Ubiquitous Computing: Experience, Design and Science

Version 4 23/2/06

http://www-dse.doc.ic.ac.uk/Projects/UbiNet/GC/index.html

Dan Chalmers, Matthew Chalmers, Jon Crowcroft, Marta Kwiatkowska, Robin Milner, Eamonn O'Neill, Tom Rodden, Vladimiro Sassone, Morris Sloman

1 The Challenge

The emergence of powerful digital infrastructures, wireless networks and mobile devices has already started to move computing away from the desktop and embed it in the public spaces, architectures, furniture and personal fabric of everyday life. Handheld and wearable computers, mobile phones, digital cameras, satellite navigation, and a host of similar devices join the Personal Computer as commonplace digital tools. We are increasingly becoming accustomed to using a heterogeneous collection of computing devices to support a growing range of activities. These embryonic forms of ubiquitous computing technology have already had a major impact on the ways that people work, learn, entertain themselves, and interact.

The current generation of interconnected devices represents only the start of a shift towards a world of ubiquitous computing. Such devices will continue to diversify in the ways in which they sense and impact the physical world. We will increasingly share the world we inhabit with a massive set of embedded computational elements capable of sensing our activities and responding to them in a variety of ways. This shift requires us to change our view of computing from its current device centred perspective to one where "it is invisible, everywhere computing that does not live on a personal device of any sort, but is in the woodwork everywhere."[1] This has fundamental consequences for how we might reason about, construct and use computer systems. The technology has developed apace, far ahead of our ability to reason about these new systems, to develop the engineering principles underpinning their construction, and to understand how we might experience the ubiquitous environment enabled by them.

Let us illustrate the scope of this challenge by considering a medical scenario. Envisage that the entire population of the UK is to be instrumented as part of the NHS for systematic monitoring of metrics such as heartbeat, skin conductivity, blood sugar etc. In fact, we are already seeing embryonic explorations of this arrangement for people at risk, allowing medical staff to take prompt remedial action. If this arrangement could be applied to all it would have significant benefits for health care, health promotion and medical research. We might envisage dynamic medical records that are both up-to-date and consistent, between patient, hospital and even emergency services. We could consider timely intervention and diagnostics. We might even envisage autonomic responses where major incidents are dealt with in a timely manner. For example, these responses may trigger defibrillators for cardiac patients undergoing unattended attack. This scenario illustrates some of the key issues that need to be resolved, as a world unfolds in which ubiquitous computing is the norm.

In essence, ubiquitous computing is a challenge that affects all of computer science. It asks fundamental questions about how we might reason about computer systems and computability, how we might develop complex ubiquitous systems, and how we might understand the experience of environments that are supported by ubiquitous computing. The challenge draws together researchers from three distinct perspectives:

- **The experience perspective** focuses on how people might share a world with ubiquitous computing environments. What interactive principle underpins our interaction with them, and how might a ubiquitous computing society be shaped from a socio-technical perspective?
- **The engineering perspective** focuses on the architectural and network challenges posed by the large scale, heterogeneous and dynamic nature of ubiquitous computing. What engineering principles are needed to allow a vast array of devices to be interconnected in a system, and how might we understand and respond to the system's emergent behaviour?
- **The theoretical perspective** focuses on concepts and rigorous models that capture the behaviour of ubiquitous systems at varying levels of abstraction. How do we reason about such a system, in order to understand its aggregate behaviour in terms of the behaviour of its subsystems?

[1] http://www.ubiq.com/hypertext/weiser/UbiHome.html

The core of Ubiquitous Computing lies in the convergence of these different perspectives, leading to a successful blend among them. This requires fundamental research into each of the constituent areas. While each of these has its own distinct perspectives and goals, they are closely linked. They may advance with somewhat distinct time-scales, tools, principles and milestones, but their development will be coordinated by projects that contribute to each perspective. Collectively, they constitute a response to a Grand Challenge whose goals are as follows:

- To develop ubiquitous computing methods and techniques that are sensitive both to the needs of individuals and society, and the impact upon them. These will support the realisation of human experiences and will include new forms of interaction and new interaction paradigms that make ubiquitous computing usable by all.
- To define a set of system design principles that pertain to all aspects of ubiquitous computing; are agreed among both academic and professional engineers; are taught regularly in Master's Degree courses; and are instantiated in the design and rigorous documentation of several computational systems with a successful operational history.
- To develop a coherent informatics science whose concepts, calculi, models, theories and tools allow descriptive, explanatory and predictive analysis of ubiquitous computing at many levels of abstraction; to employ these analyses to derive all its systems and software, including languages; and to justify all its constructions by these analytic tools.

These are ideal goals, but there is no argument that places a limit on the extent to which they can be achieved. The Grand Challenge must be addressed by collaboration across these perspectives, developing scientific theory, engineering principles and usage guidelines together in an iterative manner. In the spirit of a Grand Challenge the second and third goals are phrased to allow assessment, after one or two decades, of success in attacking the goals. The unpredictable nature of ubiquitous computing makes it impossible, at present, to formulate criteria of success for the first goal; the desired forms of experience will only emerge incrementally.

What are the qualities that characterise a Ubiquitous Computing System (UCS)? It should be embedded in and become part of the environment, allowing the user to focus on the activity at hand and not the system. Its purpose is to serve people; this certainly entails interaction with users and control by them – dealing with breakdowns, setting preferences etc. Nevertheless, much of its management (e.g. configuration, handling of faults and adaptation to context) will be done autonomously and people will not be aware of it. A UCS may involve large – even enormous – populations of entities that deploy themselves flexibly and responsibly in its work. An entity may be a hardware device, a software agent or an infrastructure server; for some purposes it may be a human; it may also be an agglomeration of smaller entities. Thus a UCS that pervades our lives, but remains controllable, will demonstrate many qualities:

- It will be **fluid**; its structure will vary in the short term and evolve in the long term.
- Each non-human entity will be **purposive,** whether its purpose is expressed vaguely or formally; this is what explains its actions.
- It will be partially **autonomous**; some of its actions are determined by its purpose and its interactive experience, rather than by invocation from a higher authority.
- It will be **reflective**; a subsystem can report its experience to a higher system (perhaps to a human), to permit intervention or guidance.
- It will be **trustworthy**; it will behave in a dependable manner and will not adversely affect information, other components of the system or people.
- It will be **sustainable**; its components – hardware and software – are designed and built for long-life, efficient and effective maintenance and eventual decomposition, while its lifetime impact on the environment (including humans and power sources) is appropriate but minimal.
- It will be **efficient**; any delays in its performance will be tolerable.
- It will be **scalable**; its subsystems will differ in size by many orders of magnitude, yet unmanageable complexity will be avoided by applying the same principles of design and methods of analysis at each level.

Qualities such as these (there will be more) must permeate the whole of a UCS. In the following sections we expand on many of the concerns we have mentioned, and establish a few subgoals. In each of the following sections the reader may repeatedly detect the relevance of these qualities, even when they are not explicitly mentioned. We consider these subgoals from the experience perspective in Section 2, from the engineering perspective in Section 3 and for the theoretical perspective in Section 4. While these perspectives are presented in separate sections, we emphasise that they pertain to a single Grand Challenge and their activities must be

closely related. In section 5 we propose what our next steps should be. They will include exploratory projects carried out with modest aims; crucially, they will combine different perspectives of experience, systems and theory. When several such projects are complete we can hope to define a roadmap that predicts a ten-year exercise to achieve the goals of the Challenge.

2 The Experience Perspective

The experience perspective focuses on how ubiquitous computing systems might be used to realise environments and how we all might live with them. How will people relate to future ubiquitous computing environments, and how can we best support their design? What sense will people make of a world with a massive number of ubiquitous computing elements? How should these elements present themselves to people? How might we exploit the capabilities suggested by ubiquitous computing environments and what are the implications for the society we live in?

