Supporting Collaboration in Mobile-aware Groupware

Keith Cheverst, Gordon Blair, Nigel Davies and Adrian Friday

Distributed Multimedia Research Group,

Department of Computing,

Lancaster University,

Lancaster, LA1 4YR.

telephone: +44 (0)1524 65201
e-mail: kc,gordon,nigel,adrian@comp.lancs.ac.uk

1. Introduction

This paper builds on the experiences gained from the extensive work carried out during the MOST project [MOST,95]. This project investigated the use of mobile technologies to support multimedia collaboration between highly mobile field engineers in the safety critical domain of the U.K. power distribution industry. The MOST team developed a prototype distributed groupware application which was arguably the first collaborative mobile application ever built that was capable of adaption in a mobile environment [Cheverst,94]. The application was designed as an expandable toolkit comprising a number of modules including a shared GIS module. This module enabled groups of field engineers to perform spatially-aware collaboration by supporting the display and annotation of network schematics across groups. The MOST application was evaluated by real end-users in a trial scenario using the GSM service for communications. This evaluation provided a valuable set of implications regarding the development of distributed groupware in conjunction with mobile technologies.

The main argument of this paper concerns the need for a new type of *awareness*, i.e. *mobile-awareness,* that provides group members with relevant information concerning the effect that the constraints imposed by their mobile environment might have on the group's collaboration. This approach couples and expands upon the notion of adaption [Davies,94] and the approach argued in [Dix,95], i.e. providing users with increased levels of awareness to deal with the problems of group collaboration in the presence of unreliable communications. In order to investigate some of the issues regarding the development of this class of groupware, we have engineered a *mobile-aware* version of the MOST application using our own flexible QoS-based group service.

2. The Impact of Mobile Communications on Distributed Groupware

Mobile communications implies the utilisation of different networking technologies in order to maintain network connectivity whilst mobile. For example, mobile computers can be either disconnected, weakly connected by low speed wireless networks such as GSM, or fully connected by high speed networks ranging from Ethernet to ATM.

The problem faced by developers when building mobile applications is that when users roam between areas of different network infrastructure this can result in rapid and massive fluctuations in the quality of service (QoS) provided by the underlying communications infrastructure. For example, a user might begin the day with their portable computer docked to a docking station with a high bandwidth (i.e. 100 Mbps) ATM network link. Later on, the user may choose to undock their portable and move around their department whilst maintaining network connectivity through a lower bandwidth, local area RF (radio frequency) based, network such as WaveLan (providing a maximum bandwidth of 2 Mbps). When required to leave the department building, the user could still achieve network connectivity by utilising the wide area but low bandwidth (i.e. 9.6 kbps) GSM service. However, whilst using this service the user might occasionally enter areas referred to as 'coverage blackspots' and temporarily loose network connectivity.

One of the MOST application's key requirements was the ability to operate over both heterogeneous networking and processing architectures. In particular, the application was required to be capable of switching between analog PMR, GSM and wired ethernet based networks and also capable of running on both portable windows based PC's and legacy based UNIX platforms. For this reason, MOST used the ANSAware [APM,92] as its development platform which is a partial implementation of the Advanced Networked Systems Architecture (ANSA) [APM,89] and has been influential in the specification of RM-ODP [ISO,95a],[ISO,95b]. In common with more recent open standards, such as CORBA [OMG,98], RM-ODP based platforms provide application developers with a number of *abstractions* or *transparencies* for masking out various features of a distributed computation. Two examples of the transparencies provided are *network transparency* and *processing transparency* which, together, enable systems to operate over a variety of machine/network configurations. Another transparency provided is *group transparency*. This gives the application programmer an abstraction for dealing with groups that hide specific group details such as the identity of individual group members.

Unfortunately, when developing mobile-aware groupware these transparencies can hide from the programmer the details required for providing the user with mobile-awareness. For example, the strict enforcement of network and group transparencies makes it impossible for the application programmer to receive sufficient levels of feedback regarding changes in the quality of communications available to individual group members. The result of this is that mobile groupware developed using these transparencies tends to hide group communications problems from users and thus forces them to assume a constant level of communications quality. For example, when evaluating an initial version of the prototype MOST application end-users felt that they were not given sufficient information concerning many of the constraints that the unreliable mobile communications environment was imposing on their collaboration. More specifically, the following criticisms were raised :-

Insufficient Temporal Feedback

End-users were confused and frustrated when given no feedback or appreciation of the fact that establishing a connection to the rest of the group using the GSM service could take over ten seconds and that significant delays could occur before their shared operations where received by all group members.

