Mobile Processes and Mobile Channels

Kevin Chalmers
Napier University, Scotland

Imed Romdhami
Napier University, Scotland

Jon Kerridge
Napier University, Scotland

INTRODUCTION

Since the late 1990s there has been an explosion of new Internet technologies, making the Internet a primary medium for academic, business, and focus interest groups. As a result, new applications are constantly emerging and evolving. Couple this with the concept of mobility, something that is all around us today, and it can be seen that new applications require vastly different characteristics to those envisioned by the original Internet architects. For example, point-to-point IP traffic was designed to allow stationary parties to communicate, but when one of these parties becomes mobile with a wirelessly enabled mobile device, the original models become more difficult to implement.

This trend is a direct result of the fast development of mobile and wireless technologies, and it brings with it new problems. A nomadic user, equipped with a laptop or personal data device with wireless Internet connectivity, still requires and expects seamless communication while they migrate between locations. To overcome this and other issues, mobility is generally tackled at a hardware level, using and modifying existing underlying network techniques to accommodate this new mobile aspect. Designing a truly mobile system, however, also requires mobile software—software components that have the same freedom of movement as the hardware they are executed on.

BACKGROUND

Recent advances in programming frameworks, such as Microsoft's .NET framework and Sun Microsystems's Java platform, provide a number of mechanisms allowing applications to not only be portable between devices, but for applications to actually move from one device to another during runtime (Brooks, 2004; Delamaro & Picco, 2002). These types of systems fall under the general heading of code mobility platforms (Fuggetta, Picco, & Vigna, 1998), the same category as mobile agents and remote invocation systems. These interactions are usually modeled using more traditional techniques such as UML, but this does not truly allow us a very high-level view of how a mobile system actually behaves, being based on architectures aimed at much more static, single machine systems. Therefore a different paradigm should be employed, and here we examine mobile processes and mobile channels. Viewing a mobile component as a mobile process and a communication between them as a mobile channel, a much broader viewpoint of system mobility can be achieved, facilitating development, understanding, and maintainability.
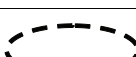
Mobility

Mobility is all around us today. Wireless technology and mobile phone ubiquity have brought about a technological landscape that is no longer bound to a specific site, but can be carried globally from location to location moving around as we do. The trouble is we still expect to have the same level of service as we migrate as we get when we are stationary. This is of course difficult (imagine trying to hop from one moving train to another instead of having stations), and research into how seamless communication can occur is ongoing in such fields as mobile IP (Johnson & Perkins, 2004; Perkins, 2005) and migratory TCP (Sultan, Srinivasan, Iyer, & Iftode, 2002).

Here, we consider mobility more at the software level, although the basic concepts can be utilized for modeling lower level interactions. Indeed, most of the ideas presented are based on a more formal model—the π-calculus (Milner, 1999)—which has been used to model hardware processes such as mobile phone migration between base stations (cells). We do not, however, use this more formalized style here, but adopt a much softer view of mobility.

In the following paragraphs we will describe what mobile processes and channels are, and how they can be used to model some basic systems. These simple yet powerful building blocks are available in the JCSP (Communicating Sequential Processes for Java) package (Welch, Aldous, & Foster, 2002), which is based on yet another formal model, Communicating Sequential Processes (Hoare, 1978). Recent
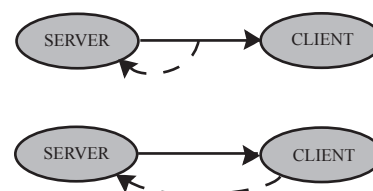
MOBILE CHANNELS

A mobile channel can be viewed in simple terms as a channel object that can be sent across a communication channel. This could be another channel altogether, or the channel input end could send its opposite output end if necessary (although the need for this is questionable). Milner (1997) considers this ability of communication a sign that it is sufficiently adult, in that it can utilize a property of itself. In other words the answer to the question *Can a communication communicate a method of communication?* is yes. Usually, only a single end (the input or output) is sent, allowing another process to communicate with the sender, or another process, over the received channel. If, for example, we examine a streaming server system, it is possible to create a client-server communication by the server sending a mobile channel end to the client, as in Figure 1.