Ubiquitous computing places the user at the centre of a new way of understanding and designing computer systems, by seeking to change how we interact with and experience such systems. Understanding how the user will experience ubiquitous computing involves close relationships with a range of other disciplines and has implications for both the Engineering and Theory perspectives. Those addressing the experience perspective must engage with the engineering and theoretical perspectives, and vice versa.

The need to situate ubiquitous computing technologies within the real world makes people, engaged in individual and social interaction, central to how we might reason about, build and design for ubiquitous computing. Understanding how ubiquitous computing technologies can be interleaved with human activities, and the consequences for technology design, presents a major challenge. A number of fundamental research topics are essential for progress in this key area.

Understanding Human Activities and Context

Understanding the nature of human activities is itself a significant research question for many disciplinary traditions, such as psychology, sociology, and ergonomics. The variety of perspectives creates problems, since each discipline may exploit different tactics to uncover human action. Understanding how to represent human activities for the purpose of ubiquitous computing draws upon these different traditions, and represents a significant multidisciplinary challenge. The diversity of approaches within the human and social sciences highlights the complexity involved. The broad approach of ubiquitous computing often builds upon insight from sociology which focuses the artefacts of use within the real world as situated, contextual and social. This emphasises the richness of human interaction, but also tends to resist more abstract representations of activity which are important to those who seek to engineer and reason about ubiquitous computing systems.

Representing Human Activities and Context

We are faced with the equally daunting task of how we represent human activities in computational models. We need to choose which features of these activities we wish to emphasise, and how to represent them abstractly. Abstract representations of human activity lose much of its subtle detail. They inevitably formalise the nature of the activity and tend to fail to convey how these activities evolve, change and are reflected on by those who are modelled. Despite these reservations, the nature of digital technology means that ubiquitous computing must develop abstract representations. In representing people and artefacts we therefore need to choose what we exclude and include in the model, what we represent in detail, what is abstract, what systems actively interpret and what we leave to human interpretation. The core challenge then is how we shift from the infinite detail of real world activity to the finite and formal representations of computer systems without losing its subtlety. The difficulty of this challenge grows with the number of people, scale of technology and volume of information we wish to work with.

The Nature of Context

To understand and represent human activities, a central role is played by context; indeed, ubiquitous computing is sometimes called ‘context-aware computing.’ The real world, and the activities of those who populate it, raises fundamental questions about the nature and role of context. How to present context to computational elements, and how to exploit these representations, pose fundamental research problems. For example, we can take an objective approach by focusing on the measurable physical characteristics of a device or the location of a person; but how might we reflect subjective and historical features of complex activity? We might wish to consider context as something to be sensed and measured by ubiquitous computing, but it is also dynamic and heterogeneous, in that it covers location, movement, artefacts, buildings, information, people and so forth. It is constructed by people reflecting on their situated activity, in an ongoing process in which past experience and individual understanding, and ongoing interaction with people and artefacts, are influences and constraints. The

challenge, then, is to determine the nature of context and its role in representing and shaping activity. We need to determine how ubiquitous computing technologies help users in shaping this context, and in consequence how we sense, represent and reason about key features of a dynamic and changing world. This work will require significant dialogue with those from an engineering perspective seeking to develop models of context.

2.1 Human Interaction in Ubiquitous Computing

The issue of human interaction pervades ubiquitous computing, as part of its holistic view spanning technology, use and users. We need to determine how people will understand and interact with a ubiquitous environment and how they may use it to interact with other people. Ubiquitous computing requires us to develop new techniques to interact with digital devices and poses fundamental questions for theories of human computer interaction.

Interaction with Environments

As ubiquitous computing environments become increasingly part of our everyday lives, we need to understand how people will interact with and exploit them. We also need to understand how user interaction will help shape these environments. Interaction will interleave influences that are contextual, individual and social and technologies will be appropriated with interaction evolving over time. People face potential challenges when attempting to establish useful or enjoyable interaction with ubiquitous computing technology. This interaction needs to fit with their contexts, interests and aims. The capabilities of different digital technologies, settings of use and individuals' own past experiences act as both resources and constraints in shaping this interaction. Interactive elements in the environment will range from small scale embedded or wearable devices focusing on the individual to large scale installations that focus on the general public. Each interactive element may contribute significant overhead and complexity to users' interactions, if it has a different mode of interaction from other devices and fits badly with people's everyday activities. Reducing this overhead and accommodating varying forms of interaction is central to the design of ubiquitous environments. A major challenge for ubiquitous computing is to discover how people arrive at patterns of interaction that are productive rather than problematic.

Interaction through Environments

Ubiquitous computing environments will not only provide new technologies for us to interact with; they will significantly affect our interactions with each other. Their communication technologies add to the means by which people in shared and different locations can interact, beyond email, telephone, letters, and so forth. A key element of these environments is that people can interact with each other through a hybrid mix of technologies and interaction devices, including multi-media and multi-modal technologies. The pace of interaction is likely to be closer to that of spoken conversation than a written exchange of letters; people may see and hear enough of each other to become aware of each other's ongoing context and activity. How do we go about understanding these different patterns of communication, and begin to describe and reason about such systems in use? Moreover, how will ubiquitous computing environments affect the existing frameworks and understandings that depend upon shared physical spaces, given that the communication technologies change many properties of these spaces?

Interaction Techniques

We have to explore a range of new techniques that support interaction with and through diverse new devices and sensing technologies. These include gesture-based approaches exploiting movement in relation to surfaces and artefacts, haptic approaches exploiting the physical manipulation of artefacts, and speech-based interfaces. We need to support interaction and collaboration at multiple scales, from small–scale portable devices through to large–scale interfaces that are designed to support public interaction. As new interactive technologies and materials emerge we need to consider their effect on people's interaction. We need to consider how interaction techniques scale up to support the combined use of multiple interactive elements. We need to not only explore direct and engaged interaction, but also indirect interaction whereby sensors detect and interpret our actions to drive applications. The key challenge is how to admit a diverse and growing set of interactive techniques, while ensuring that the world we inhabit remains coherent.

Theories of Interaction

We need to develop conceptual frameworks for evaluating, describing and understanding interaction with and through ubiquitous computing environments.. This requires a broadening of our traditional interdisciplinary frameworks to admit the influence of ubiquitous computing. This implies that we should treat ubiquitous computing as part of language and culture, and opens up powerful associations with other disciplines that handle activity, space and structure. Existing disciplines have developed frameworks useful for design or explanation, and researchers have already drawn upon these disciplines in order to discuss, shape and predict the use of ubiquitous computing. Philosophy of language and interpretation, semiology, linguistics, activity theory,

situated action, distributed cognition, ethnographic studies, architecture and urban design theory have become important resources for ubiquitous computing. The challenge is to develop these explorations, in order to provide conceptual and theoretical tools for understanding activity in ubiquitous computing, and relate them to existing approaches in Human Computer Interaction (HCI) and Computer Supported Cooperative Work (CSCW).

2.2 Designing Environments

Ubiquitous systems present a number of design challenges. We must cater for new interaction techniques and possibilities, be aware of new technological arrangements, be sensitive to the needs a wider community of users, and support different forms of access and interaction. We must also develop new ways of assessing the effectiveness and value of these environments.

