Engineering global ubiquitous systems

We are urgently in need of engineering design patterns for ubiquitous computing. UBICOMP threatens us with a grand dose of the ‘Interconnectedness of everything’ and so even the smartest of systems designers cannot hold all the factors in mind when building and deploying a UBICOMP application. Human factors, technology change and unexpected environmental circumstances mean that we need to equip the engineer with ready-made toolkits based on sound underlying principles that have been verified through more formal and societal testing than has been necessary hitherto in many successful software projects.

We are moving towards a society of things.

As we add information processing and communications capabilities to more and more everyday objects, we need to understand how these smart new devices relate to each other and to ourselves. So far, large complex distributed artefacts have appeared mainly as utilities. Thus, the analogy of the ‘information superhighway’ (as in roads) for the Internet, and ‘plumbing’ (as in water supply) for networking home and a ‘grid’ (as in the electricity grid) have been commonplace and useful for understanding how to engineer such systems.

As we embed computing in more and more disparate devices, and inevitably link them up in a looser set of federations, the utility engineering metaphors break down badly. This is bad, since we have few theories to underpin the design of these new systems, or explain their emergent properties to users.

However, like it or not, computing scientists and engineers are already having to build global ubiquitous computing systems before we have a fully fledged theory, and before we have a well-found understanding of its human experience aspects.

In decentralized systems, a number of theoretical models as well as engineering techniques have been applied: systems theories, control theory, graph theory and process algebras, including more novel mobile process models, are in development; control engineering, communications systems engineering (including performance modelling) and distributed systems engineering all have their place. However, all are stretched to breaking point by the novel properties of these new systems.

— We do not even know the fundamental limits of multi-hop, multiparty radio systems. This is a matter for both theory and practice.

— Hierarchy is an insufficient mechanism to manage the scale and heterogeneity of these new processing elements and communications links and their multifaceted relationships.

— We are only just starting to learn a little about real-world human and device mobility. This is a matter for models as well as real world data.

— We have no detailed or realistic models for the collective failure modes.

One contribution of 19 to a Discussion Meeting Issue ‘From computers to ubiquitous computing, by 2020’.
We do not have a methodology to design or understand systems of loosely integrated agents.

We are dealing in mixed reality where there are both sensors and actuators. We are not used to modelling the state of external physical world at the same time as programs.

We do not have appropriate models of or design patterns for ownership.

We do not commonly consider energy consumption in these systems, yet many are dependent on batteries, and we are under pressure to accommodate such considerations strongly in designs.

In general, we need a way to model, engineer and understand a society of devices embedded in a society of humans, embedded in a real physical world. Such models would go far beyond the current distributed systems engineering frameworks, viewpoints and transparencies. Once such models are uncovered, they may be mapped into design patterns—many of these may resemble natural systems. Hence, interdisciplinary research with natural biology may be an essential element for this challenge.

Jon Crowcroft

The Computer Laboratory, University of Cambridge,
Cambridge CB3 0FD, UK
E-mail address: jon.crowcroft@cl.cam.ac.uk