Places to Stay on the Move: Software Architectures for Mobile User Interfaces

Alan Dix\textsuperscript{1,2}, Devina Ramduny\textsuperscript{2}, Tom Rodden\textsuperscript{2} and Nigel Davies\textsuperscript{2}

\textsuperscript{1}aQtive limited, Birmingham Research Park, Birmingham, UK; \textsuperscript{2}Department of Computing, Lancaster University, Lancaster, UK

Abstract: Architectural design has an important effect on usability, most notably on temporal properties. This paper investigates software architecture options for mobile user interfaces, in particular those for collaborative systems. One of the new features of mobile systems, as compared with fixed networks, is the connection point to the physical network, the point of presence (PoP), which forms an additional location for code and data. This allows architectures that bring computation closer to the users, hence reducing feedback and feedthrough delays. A consequence of using PoPs is that code and data have to be mobile within the network, leading to potential security problems.

Keywords: Active networks; Client-server; Collaborative work; CSCW; Mobile computing; Software architecture

1. Introduction

At first sight, it seems that software architectures are about the internals of system design and not a necessary concern for the user interface. However, internal details have a way of showing themselves at the surface.

One aspect of this is the changing view that users have of systems as a result of the merging of computing and communication systems and the maturing of distributed computing techniques. With the growing prominence of an almost universally accessible information infrastructure, increasingly we are seeing our interaction focus on the communication infrastructure rather than the devices that access it.

Nowhere is this more evident than in the World Wide Web, which represents the most dramatic example of this shift in how we view our interaction with computer systems. The popular acceptance of massively interconnected computer systems has seen computer and communication systems being seamlessly interwoven within the everyday life of the general public. Interaction is now routinely through the web (rather than the machine used to access it) and our everyday experience of this interaction makes inherent assumptions about the architecture of the infrastructure.

As the opportunity to access the underlying infrastructure increases with the development of mobile devices, the link to the underlying architecture is likely to become even more significant and the need to consider the architecture of the infrastructure underpinning user interaction will grow.

Arguably, even with the web, users see a virtual architecture – is that ".co.jp" domain really served from Japan or from some other place? However, consideration of physical architecture becomes unavoidable when we look at temporal aspects. Again, the obvious delays on the web owing to network latency and bandwidth have raised the profile of these issues. For interactive networked applications, the need for rapid feedback – seeing the results of one’s own actions – is essential, leading to various forms of client-server architecture. For collaborative applications feedthrough – seeing the effect of other users’ actions – is equally important, but even more intractable owing to the physical distance between the users.

The importance of feedback and feedthrough can be demonstrated by considering the way in which a selection button behaves in most interfaces. When a user selects a button, this is reflected through a change in the appearance of that button. This feedback is central to the interaction. But when we consider shared interfaces, a new situation arises in that the effects of the actions need to be made available to more than one user. This implies that, in addition to providing feedback by changing the state of the selection button, the application also needs to feedthrough the effects of the action by altering the state of the selection button on each user’s interface. This feedthrough of effect is important.
in providing an awareness of the actions of others and in allowing users to coordinate their actions. Delays in the feedthrough of action will obviously make coordination of activities more difficult. Mobility brings yet more issues with unreliable and often low-bandwidth communications, small screens, restricted input devices. In some previous work, we have looked at the general nature of mobility and context awareness [1]. In this paper, we will analyse the architectural options available in developing mobile interfaces, especially collaborative systems. Our particular concern lies in the dynamic nature of the infrastructure required to support mobile and context-sensitive applications and the impact on the relationship between interaction and architecture.

We will begin by reviewing the current state of software architectures in human-computer interaction (Section 2) and discuss how these are realised in single-user, multi-user and web-based applications over fixed networks (Section 3). In Section 4, we will start the main work of this paper by looking at the additional issues which emerge for mobile single-user architectures and, in particular, the role of the Point of Presence (PoP) as an additional potential site for computation. Sections 5 and 6 are the heart of this work as we will describe how it is possible to support different architectures by making use of PoPs in collaborative mobile applications and how these can be dynamically reconfigured to optimise feedback, feedthrough and resource usage. Finally, in Section 7, we will look at some of the issues this raises for code mobility and security.

This paper does not present a particular system or single solution. Instead it is based on our own experience and that of others in producing fixed and mobile collaborative applications. However, the analytic framework takes us beyond existing systems design and is intended to inform and guide our own and other researchers’ ongoing development in the area.

2. Software Architectures in HCI

Concerns about the architecture of interactive systems are not new to HCI. Experience has shown that the internal structure of a system has a dramatic effect on its external behaviour [2]. Because of this, user interface architecture has been a concern for many years and has seen the development of models such as Seeheim, MVC, PAC and Arch/Slinky as well as the whole stream of UIMS systems [3-6]. The problem involved in developing appropriate interactive behaviour has focused on uncovering an architecture that offers the most appropriate set of dynamic behaviours for the activity being supported.

In collaborative systems research, the general interest in software architectures has continued with CSCW architectures and toolkits including Rendezvous, MEAD, Suite and Groupkit [7-10]. These collaborative systems almost always imply some form of networked solution and, increasingly, single user systems also involve access to central information, giving rise to a whole industry in client-server and n-tier applications.

The emergence of distributed architectures to support interaction across geographically disparate communities of users has seen a series of debates about the nature of these architectures and their impact on interaction. This debate has tended to centre on the propagation of the effects of the actions of users to others involved in the activity being supported. Essentially, the critical issue here is the means by which feedback of actions and the interface can best be supported. A core element in this discussion has been the tension between the responsive nature of replicated architectures, which allow rapid feedback to be provided locally and the need for some centralised component to make users aware of the action of others by providing feedthrough of actions.

The majority of these arrangements tend to assume “control” over the entire system, with bespoke software running at the users’ own workstations and at various central servers. The implicit assumption in many of these systems is that the machines are connected to a single local area network and the properties of this network are stable. However, two developments have recently challenged this assumption and, hence, the whole basis for network-based user interfaces.

The first is the World Wide Web, which has promoted alternative architectural arrangements for applications. A web “application” may include code running at a web-server (via CGI scripts or other server side technology), web pages displayed on browsers of many different kinds, and applets or similar downloaded code [11]. In previous work we and others have investigated the ways in which CSCW architectures can be married to the web infrastructure [12-15].

The second development is the massive growth in mobile communications and mobile computing. Although the end points here may be well