Design Approaches

The diversity of ubiquitous environments requires us to reconsider our approach to design. For example, the physical characteristics of devices have major implications for the design of their human interfaces. Inevitably, interaction design has dealt with the properties of the device as part of the human-system interface. As ubiquitous environments encompass an increasing range of mobile, fixed and embedded devices, each very different from the desktop computer, the need becomes stronger to understand when and how to decouple interaction design from the physical characteristics of the device, abstracting over them and allowing flexible deployment in a seamless way, and when and how to focus on the physical characteristics of devices so as to tailor interaction to them and exploit their properties in a more 'seamful' way. These complementary approaches are only starting to emerge as technologies mature. Many draw upon previous approaches within HCI and CSCW and augment these with approaches from other disciplines, including those focusing on the built environment. Existing explorations tend to focus on small scale experiences for a limited number of users, employing a variety of iterative prototyping approaches. However, as ubiquitous environments grow in size and the diversity of users, it becomes more important to develop approaches that cope with large–scale design.

Design for Diverse and Evolving User Needs

Ubiquitous computing environments broaden the relationship between users and digital technologies, and thus require us to change the ways in which we design for user needs. Rather than focusing on meeting the needs of one particular user community, we seek to develop tools and facilities that can be used by a broad population for a number of purposes. Handling these multiple and potentially conflicting needs represents a major part of the challenge of uncovering people's needs and designing to meet them. New approaches to requirements development are needed that are sensitive to the diversity of users. New approaches to design are needed that support people who need to adapt their environment. These approaches must reflect how use and technology co-evolve. At present we lack the core principles of how to express and support the diverse and evolving nature of people's needs.

Richer Usability Principles and Measures

Usability has been a key driver for human computer interaction. Research has often sought to describe the effectiveness of an interactive device or the outcome of the interaction in terms of usability principles. Identifying these principles has become increasingly difficult, in part because of our growing understanding of the ways that technology is embedded in and dependent on socio-technical arrangements. Key to this is the development of usability measures that reflect successful design. Issues of what constitutes success, and whose success we are talking about, have become more apparent and more problematic for ubiquitous computing. People have different ideas about what success is, and a priori criteria may become inappropriate as circumstances change, experience develops, and new ways are found to fit a technology into people's lives and situations. As ubiquitous computing technologies become embedded in a wider range of contexts, their use and meaning are continually articulated and developed, with 'objective' or 'authoritative' statements about success and utility being part of this process. This demands a new generation of usability measures that are sensitive to the situated nature of these technologies.

New Forms of Evaluation

Ubiquitous computing environments raise significant research questions about our assessment approaches. Indeed, there is a tension between different styles of analysis and evaluation of ubiquitous computing. On one side we have the exploratory and qualitative assessment often favoured by ethnographers, and on the other is the hypothesis-driven and quantitative approach often favoured by technologists and cognitive psychologists. We should not narrowly focus on either extreme. Future work should seek to enhance each with aspects of the other, in hybrid approaches, accommodating the various ways of understanding people's activity. For example, we can link ethnography-based observational techniques, often based on video and audio recordings, with analysis of

logs of transmitted data and system use, to gain understanding. When examining a system one might search for data in a system log, or analyse logged data statistically; this might highlight particular key events – which might be looked at in greater detail in the corresponding video. Similarly, one might watch through videos from start to finish, noting times of interest, and then locate systemic data and generate maps or statistics. Supporting such shifts between ways of interpreting an event may require new systems for synchronisation and analysis of data of various sorts, but also collaboration between specialists representing different 'schools of thought'.

2.3 Social, Business and Ethical Issues

The deployment and use of ubiquitous computing technologies that pervades our everyday lives will have significant social, business and ethical impact. Understanding this impact and responding to the challenges and opportunities it presents will be essential for the development of ubiquitous computing. This understanding will involve a broad range of disciplines and will ask questions of future regulatory bodies and public policy.

Privacy, Trust and Accountability

Given the use of observation and tracking technologies in ubiquitous computing environments, privacy and trust become a central concern both for those who build these environments and those who use them. If the general public is to entrust an environment with personal information, they must first consider it trustworthy. The development of security mechanisms by those tackling ubiquitous computing from an engineering perspective needs to be complemented by a user–centred understanding of the issues of trust and privacy that emerge from the use of environments in practice. Potential users of ubiquitous computing have already expressed concerns with issues around trust, whether trust in the infrastructure and its ability to register significant events, trust in secure access to personal and sensitive material or trust in its dependability. We need to understand more clearly how trust and privacy are perceived, and the features of the environment that encourage and undermine these perceptions. What guarantees do people require and how should these be provided by the infrastructure, by our use of the technologies and even by regulatory bodies? A key issue is the accountability of ubiquitous computing systems; how do we design for the different levels of accountability and responsibility needed to promote trust?

Deployment, Sustainability and Environmental Impact

As ubiquitous computing technologies mature we need to consider the means through which these infrastructures will become an everyday part of the world we live in. How do we ensure that they are sustainable and permanent? We need to develop appropriate socio–economic models of deployment. Who will fund the infrastructure costs required to support ubiquitous computing everywhere and should the charging models be similar to that for mobile telecommunications? This must involve researchers from a business and economics background, to uncover the core principles of deployment. A key question in ensuring sustainability is how to reduce the management costs for these complex systems. We need to explore how to involve the various stakeholders in the management of these infrastructures, and must devise appropriate representations to help them in this task. Similarly, researchers from environmental and ecology disciplines must study the impact of ubiquitous computing technology on our natural environment and resources, and on our health. What is the environmental impact of miniaturised disposable electronic devices ('smart dust') that may be sprinkled in rivers and fields, on animals, or embedded in clothing and packaging? What impact will the increased proliferation of wireless communication have on health?

Ethics and Research

There is a difficult balance to strike between acquiring information about users that may improve the effectiveness of ubiquitous computing systems, and protecting ordinary citizens from unscrupulous actions. If the systems are designed too tightly, they become rigid and difficult to use; if too loosely, they become vulnerable to abuse and may incite negative media reports and public opinion. A key issue is to develop ethical principles surrounding research into ubiquitous computing when it involves users. What is ethical practice in this sensitive domain, what support can be provided for researchers, and how can these be tied to reassurance for concerned parties? We may need practical tools for researchers, to help them plan and conduct research projects that require the use of personal data according to good ethical practice. As part of this work we need to understand the ethical barriers in the use of ubiquitous computing environments. We must identify issues that may severely constrain or even obstruct the successful use of ubiquitous computing technologies, and bring such issues to the attention of policy makers. A key part of this work is to engage the general public in ethical debate on ubiquitous computing technologies. We need to find ways to raise awareness of the technologies, in order to foster engagement and debate.

3 The Engineering Perspective

When considered from the engineering perspective, the dominant research questions focus on the mechanisms and techniques for designing and constructing ubiquitous computing systems (UCSs). They have to cater for mobility of people, vehicles or trains containing embedded systems interacting with fixed systems in buildings, roads or in the environment. There are difficult research challenges in engineering ubiquitous systems to provide the required context dependent behaviour with security and dependability while considering the constraints of mobility, power and limited device capabilities.

3.1 Physical constraints and context

The physical constraints and context dominate the design and construction of ubiquitous computing systems.

Size and Power

Microminiaturisation is needed for devices to be implanted in people for healthcare applications or integrated into clothing and everyday artefacts. The vision of millimetre size devices with sensing, processing and communication capabilities is far from reality. These devices must be capable of surviving in harsh environments – jungles, rivers, on buildings, in the bloodstream etc. They often cannot be wired into power supplies. Even within buildings, providing wired power to hundreds or thousands of sensing devices is not practical and neither is changing batteries. Use of solar cells, fuel cells, heat converters, motion converters etc. may all be possible. The challenge is to design very low-powered devices, replacing the traditional emphasis on faster chips. This requires optimising the design of circuits, communications, operating systems and languages to avoid squandering power.