The client process can now communicate back to the server using the mobile channel end. Mobile channels, just like normal channels, can also communicate channel ends. Or channels can be passed much further down a chain of connected processes, allowing remote systems to start communicating without prior knowledge of one another. This allows such network application interactions as peer-to-peer to be modeled. This would simply involve a peer requesting the output end for another peer from a central server, and the opposite doing the same for a connection back to it. This allows the two peers to communicate with each other without knowing where they specifically were before the interaction.

When we talk about a mobile channel end, we do not necessarily mean that the underlying software component is actually migrating around a distributed environment. This kind of interaction is possible, but it is also true to say that the location of the channel end can be passed and the necessary channel created dynamically when this location is received. If we consider channel mobility to also include this, then we can see that simply typing in the channel location (a URL for example) becomes a simulated mobile channel. In short, almost every distributed connection can be modeled as a mobile channel. What the more structured mobile channel enables is the movement of a connection without losing any sent or received messages, while still providing

Table 1. Symbols

Symbol D	escription
→	A (stationary) channel
⬭	A (stationary) process
⇢	A mobile channel
⬭ (dashed)	A mobile process

work (Chalmers & Kerridge, 2005) has simplified the usage of the mobility constructs within this architecture, allowing many interesting approaches to system development to occur, some of which shall be described presently.

SYMBOLS USED

Before moving on, we will define in Table 1 some basic symbols that shall be used.

SIMPLE PROCESS NETWORK

Processes communicate to each other using channels; synchronizing during a read-write operation (i.e., when a read occurs, the process waits until the associated write operation occurs, and vice versa). A set of processes interconnected by channels is generally referred to as a process network. Also of note is the fact that messages travel in one direction on a channel, as opposed to the two-way viewpoint normally thought of when considering network connections. For example if we consider a client-server application, using channels to model the connection, two separate channels will be created, one from the client to the server, and one from the server to the client. Using this level of interaction allows some interesting models to be built, especially related to events generated by messages being sent across a channel, which allows distributed event-based systems to be simply developed. Remember that channels synchronize on communication also, adding more interesting models.

Process and channel networks can be built up to a very high level of complexity if required, but individual processes themselves are generally quite simplistic with simple input and output interfaces. Processes can also contain other processes and channels operating within them, permitting the detail of a process network to be viewed at differing levels if required.

Figure 1. Mobile channel

Related Content

On-the-Move and in Your Car: An Overview of HCI Issues for In-Car Computing

G.E. Burnett (2009). *International Journal of Mobile Human Computer Interaction (pp. 60-78).*

www.irma-international.org/article/move-your-car/2762

Mobile Services as Resources for Consumer Integration of Value in a Multi-Channel Environment

Veronica Liljander, Johanna Gummerus, Minna Pihlströmand Hanna Kiehelä (2013). *Strategy, Adoption, and Competitive Advantage of Mobile Services in the Global Economy (pp. 259-282).*

www.irma-international.org/chapter/mobile-services-resources-consumer-integration/68087

Assessing Human Mobile Computing Performance by Fitts' Law

Thomas Alexander, Christopher Schlick, Alexander Sievertand Dieter Leyk (2009). *Mobile Computing: Concepts, Methodologies, Tools, and Applications (pp. 206-224).*

www.irma-international.org/chapter/assessing-human-mobile-computing-performance/26501

Interactive Product Catalog for M-Commerce

S. Guanand Y. Tay (2007). *Encyclopedia of Mobile Computing and Commerce (pp. 345-351).*

www.irma-international.org/chapter/interactive-product-catalog-commerce/17099

Online Authentication Using Smart Card Technology in Mobile Phone Infrastructure

Teddy Mantoro, Admir Milišicand Media Ayu (2011). *International Journal of Mobile Computing and Multimedia Communications (pp. 67-83).*

www.irma-international.org/article/online-authentication-using-smart-card/58906