A Very Personal Look at UBICOMP

- A massive spectrum of research from Ethnographic Studies, through programming models and systems infrastructure, to networked sensors

- Each necessarily characterised by *stunning* point-examples of technology and problem solving.

- Some toolkits are emerging, such as Smart-ITS, EQUIP, TRH, I-AM, SCINET, Context Toolkit,…

- A challenge – to draw together these advances to provide coherent building blocks, frameworks and tools to build small or large scale UBICOMP Systems.
Observations

- We are a young science, and so...
- Unclear how we *engineer systems* out of these components
- UBICOMP systems are (to us, anyway...) a fascinating direction for software engineering and systems research in general.
- The problems are those of systems engineering
The goal of location transparency has been assiduously pursued

- The web, CORBA, e-mail, ...
- Remove significance of – and usually any knowledge of – the (absolute or relative) locations of agents in a system
- Allow arbitrary interactions
But the world isn’t like that – 1

- Networks – especially the Internet – aren’t flat: they have a distinctly non-trivial topology
  - Firewalls *etc* introduce disconnectedness
  - Objects’ semantics are critically dependent on their location
  - ...and in a smart space, location changes

This observation also underpins Cardelli and Gordon’s ambient calculus
But the world isn’t like that? – 2

- Everything doesn’t happen everywhere
  - Certain activities occur (at least preferentially) in particular locations
  - People aren’t in two places at once
- Task and space impose a certain degree of orderliness on events
  - This happens after that, although not necessarily without interruption
  - If I do this here and that there, I have to get from here to there
  - The information and support I need while doing this may change when I start doing that
- Not everything is allowed – or disallowed, for that matter
  - Permission is a remarkably subtle concept
  - Not everything that happens happens for a reason..
Off-the-shelf technology

- Wireless networking
  - Bluetooth, IEEE 802.11, IrDA, GSM, ...
- Encryption and authentication
  - Software – GPG, the emerging public key infrastructure, ...
  - Hardware – iButtons, smart cards
- Localisation
  - Coarse – GPS, some cellphone operators, ...
  - Medium – RF tagging, computer activity, ...
  - Fine – person tracking cameras, “cricket” beacons, ...

All provide really useful plumbing and other low-level capabilities – but lack systems integration through applications
Dimensions of a system

www.smartlab.cis.strath.ac.uk

Physical

Topology

Organisation

Industrial Acme Inc
Acme Widgets
Acme Floobits

Nocturnal Aviation Ltd
Information gathering
Personal smearing
Money laundering

Devices

Personal

Information

Intranet

External web

A1

A2

A3

A4

A5

B1

B2

B3

B4

B5

Seminar room

Office
Policy
“Only a person working for Nocturnal Aviation can see the intranet”
Each dimension of the system defines a particular part of its behaviour, with the dimensions inter-related

- A person’s location affects the tasks they may (preferentially) perform
- A person belongs to an organisation, and that affects the information they should be able to access

Many current systems hard-wire some of these dimensions together, weakening their capabilities

- If you’re off-site, you’re an enemy; if you acquire this information, you can keep it; if you’re in here, you belong
Example

- A firewall introduces a barrier between objects
  - Prevent some interactions, based on location/credentials/protocol
- A concrete reification of something more subtle
  - Systems-architectural statement, business rationale, policy goal, ...
- If any of these factors changes, the firewall becomes a hindrance
  - Employees working off-site can’t access private material
  - ..but visitors working on-site can

All interactions seem possible under the model, but some are prevented.
Each instance of a middleware technology has a particular “architectural” style at its core.

**Question:** What are the architectural primitives for UBIQUITOUS Computing?
Issues for Architecture

What I’d like to do today is share with you some questions I have about building these architectures.