Wireless Communications

The fact that ubiquitous computing devices will be incorporated in mobile entities such as people and vehicles, as well as the impracticality of connecting thousands of devices to a wired infrastructure means wireless communication is essential. However wireless communications often require far more power than other forms of processing. There are currently many different wireless communication standards used in various environments – cellular phones, Bluetooth, Wi-Fi, Wi-Max, Zigbee etc., each with different tariffs. Mobile users may need to switch seamlessly between these different types of communication, depending on their current context or to manage costs or security and privacy concerns.

Context

One of the key aspects of ubiquitous systems is that they are context-aware in that they need to know about and should be able to react and adapt to their surroundings, including their own location, the presence and activity of neighbouring devices and agents (human and digital), and the available resources such as communications, power and processing capability. Interaction modes can switch between voice and visual depending on user activity. Rich models of mobility and context in the physical world are needed to support the design of UCSs. A UCS cannot assist users without determining their activity. Low-powered sensor technologies, techniques for fusion of data from multiple sensors, and techniques for inference from sensed data and user input are essential. Analysis of recent history and statistical patterns of usage, interactions and context can also be used to infer both current and future context. This work needs to be undertaken in partnership with the experience perspective which seeks to understand the context is terms of human activities.

3.2 System Structure

Despite physical constraints, the structure and mobility of the physical entities in a UCS will be complex. Even more complex will be the virtual structure of software entities, and how this virtual structure interacts with the physical structures of the UCS and with the world within which the UCS is set.

Software Mobility

Software entities may be dynamic and ‘virtual,’ but they are instantiated in physical devices, which are located and linked in physical space. However software entities can also be mobile by moving between devices over communication links, in order to facilitate adaptation of the UCS to current context, new requirements, failures or security attacks. The logical interconnection structure of software entities will be very different from how it is realised in terms of physical devices and physical links, and will be very much more dynamic. A significant part of good design will involve principles for software mobility, changing system structure and physical location for reasons such as efficiency, intelligibility, security and dependability. New approaches are needed to provide flexible and adaptable software and hardware, for both mobile devices and mobile software entities

Self-configuration

The scale of UCSs demands 'autonomic' (self-organising, self-managing, self-healing) systems of entities to simplify installation and evolution of devices, services and applications. It will not be feasible for non-technical users to install software and configure potentially many thousands of devices in their homes, cars and workplaces. As soon as a system can reconfigure itself, the problem arises of how a migrating agent – whether human or digital – can discover the resources and services it needs in its new location. Related to self-configuration is a renewed interest in self-stabilising algorithms, in systems that tolerate a steady set of failures or disturbances, but are always evolving towards a stable and hopefully also correct or useful configuration. This can apply on many timescales. However, there will be times when the system may have to change from full to partial autonomy, e.g. to invite human intervention in exceptional circumstances, and when people wish to over-ride the autonomic decisions. A key design challenge is to enable shifts between different degrees of autonomy, based on an understanding of whether, when and how to involve people in adapting behaviour. This requires links with those seeking to understand interaction from the experience perspective.

Hierarchy and Composition

Hierarchies of modules, procedures and data structure have been the rule in traditional software. They are a valuable means of abstraction to support design, analysis and implementation. There are clear advantages in considering a cell, such as a body-area network of devices monitoring the health of a patient, as a self-managing, autonomous unit for the purposes of determining context, self-configuration and self-healing. However this autonomous cell needs to interact with cells belonging to medics and may use services in the infrastructure for communication and logging of patient data. In some situations a cell may be treated as a component that is incorporated, controlled and managed within another cell, for example the body network monitoring a patient in a postoperative intensive-care ward. In other applications, cells may interact as independent autonomous peers to collaborate and cooperate e.g. visitors in a museum sharing experiences. Thus the engineering challenge seeks design principles that allow interactions and associations based on hierarchy, composition and ad-hoc peer-to peer collaborations as people take part in spontaneous mobile interactions.

3.3 Security and Dependability

As ubiquitous computing systems are used for critical applications, such as healthcare monitoring or controlling vehicles on motorways, issues such as security and dependability become more important.

Security, Trust and Privacy

UCSs are very dependent on wireless communication which is intrinsically broadcast and hence easily monitored. Messages routed via unknown intermediate nodes may be susceptible to confidentiality or modification attacks. Nodes can be bombarded with messages in order to deplete battery power. Security is thus a critical concern in such a potentially hostile environment, particularly for applications involving financial transactions or healthcare monitoring. A ubiquitous application may involve collaborations between ad hoc groups of entities. It may require migration or downloading of code, and may involve people moving and changing the system configuration. New encounters occur, and there are complex issues in knowing what entities to trust. Does a server trust an agent enough to allocate processing resource to it? Does a device trust a neighbour to send message packets for onward routing? (The latter could be a 'denial of service attack,' aiming to deplete the device's battery.) Does a device trust a person asking to input data or reconfigure it? Does a human using the UCS trust a host, a service, a device, or another human communicating and collaborating through the system? Based upon predefined trust, recommendations, risk evaluation and analysis of past interactions, an entity may derive new trust metrics and authorisation policies for what access it will permit to its resources, what services it should refrain from using, or what security mechanisms (such as encryption) to use. It may need to validate credentials without access to network infrastructure and certification authorities. Context aware security mechanisms are needed to adapt to current situation such as in a meeting, where you would want to share resources with colleagues, or in the street, where no access should be provided to strangers. Defining mechanisms for constrained devices to detect and adapt to denial of service attacks is also a challenge.

Ubiquitous systems are generally context-aware in that they detect and monitor the current context of users. This may include information on location, activity, who one is with, and medical condition. If this information is made available to unauthorised third parties it can result in serious violations of privacy. How the information is actually used by authorised users may also raise privacy concerns. These privacy concerns raise both technical and social issues of how to control access to information and how to allow users to control whether they are tracked and when they may wish to be anonymous. This raises issues of a user's perception of trust and privacy which relates to the user experience perspective.

Dependability

As UCSs become more widespread we will become more dependent on them and reliant on them working correctly. A health-monitoring system which facilitates early release of a patient from hospital has life-dependency implications as do systems relating to controlling vehicles on motorways. The overall complexity of such systems are such that one has to assume that their software, hardware and communications will suffer from faults – these may be accidental, or the result of deliberate attempts to subvert or damage the system. Self-healing capabilities are needed to mask or recover from the effects of these faults; particularly as such systems become more integrated into daily life. Although there are techniques for developing secure and dependable systems, most large-scale computing applications have proved to be notoriously unreliable and insecure. There are also problems of being able to identify all the requirements for such complex systems or foresee all situations and circumstances in which they will be used. Making systems which can adapt to many unforeseen requirements and situations is beyond our current engineering capabilities.

Exceptions

If an entity cannot achieve its purpose, its best recourse may be to raise an exception with – or report failure to – its superior in a hierarchy, if it has one. If it is reflective, it may provide enough information about what happened to allow the superior to perform remedial intervention; otherwise the superior may have to 'pass the buck' upwards. Thus hierarchies may be useful for failure management. On the other hand, a self-managed entity without a 'boss' may have to deal with the exception itself by instigating self-healing strategies. In the worst case, if the exception cannot be dealt with, then the entity must fail-safe, e.g. switch itself off. New techniques are needed to allow effective management of failures, both with and without hierarchies, with very dynamic structures of entities.

We cannot expect systems to always configure themselves autonomously. People may need to be involved in replacing failed components, tailoring system behaviour to meet specific requirements, as part of the co-adaptation of system structure and system use. Actual studies of how people use, deploy, maintain and repair a large UCS are needed to inform the engineering of these systems.

3.4 Information flow

A UCS could include millions of sensors generating huge quantities of information. It has to maintain information about resources, agents and purposes. There will be very dynamic information being generated about current context and state of mobile entities. It will be used to transfer media streams to and from users. This will introduces issues relating to network design, information processing and provenance which we discuss below.

