for the GUC shall employ only these concepts and calculuses, and be analysed and justified by these theories and tools.*

We envisage a 15-year time frame for this challenge to come to fruition, and much of the research will require global effort. It is, of course, possible that our challenge will never be entirely met. However, even partial successes will yield strong advantages. And it is ultimately the responsibility of computing research to place the GUC on as rigorous a basis as possible.