A reminder of the dream:

- “IT systems intimately integrate with everyday environments and supporting people in their activities”
- Unbounded sets of building blocks [embedded devices, stand alone devices, or software entities]
- Develop architectures with “universal” applicability – architectures inspired by real world scenarios
- Define and develop core architectures and underlying frameworks for ambient systems
To achieve this architectures and systems infrastructure must be able to:

1. Embrace contextual change
2. Encourage ad hoc composition
3. Facilitate sharing
4. Support both local and the global computation
5. Have multiple use view-points [interactions designer/user/architect/programmer/system]

[References on request. Grimm and Bershad. Nixon, Dobson and Lacey. Dearle, Kirby, and Morrison. Greenalgh et al]
1. DESCRIPTION

“"It should be possible to describe components and their interactions in a way that explicitly prescribes their abstract roles in a system”"

- A core challenge relates to how we build, share and use understandable descriptions.
- We need to describe not only the information but also the system and its configuration.
- We need to be able to reason about the semantics of these descriptions [sic semantic web technologies].
2. COMPOSITION

- “It should be possible to describe a system as a composition of independent components and connections”

- Current work on composition talks of composition rules, policies, aspects, etc. With a focus on composition from known sets.

- Ambient systems will additionally have a variety of composition rules [for the user, applications developer, system, hardware]

- There should be some structure for relating between the levels.

- What are the semantics of these rules [how do we describe a closure so that we can reuse/decompose the composition?]
3. DYNAMIC Composition

- “It should be possible to reuse components, connectors, and architectural patterns even if they’ve been developed for another purpose!”

- We need to be able to describe families of systems, their semantics and constraints from open sets [Shaw and Garlan].

- Typically, existing composition approaches use closed or parameterised sets.

- How do we support dynamic composition and still maintain a robust, predictable system?
4. INTERFERENCE [Kindberg and Fox]

- “We should be able to deal with changing and conflicting resource requirements in the environment”
- JINI, and others, adopt a notion of leasing resources.
- However, they give no solutions to the free market economy of the ambient systems world [Ginis and Chandy].
- Even the simple case is intractable.
5. GLOBAL versus LOCAL

- “We should be able to utilise the appropriate set of local and global resources to achieve the task”

- Ambient systems are not just local interactions.

- Ability to move between local environments and retain context

- So, where is the information and computation placed?
6. CONTROL

“We should be able to describe how the system is controlled with respect to a variety of changing parameters and from a variety of viewpoints”.

- Locating services and resources
- Manage resources from the perspective of the group/region/domain/...
- Coordinate the progress from a certain viewpoint
- Be able to express partial requirements
- Detect and recover from failure
7. VIEW POINT

- “We should be able characterise the usual interactions styles”.

- Optimise for the usual interactions
- However, this typically inhibits the ability of the system to adapt to the unusual.
- Equally, the vocabulary for describing ambient systems varies between domain. Is it possible to have a common framework within which the semantics can be debated?
8. CONTEXT

- “We should be able to contextualise interactions in order to adapt the infrastructure, information, or its delivery, to the semantics of use”

- Relates very much to viewpoint. How do we codify the behavioural characteristics of the user [Wilson et al]

- What is the peripheral variable set for this user, doing this task, in this situation [Coutaz et al]

- A core challenge in relating hardware sensed context with their semantics of use at that time. [Crowley et al]
9. REPRODUCIBILITY?

- “Should we be able to compare the “function” of one system with another?”

- As a scientist I feel uncomfortable with the ad-hoc nature of our evaluations of this new technology.

- There are a number of tools in different domains [Cultural probes, Technology Probes, Ethnographic studies, etc...], but none in the middleware domain.

- A core question – how do we compare our work to that of others in the domain.
Fundamental Problems

- Semantic multiplicity
  - Lots of events are “the same” in the higher-level view

- Non-linearity
  - There are multiple paths through the system, all “reasonable” and “possible”, rather than there being a canonical “right” process

- Sensor fidelity
  - Anything dealing with physical phenomena is noisy, so (for example) people are identified incorrectly, or not observed

- Latency
  - By the time you work something out it may be too late to do anything about it

- Placement of information and computation
  - Where does the system do the computation, how does it get the correct information, and how does it achieve this?
Concluding remarks

- The excitement I feel about this domain is partly due to the fact researchers in the field of Ambient Systems are still **dreaming** and **creating problems** [Lyytinen/Yoo].

- However, we need to be able to benchmark, validate and compare our findings.

- Catch-22 - Challenge to support serendipity **and** provide stability/predictability for the user and the system.

- Continue to incorporate the social, organisational, and human aspects into the design of the middleware.

- I’ve ignored many issues, trust, privacy, social aspects, ... to mention a few.