Network Design

The engineering of fixed ('wireline') data, voice and video networks is well-understood. Techniques exist for estimating the behaviour and combination of sources in a traffic matrix, and their impact on each other for a given network topology of switches, queuing systems, and link capacities. These techniques allow the design of networks for a given performance goal, and allow one to construct dynamic routing strategies to adapt to varying traffic conditions, link status and error conditions. In a UCS we find no sharp boundary between its communications system and the application. Network technology, networking algorithms and protocol design must be seen as problems intricately interwoven with the whole design of a UCS. Both information and entities may move as a significant portion of the communication system of a UCS will be wireless and mobile. In such networks, there is very little experience so far of the inherent performance variability due to mobility. Link capacity, delay and loss can also vary due to environmental factors. To add complexity: many wireless network technologies use adaptive techniques for modulation, coding (e.g. CDMA) and routing (e.g. multipath and mesh wireless networks, and even hybrid radios). Recent work suggests that far more cross-layer optimisation is needed to make these systems deliver reliable performance. Thus the algorithms running 'on top of' the network are no longer independent of the network. Approaches such as network coding are proposed to combine data from multiple sources. Together with techniques such as distributed interest-based filtering, these yield far more complex source behaviours and traffic matrices; all this, with a dynamic adaptive topology.

Information Overload and Relevance

A vast number of sensors can generate an equally vast quantity of data. Some of the sensors will be faulty and so some of the information will be incorrect. To avoid overwhelming traffic, this data should be filtered, aggregated, reduced etc. as close to source as possible. It may be better to send queries to sensor networks rather than assume they send information to databases. Sensors may have to be programmed to look for specific events, and may use histories of sensor data as well as predictions and models to adaptively handle changes in context and requirements, as well as failures.

Users may easily be overloaded with information being generated from many different sources. Systems for filtering events based on interest or relevance to the consumer are needed. This is already a feature of multi-player mixed reality games, where so-called 'area-of-interest' management is an important component of the scalability of a game, both for systems and for human users. Software entities in the ubiquitous computing paradigm will need techniques for controlling their context and communication, analogous to humans finding private and secluded places.

Information Provenance

Information is increasingly delivered to us indirectly, via a complex set of unfamiliar channels. We need to be told the authoritative source of information, i.e. its provenance. In ubiquitous systems there will be vast numbers of different sources of information as indicated above – sensor data, context information, information explicitly communicated by people such as voice, text messages and video. Not only humans but also other entities will need to assess the reliability of data received. The problem is further compounded in sensor networks which filter and aggregate information as it is being relayed through the network.

4 The Theory Perspective

The scale and complexity of ubiquitous computing systems make them a formidable object of scientific investigation, one that we simply cannot neglect. Ubiquitous systems must provide assurance in terms of dependability, predictability, and much besides. Understanding and analysing the structure and dynamic adaptation of these heterogeneous systems will be a challenge. This assurance and analysis, based upon rigorous models, will form the foundation for design and engineering of ubiquitous systems and the experience of their users. Moreover properties such as correctness, utility and dependability become harder both to define and to design for in ubiquitous computing. For all these purposes computer scientists need to harness existing theories, and formulate new ones. These theories will come not only from computer science and mathematics, but also from other fields such as linguistics and sociology.

In this section, with no pretence of completeness, we comment on central concerns that theories for ubiquitous computing must address. We consider: how the elements of a system will be organised to allow their interaction; how a system may be aware of its context and reconfigure itself; how to understand the way information flows; and finally how we may come to gain confidence in the performance of ubiquitous systems.

4.1 Structure and interaction

As billions of entities may be involved, we cannot understand their behaviour unless we understand how they are organised. Moreover, the behaviour of a system just consists of the interactions of its members. Structure and organisation lie at the heart of our challenge.

Hierarchy and complexity

The natural way to manage the complexity of a population of entities is to arrange them in a hierarchy. This principle has guided the design of mathematical process calculi over the last quarter-century. Entities exist at many orders of magnitude; a complex entity is a collection of simpler ones, and its properties are determined by properties of its members. One such property its capacity to interact with other complex entities; their interactions are defined by the capacity of their respective member entities to interact. The agents of agent technologies are an example of such entities.

Open systems

It follows from this approach that such an agent has the capacity to interact with its environment, which may vary unpredictably; an agent is an open system. In a ubiquitous system the openness of its subsystems is an essential property; for example, humans or other agents may vary the context in which an agent exists, and the definition of the agent determines how it will perform in any context.

Uncertainty

Changing context is not the only source of variation in an agent's behaviour. It is common practice, both with process calculi and in agent technologies, to admit non-determinism in an agent's behaviour. Even if agents' behaviour is deterministic, non-determinism in a model of agents may arise from lack of knowledge about them.

There is a strong motivation for refining a model of agents by assigning probabilities to their possible behaviours; that is, by working with a stochastic model. A stochastic element arises in many aspects of ubiquitous computing; consider, for example, probability for being at a particular location based on previous location, or for favouring an interaction with agent A over one with agent B. A great advantage is that it admits

simulations that sample the space of possible behaviours. Indeed, stochastic simulations have already been used in applying process calculi to biological phenomena.

Hybrid Models

Much input into ubiquitous devices will be from sensors that generate analogue or continuous data such as temperature or sound, as well as discrete information. Models that capture both continuous and discrete dynamics, called hybrid systems, have been studied but are limited. For example, timed automata only allow time as a continuous variable, but cannot model continuous space; yet both are needed to facilitate practical application, e.g. to a driver-less traffic system. The challenge is to obtain appropriate representations of state sets generated not only by discrete dynamics, but also by continuous dynamics governed by differential equations.

Elementary models for hybrid systems already exist, and we expect considerable advance in the near future. A strong link is needed with work in control theory, which has long studied continuous systems.

4.2 Configuration and awareness

Neither the hardware nor the software entities in a ubiquitous system will remain fixed. This poses a double challenge. The first is to extend existing models, such as process calculi, to accommodate space and mobility; promising candidates already exist, but difficulties of analysis still remain. The second new challenge is to accommodate goals and purpose for UCSs' mobile agents, and the effects on them of a constantly changing context; the added difficulty here is how to organise and analyse the entities' awareness of, and reaction to, such change.

Space and mobility

Since space and mobility are endemic in UCSs, the recent trend in process calculi that treat these concepts directly, such as Mobile Ambients, is likely to be exploited. This is the beginning of what may be called structural discrete dynamics; developments of it are certainly needed to accommodate the structures of a UCS. The extra challenge of adding the continuum, enabling smooth incorporation of differential equations, will lead to further innovation.

Computing has a successful tradition of designing logics that cater for particular computing phenomena (for example Hoare logic for state change, temporal logic for possible futures). Many general results have come from relating such logics with algebraic process calculi and with the models of agent technologies. Mobile UCSs will require spatial logics, to match temporal logics; for example, consider an assertion "no agent has requested this resource at this place before" which combines the spatial with the temporal.

Resources

Mobile agents in a UCS expect to acquire resources as necessary from the environments (e.g. the other agents) that they visit. Access to resources can be controlled in terms of boundary crossing in a current spatial model such as Mobile Ambients, augmented by a type discipline. Agents requiring resources may be either hardware (e.g. sensors requiring power) or software (e.g. agents requiring memory allocation). Methods to control allocation of resources, based upon logics and types, are already under investigation. Availability of resource will also be part of the information that flows in a system (see below). An important issue is whether and how resource should be paid for. This may involve negotiation, a capability of autonomous agents that has attracted considerable research.

