XMIDDLE: Information Sharing Middleware for a Mobile Environment.

Stefanos Zachariadis, Licia Capra, Cecilia Mascolo and Wolfgang Emmerich
Dept. of Computer Science
University College London
Gower Street, London WC1E 6BT, UK

{S.Zachariadis|L.Capra|C.Mascolo|W.Emmerich}@cs.ucl.ac.uk

The capabilities of modern mobile devices enable new classes of applications to exploit the ability to form ad-hoc workgroups and exchange data in a very dynamic fashion. They also present, however, new challenges to application developers, related with the scarcity of resources that need to be exploited efficiently. Moreover network connectivity may be interrupted instantaneously and network bandwidth remains by orders of magnitude lower than in wired networks. To address such issues, we have designed and implemented XMIDDLE, which advances mobile computing middleware approaches by choosing a more powerful underlying data structure (XML) and by supporting off-line data manipulation.

XMIDDLE provides an approach to sharing that allows off-line data manipulation, synchronization and application dependent data reconciliation. On each device, a set of possible access points for the owned data tree are defined so that other devices can link to these points to gain access to the information. In order to share data, a host needs to explicitly *link* to another host's *exported* data branches. The data (and associated metadata) are cached on to the target host, allowing disconnected operations on the information. Hosts may explicitly disconnect using the *disconnect* primitive, even though these hosts may be *in reach*. The specific definition of *in reach* depends on the network protocols and hardware devices used. Considering wireless LAN and Bluetooth, *in reach* means in radio range.

XMIDDLE supports explicit disconnection to enable, for instance, a host to save battery power, perform changes in isolation from other hosts and not receive updates that other hosts broadcast. Disconnection may also occur due to movement of a host into an out of reach area, or to a fault. Upon reconnection, XMIDDLE can automatically reconcile any changes to shared data, using application metadata to resolve any conflicts. XMIDDLE implements a distributed versioning scheme, storing versions or snapshots of shared branches on each host. The versions can be stored as a delta of changes from another version and can be further

Figure 1: (a) The addressbook application running on the iPAQ (b) More features of the application, such as peer discovery

compressed to minimise memory requirements on the device. Furthermore, the middleware allows hosts to delete versions in order to further conserve memory. Hosts reconciling changes can identify common versions and use the latest one as a basis for reconciliation, thus conserving computational power and network bandwidth.

XMIDDLE also features a protocol plug-in architecture, whereby a host can dynamically update its list of available protocols at runtime, so as to acquire new behaviours, allowing, for example, applications to use different reconciliation algorithms depending on the context.

We have implemented the XMIDDLE platform and an addressbook application in Java, on a Compaq iPAQ PDA running Linux. The XMIDDLE platform currently requires just 136KB of space, while the addressbook application requires 28KB (both uncompressed). We present a demo of our implementation, running on PDAs and laptop computers, equipped with 802.11b compliant wireless cards. We demonstrate the linking and reconciliation protocols, as well as disconnected operations on the shared data. XMIDDLE's resilience to involuntary network disconnections is also shown.

References: XMIDDLE: A Data-Sharing Middleware for Mobile Computing. Cecilia Mascolo, Licia Capra, Stefanos Zachariadis and Wolfgang Emmerich. In Personal and Wireless Communications Journal, Kluwer. April 2002.
http://pizza.cs.ucl.ac.uk/xmiddle/