Context and reflectivity

Entities become aware of their context through interaction with sensors or other entities. Attributes of an entity's environment, thus discovered and continually updated, will affect the entity's behaviour in many ways. In particular, it will involve self-(re)organisation of structure as well as of goals and operational strategies. It may for example enable a population of sensors to act as a team and record new sets of events, or an agent to report the history that led up to a failure. One particular use of the attributes, called reflectivity, is particularly challenging for theoretical models. An agent is called reflective when it is able to build its own model of the behaviour of the system in which it works; the behaviour it models may range over both (past) time and space. Thus a challenge for theories, and for programming languages derived from them, is to model and analyse systems that also model themselves.

4.3 Information flow

UCSs will involve huge quantities of information flowing from sensors, to and from mobile users as they interact with their environment, play games or are monitored for medical conditions. The engineering challenges for information flow have been discussed in Section3.4. Some theoretical concerns are as follows.

Network analysis

The engineering of fixed topology networks is well-understood. It often rests upon theories: for example, techniques have been based in large deviation theory to describe self-similar sources, in information theory to allow for traffic matrix estimation, and in control theory and computer science to understand the regime of stable performance. Much of this theory must be reworked for a UCS, in the presence of mobility. Patterns of mobility must be modelled, in order to repeat the successes for fixed networks. Indeed, it is no longer clear that a "network" is a distinguishable subsystem of a UCS; thus the traditional networking concerns of throughput, robustness, latency, recovery, etc., have to be addressed for the UCS as a whole.

Provenance, archive and annotation

The need to query distributed data sources gave rise to an early practical example of mobile agents. Nowadays, distributed data is typically semi-structured. Adapting relational database theory and language design to semi-structured data is a non-trivial task, creating new research challenges. For example, copying and mutation highlight aspects of data that were previously unproblematic. Copying seldom preserves the provenance (i.e. the pedigree) of a datum; how then do we assess its credibility? Frequent change of databases endangers the archival requirement; how can we ensure that the sources we cite remain inviolate? An associated issue is annotation, the overlaying of data with extra detail or comment. How does this affect querying a database?

These problems require close collaboration between software engineering and theoretical models of data. A rigorous approach is all the more important in ubiquitous computing, where not only humans but also autonomous agents will both supply and access data. We also need to merge research on semi-structured data and process models, since there is no sharp distinction between mobile data and mobile processes.

4.4 Dependability

All the above topics support the understanding, and therefore the better design, of UCSs. This means not only that they will supply the services intended, but also that our theories provide specific reasons why we can depend upon our designs.

In this section we examine important topics relevant to the dependability of UCSs; we then propose that our UCS theory will rely upon an aggregate of specific models, together with a rigorous notion of realisation of one model by another. Finally, we mention the importance of tools to support all theoretical analysis of UCSs.

Trust

Humans wish to trust UCSs. But trust between autonomous agents will also be an important ingredient; certain agents will be responsible for allowing others to cross boundaries, or for allocating resources to them. A discipline of trust between agents will only be effective if it is rigorously defined. Logics and languages have been proposed for expressing such disciplines, in terms of notions such as belief and authority, together with information about past interactions with the agents in question.

Such disciplines need experimental assessment as well as rigorous analysis. It is no good having a system where trust is never betrayed, if the system never does any work or if users find the system unworkable. An early objective is therefore to design experimental scenarios to assess the viability of trust disciplines.

Protocols and security

Much information is transmitted under protocols, which will be complex in order to ensure reliable transmission, efficient network use and security. But many protocols in heavy use have serious defects. Also, new problems arise with mobility; it is not easy to design —and still harder to analyse— protocols that attempt to maintain collaboration in groups whose entities join, leave or temporarily go out of wireless range. The expected orders-of-magnitude increase in the size of UCSs will lead to greater complexity of protocols. There will also be greater variety; for example, protocols for interaction between autonomous agents are likely to differ those employed by (say) itinerant database updates and queries.

A special class of protocols is associated with security of information. Over the past two decades much analysis of security protocols has been carried out using special purpose logics and calculi. It is likely that this concern will be intensified in UCSs, especially if interactions hitherto performed by humans are delegated to autonomous agents acting on their behalf.

Multi-level modelling.

No single model will suffice for all that we need to analyse in a UCS. How do we bring several models to bear on the same system? Clearly they must be consistent in some way. An elementary example is a program written in C, and a specification in Hoare logic. The former is a concrete model, the latter an abstract one, and they are consistent if the program satisfies the specification.

Levelling appears in many guises. For example, a system may be described in a process calculus, and a desired property of the system expressed in a logic of trust. We have to prove these consistent, e.g. that agent A never allocates resource to agent B without first receiving evidence for trusting B. At the next level down, the same process calculus description may be realised in Java; we have to verify that this realisation is correct. Such examples will multiply as we seek to define UCSs in ways that both users and implementers understand.

Thus we cannot expect a single homogeneous model suitable for all UCSs, or even for a single UCS considered at different levels of abstraction. Yet the aggregate of models for all UCSs must have integrity, allowing us to deduce properties at a higher level from a model at a lower level. This integrity amounts to defining how a higher-level model is realised by a lower-level one. Success of the theoretical analysis of a UCS will be measured by the integrity among all its models.

Languages.

A vehicle for much of UCS design will be programming languages. These will be various; for example, a language to programme movement in a sentient building has quite different requirements for one for expressing negotiation among intelligent agents. The only way we shall achieve integrity among these languages, as well as a rigorous understanding of each one, is that each language is derived from (or is an executable subset of) a specific model; then the integrity of languages is inherited from the integrity of models.

Of course, theories and calculi have always influenced the design of programming languages. But this influence was partial; ad hoc elements have always entered language designs, often to the detriment of their soundness. UCSs will place great reliance on the soundness of language, since many programs are likely to be inaccessible to correction. Thus a key element of our theoretical challenge is to ensure that a relevant theory lies fully behind every language used.

Model-checking and theorem-proving.

Many software tools must support theoretical analyses of UCSs. One of these, model-checking, is concerned with verifying assertions about a system by automatic traversal of its state-space. Model-checking has been successfully applied to assertions expressed in specific logics, such as temporal logic.

Recent attention has been devoted to infinite-state and probabilistic model-checking, both important for UCSs. We also need to extend it to handle mobile systems, and also to methods that are scalable and compositional — since UCSs will be huge.

More general tools for automated deduction can analyse a wider class of assertions, and admit human assistance. These theorem-proving tools have made great strides in the past few years. In particular, they have been used to verify or detect flaws in complex real-life protocols for data transfer, such as TCP and UDP, and a variety of security protocols.

We have reached the stage at which a significant part of the effort in deploying theoretical models should be devoted to automating their analytical powers, with tools such as we have briefly described.

This completes our brief survey of theories needed to support UCSs, which form a greater challenge to understanding than any hitherto known software systems.

5 Addressing the Grand Challenge

In Section 1 we declared our strategy to tackle ubiquitous computing from three different perspectives by developing scientific theory, engineering principles and interaction and design methods for global ubiquitous computing in a strongly iterative manner. We have suggested distinct goals from a theoretical, systems and experience perspective, and devoted a section to each of them and commented on aspects important to each. It would be pleasant to be able to propose, now, a fifteen-year research programme that works towards the goals of the Challenge and interleaves these different perspectives. But it is in the nature of a Grand Challenge that we cannot expect immediately to define a path to its goals. The purpose of our goals is to focus intentions, so that exploratory research can be marshalled towards defining a path.

Research in many of the topics relevant to our Grand Challenge is already proceeding, and will indeed proceed independently of it. Within the UK we have vibrant internationally leading research communities in each of the

key areas. These previously disparate communities are coming together to focus on many of the core interdisciplinary issues highlighted in this document, and substantive links between our three perspectives have begun to emerge. These links will grow and strengthen as we develop concepts, theories and frameworks to allow us to tackle these issues.

The Challenge also evokes ideas for exploratory projects, especially projects involving new collaborations, whose aims may be modest in comparison to our three goals but which lie on the path to achieving those goals. We would refer to these as ***foothill projects***. We anticipate these projects being proposed and funded in the normal way. The Challenge is in no sense directive, but will require us to coordinate discussion among projects via the UK-UbiNet network and its workshops, and by any other available means. ***Annexe I*** provides several illustrative outline proposals for foothill projects. The topics of the initial batch of outlined proposals include:

- Analysing movement in a sentient environment
- Automating the highway
- Model-checking for ubiquity
- Rigorous protocol design
- Ubiquitous computing and the urban environment
- Ubiquitous healthcare.

We expect ***Annexe I*** also to evolve; other outlines will be added, existing outlines will be refined, and the outlines will be annotated with references to any relevant projects that are currently running, or are mounted in the future. This Annexe may serve as a directory for the work that constitutes the first phase in addressing our Grand Challenge.

If the path outlined in this manifesto is followed, within a few years the goals put forward will already have evoked research on a broader and more coherent front than would otherwise have been undertaken. It will have induced mutual awareness and connections between previously disjoint research communities within computing. This alone will have justified posing the goals of our Challenge.

Beyond this lies the possibility that, by means of the experience of foothill projects, the emerging relations between the different element of the challenge and the discussion coordinated around them, within (say) five years we are able to identify results, problems, themes and methods that lead to proposals for one or more ten-year projects with clear milestones, aiming with greater confidence at the Challenge's goals. The definition of these projects will amount to a roadmap to the goals; such projects will constitute a highly focussed and coordinated attack on the summit that these goals represent.

The proposals for these larger projects will be able to define what counts as failure, and what counts as success, much more clearly than we can at present. They may indeed fail, by their own definition. Whether they succeed or fail, within fifteen years will exist a research community that knits design and science more closely than ever before in computing, facing greater opportunities and dangers than any hitherto: the pervasion of computing into almost every aspect of human life.

ANNEX I Foothill Projects

In this section we outline some initial ideas for exploratory foothill projects that can help to create a platform for a coordinated attack on the Grand Challenge. Each outline describes a combination of current work with follow-on goals for immediate future research. In this way it involves all three of the elements that the manifesto suggested should be involved in a foothill project: design principles, theories and applications. The order of the examples is alphabetical, and has no other significance. These outlines have not, in general, been written by people who are submitting such a proposal. Each one should be regarded as defining a topic for several possible projects, which may or may not be coordinated. Readers are encouraged to discuss these topics, and others possible, on the Grand Challenge mailing list http://mailman.doc.ic.ac.uk/mailman/listinfo.cgi/ubigc. As a result of discussion this Annexe is expected to grow, both in the number and variety of topics and in the detail of each proposal.

Information on these projects and future one can be found on the Grand Challenges web site

http://www-dse.doc.ic.ac.uk/Projects/UbiNet/GC/manifesto.html

Analysing Movement in a Sentient Environment

Lars Birkedal, Robin Milner

Pervasive computing will involve movement of agents in environments equipped with sensors that may also move. Many such mobile systems with particular purposes have been realised in practical experiment, or simulated. One example of a purpose is "sentient computing", defined by Hopper as "Using sensor and resource status data to maintain a model of the world which is shared between users and applications". (A. Hopper: Sentient Computing, Phil. Trans Roy. Soc. A, 358(1773) pp 2349-2358, 2000.)

Other purposes are possible; for example, sensors (fixed or mobile) may be programmed to collaborate in guiding an agent to a goal. The variety of applications is large, and suggests that a first step is to study and analyse locality and movement per se, without prejudice towards a particular application.

The problem addressed in this outline proposal is: What is a fruitful conceptual framework in which to express a variety of rules of motion, allowing systems to be programmed conveniently, simulated (with the help of stochastic information in the rules), and analysed rigorously? An example of rigorous analysis would be that certain invariants are maintained (or maintained with certain probability) by all behaviour allowed by the rules.

To underpin both description and analysis, it is convenient to use a spatial model which can represent both discrete and continuous space, and movement in such space, as well as the usual forms of data and processes involved in traditional computing. Examples of such models are suggested by calculi of interactive computing, such as the calculus of mobile ambients or the pi-calculus. But the aim of the project is not just to design a calculus; it is to derive from the calculus a programming language, and define a programming methodology, so that the language may be used and evaluated by people whose primary interest is in applications.

The ultimate goal of the project is to unify theory and practice in this basic but challenging facet of pervasive computing. One approach, involving bigraphical systems, is already under way at the IT University of Copenhagen. An example of a system that may convince users, suggested by the group at ITU, is to model (and program) a "reflective" building: one equipped with sensors, which continually transmit data of the building's occupancy to a monitor that maintains a data structure which faithfully records the occupancy.

Automating the Highway

Jon Crowcroft

Monitoring and control of private vehicles on the public highway is high on the political agenda; this is because it is becoming feasible, and may be desirable for at least two reasons: first, from the economic perspective, it may achieve more efficient use of road resources; second, from the safety perspective, it may achieve a significant drop in injury and death on the roads. Various prototypes exist, and various projects are current. Many technologies interact, and there are numerous legal, political and economic stakeholders. We propose a foothill project to study monitoring and control with particular concern for efficiency and safety, in the context of ubiquitous systems for transport. For efficiency (of road use) the monitoring and control may be either distributed or centralised, or a combination of the two. In a distributed system the car receives information from navigation systems and roadside monitors concerning routes, conditions and prices; it (or its driver) then makes

a decision and pays. On the other hand a centralised system, such as the London congestion-charging scheme, depends entirely on a network of roadside monitors, recording data about vehicles, drivers and journeys on a central database used as the basis for billing.

To improve safety, there a spectrum of possible solutions from distributed to centralised systems. At the centralised extreme, `car-trains' have been proposed; vehicles joining trunk routes would be logically clumped, and controlled by a single aggregate unit. At the distributed extreme, each vehicle always chooses its own velocity, using data from on-board and remote sensors. There are many research problems; for example: What are the design spaces for distributed and/or centralised systems in the two cases? Can they be mixed, e.g. distributed for efficiency of road-use but centralised for safety? By what measures can each solution in the space be assessed for its contribution to both efficiency and safety?

In each possible design, what threats arise from neglect or malevolence? These threats may attack endanger correct technical function, or they may endanger privacy (for example, centralised records may be illegally mined to deduce driver habits).

Success in addressing these problems will involve a variety of theoretical or simulation models of distributed and mobile processes; and will prompt the further development of such models.

A longer paper addressing these issues is also available
http://www-dse.doc.ic.ac.uk/Projects/UbiNet/GC/Manifesto/road.pdf

Model-checking for Ubiquity

Marta Kwiatkowska

Wireless sensor networks and body sensor networks already exist, or are planned, for many purposes. Such systems must deal with continuous streams of data, analogue and digital, and must be adaptive, fault-tolerant, dependable, context-aware and energy efficient. Above all, especially in a safety-critical situation, they must work correctly; for example, a failure to identify a dangerous pattern during heart-monitoring may lead to patient death.

Model checking is an automatic technique that can establish, via exhaustive analysis of the model of a system, whether its behaviour is correct with respect to a given specification. It relies on logics or calculi that define the state space; hence its development must be closely linked to those logics and calculi. These logics or calculi themselves have to be developed, as part of the theoretical goal of the challenge.

Following major successes in detecting genuine errors in standardised protocols, model checking techniques are now widely used in industry, e.g. for hardware verification at Intel and source code compliance checking at Microsoft. However, sensor networks raise new scientific challenges:

- Sensor networks are dynamic, adaptive and context-aware. What model checking techniques are appropriate for these infinite-state systems?
- Sensor networks are often decentralised, communication failures are frequent, and techniques such as randomisation are used for their coordination. Probabilistic model checking techniques are needed to accommodate these features.
- In a healthcare monitoring scenario patients are mobile, and this affects both power usage and reliability of communication. Stochastic models of social behaviours must be developed to analyse the effectiveness such systems.
- Monitoring scenarios involve streams of data, requiring fast analysis and response. How can we ensure the correctness of such responses?
- Quantitative model checking techniques are needed to predict the power usage of sensor networks over time and to select the best network configuration given some constraints.
- How can we ensure that the methods are scalable to realistic systems? Compositionality, abstraction and component-based techniques for ubiquity are needed.

Success in meeting these challenges will depend on working closely with designers of sensor networks, to gain an understanding of the key issues and to secure acceptance of the techniques. Some current projects are:

Design, Implementation and Adaptation of Sensor Networks through Multi-dimensional Co-design
http://gow.epsrc.ac.uk/ViewGrant.aspx?Grant=EP/C014774/1

National ICT Australia's Formal Methods programme.
http://nicta.com.au/director/research/programs/fm.cfm

Rigorous Protocol Design

Peter Sewell

Communication protocols will form a key part of any ubiquitous computing system. Traditional Internet communication is dominated by the UDP and TCP transport protocols, together with various routing protocols, above IP. These rely on properties of the existing network — relatively stable connectivity, loss dominated by router congestion, and so on — that will not hold for the variety of network technologies in ubiquitous systems. New protocols will be needed.

If these are to be predictable and robust they must be well-understood, for which new design techniques are also needed. Traditional internet protocol design is largely based on natural language specifications and interoperability testing between implementations. This often leads to unnecessary complexity and subtle flaws and implementation differences.

We therefore have an opportunity and test-bed for an integrated systems and semantics approach to protocol design. The goal of this foothill project is to establish a suite of protocols for particular GUC scenarios, developing and using rigorous design techniques for the purpose. It may build on the Netsem project, which has demonstrated a viable approach to the formal specification of existing real-world protocols, expressing their behaviour with operational semantics in the automated proof assistant HOL and developing symbolic model-checking techniques to validate the specification against captured traces.

The main challenges are:

- The large-scale systems question of broadly what protocols are required, their APIs and design principles.
- Establishing idioms for expressing detailed design that:
    - are an effective means of communication in the design and implementation teams, among those with theoretical and practical backgrounds;
    - support direct and automated conformance testing of production implementations against the protocol designs; and
    - can be refined (i.e. resolving any looseness in the specifications) to give prototype implementations that can be used for experimentation and simulation.
- Establishing higher-level (more abstract) models that are suitable for approximate (probabilistic or stochastic) reasoning and simulation, ideally with mathematically rigorous relationships to the detailed designs.
- Verifying (with automated proof and/or model-checking) properties of both detailed and high-level models.
- Deploying the protocols and gaining experience in their use.

Netsem Project: http://www.cl.cam.ac.uk/users/pes20/Netsem

HOL: http://hol.sourceforge.net/

Ubiquitous Computing and the Urban Environment

Eamonn O'Neill

In urban areas we have the greatest opportunities and the strongest demands to design and build ubiquitous systems, yet we have no fundamental theory, knowledge base, principled methods or tools for designing and building ubiquitous computing systems as integral elements of the urban landscape. We are interested in designing not just the architectural space in which people move and behave and interact but also the interaction spaces for information and services that they discover and use and which support their movements, behaviour and interactions within architectural space. To design these integrated systems, we need to extend and adapt our understanding and practice of both urban design and ubiquitous computing. We need to understand and design for people's behaviour and their relationships with urban space and ubiquitous technologies. In addition, we need to solve the technical and engineering challenges of implementing city-scale ubiquitous systems.

A systematic approach to designing the urban environment as an integrated system of physical architecture and ubiquitous technologies demands a coming together of the disciplines of Architecture and Computer Science. In a system of heterogeneous devices, diverse users and varying network provision, the design and implementation of such systems require significant advances in research and practice across a range of themes that have both human-computer interaction (HCI) and distributed systems (DS) aspects. These include context awareness; service discovery; trust, security and privacy; and the physical, psychological and social impacts of ubiquitous systems. Solving these problems is made even more complex by the challenges of scaling up from laboratory-based examples to a city-scale operational system. From the HCI perspective, developing successful city-scale systems requires significant advances in areas such as interface design, context awareness and service discovery, to help people manage the demands on their attention and make the best use of their limited ability to descry what they want or need from this new combination of physical cityscape and digital services. From the DS perspective, city-scale ubiquitous systems require a fresh approach to many of the classical DS issues such as communication, fault-tolerance and security. Classical solutions such as caching, multicasting and peer-to-peer sharing will require adaptation to take into account ubiquitous technologies, while newer approaches, such as those in autonomic computing, may offer some solutions.

A few recent and current projects have begun to explore at least some of these challenges. See for examples:

Cityware http://www.cityware.org.uk/

Communities of collocation http://www.ecs.soton.ac.uk/research/projects/collocation

Equator http://www.equator.ac.uk/

Mobile Bristol http://www.mobilebristol.com/flash.html

Shared Worlds http://www.shared-worlds.org/

Urban Atmospheres http://www.urban-atmospheres.net/

Urban Tapestries http://urbantapestries.net/

Ubiquitous Healthcare

Morris Sloman

Healthcare is coming under increasing pressure to improve the quality of care delivered to patients through effective prevention and post-operative care. This comes at a time when there is a need to curtail growth in healthcare spending fuelled by ageing populations, and the prevalence of obesity, diabetes, cancer and chronic heart and lung diseases.

Miniaturised implantable and on-body wireless biosensors will reshape common practice in clinical medicine especially for the prevention of terminal illness, monitoring the progression of chronic disease, and assessing post-operative care and body reaction to complex therapeutic drug regimes. Ubiquitous healthcare systems will monitor patients as they maintain their normal everyday activities, in order to warn the patients or healthcare workers of problems as well as collecting data for trend analysis and medical research. The use of continuous monitoring circumvents the drawbacks of conventional diagnostics and monitoring (generally limited to brief time points and frequently unrepresentative physiological states or artificially introduced exercise tests), allowing both transient and progressive abnormalities to be reliably captured. The integration of body sensors with home environment sensors can also be used for monitoring of the elderly to determine state of well-being and warn family or social care workers of potential problems related to physical fitness, social activity and cognitive engagement.

The key research challenges include:

- Development of new biosensors to accurately measure medical state.
- Power management - including micropower electronic circuitry and wireless communications, MEMS based power generation from body movement, and integration of multiple power sources with power storage.
- Fusion of multiple sensor information to determine human activity and medical state.
- Inferencing normal conditions and activity and hence detecting abnormal conditions.
- The infrastructure required for very large scale monitoring and analysis of medical information and activity of millions of people, and the need to automatically warn patients, social services, medical service, friends or family about the need for intervention when abnormal conditions are detected.

- Social, ethical, security and privacy issues related to continuous monitoring of people, storing and analysing the data and how to verify the safety, security and privacy aspects of the system.

A number of UK projects are already addressing these issues:

DTI Care in the Community Programme http://www.dticareinthecommunity.com/

UbiMon http://www.ubimon.org/

Biosensornet http://www.doc.ic.ac.uk/~mss/Biosensornet.htm

SAPHE: http://ubimon.doc.ic.ac.uk/saphe/index.php?m=338