Insufficient Feedback on Group Consistency

Difficulties with communication and connectivity was not constant across all group members. For this reason, end-users wished for feedback regarding those particular group members that were unable to receive their shared operation. It is important to note that, one of the application's requirements was to maintain a high level of data availability (even at the expense of possible inconsistencies between group members) and therefore the MOST application does not enforce either total or causal ordering. The potential for inconsistencies was generally acceptable to engineers providing that they were able to receive feedback should any inconsistencies arise. However, for certain operations engineers did require some form of consistency guarantee. For example, an engineer might require an operation to have atomic delivery semantics, i.e. to be received by a certain set of group members or none at all. Similarly, an engineer might require a shared operation to be received by a certain quorum of group members.

Lack of Support for Managing the Cost of Group Operations

One of the constraints imposed by a mobile communications infrastructure is cost. Depending on the service being used a cost may be charged on a per second or a per byte basis (or not at all). In the context of MOST, it was found that engineers, using the GSM service for communications, required information regarding the ongoing cost incurred during a collaboration and also the ability to have some form of control over this cost.

In general., traditional distributed groupware systems (and the support services available to develop them) also hides details concerning the state of group communications from users and also assumes a constant level of communications QoS.

3. Mobile-aware Distributed Groupware

The key to creating some form of compatibility between distributed groupware and mobile communications is mobile-aware groupware. Such groupware builds on the concept of awareness [Dourish,92] [Lauwers,90] to provide group members with feedback to make them fully aware (or rather as aware as they wish to be) of the effect of group communications on their collaboration. Such awareness should prevent group members from being forced to make (possibly false) assumptions regarding the current state of their connectivity with the rest of the group.

To investigate some of the issues surrounding the development of mobile-aware groupware a modified version of the MOST groupware application was produced. The application has an enhanced session management module that is responsible for enabling modules to be launched across groups of engineers and which is also responsible for supporting mobile-awareness via the following :-

i) a convenient and easy to use mechanism for associating QoS-based requirements with group operations, e.g. reliability, cost, ordering and temporal requirements,

ii) a clear and graphic method for receiving feedback should the communications environment cause any of the QoS requirements to be violated.

The session manager's GUI, shown in figure 1, uses coloured icons to provide feedback to group members regarding the state of connectivity within the group and the violation of any QoS requirements. For example, a disconnected group member's icon would be displayed with a red background, while a connected group member's icon would be displayed with a green background.

Figure 1: GUI for providing increased levels of awareness.

The GUI makes extensive use of scrollable areas in order to maximise the small display area found on many portable PCs. The top left hand side of the window contains icons representing the modules which are available to the user (the globe represents the GIS module). On the top right hand side of the window is a scrollable column of icons representing engineers that can participate in a collaboration and below this there are icons for enabling members to be added and removed from the group membership. In the centre of the window is a row of 'member' icons representing current group members. Under each member's icon is a column of 'module' icons which represent the modules which that user is currently running. The user can specify QoS requirements using the smaller 'guarantee' icons (i.e. those labeled O,C,G,T and Q) positioned beneath each member icon and each module icon. These icons represent from left to right: required ordering, maximum cost, required reliability, maximum delay and required quorum. The user can toggle each type of guarantee between active and inactive by simply clicking on the appropriate icon.

To provide the required level of flexibility, the GUI enables QoS requirements to be made against both individual group members and the entire collaborating group. For example, an engineer could specify the requirement that their next shared operation should be reliably received by the entire group but that only group member 'Joe' needs to receive the operation within two seconds. This flexibility is important because certain group members, e.g. a monitoring process, might not require the same requirements as other group members.

4. Support Services for Mobile-aware Groupware

In order to support the development of mobile-aware distributed groupware, developers require novel and flexible group services. In particular, such services need to provide *flexible* group and network transparencies and also support the flow of information between the application, the platform and the communications infrastructure (see figure 2).

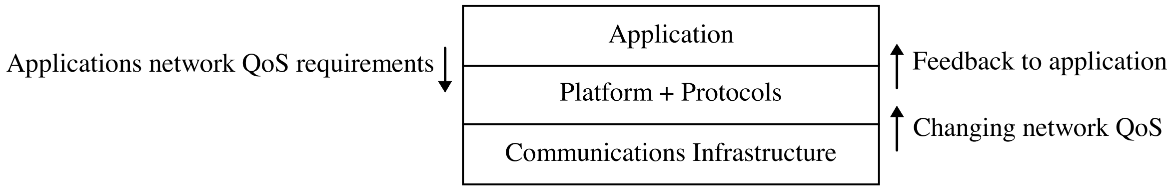

Figure 2: information flow required to support mobile applications.

Our RM-ODP based group service [Cheverst,96] supports the development of mobile-aware groupware by possessing the following three key properties :-

Flexibility

The most important property of the group service is that of flexibility. The group service enables the relaxation of message ordering and reliability guarantees which in a weakly connected environment are extremely costly in terms of performance. Such a performance penalty is completely unnecessary if the shared operation being propagated does not require such strong guarantees.

In addition to message ordering and reliability requirements, the group service enables the application programmer to specify the following :-

i) the quorum of group members required to receive a shared operation,

ii) the time-out period within which either the entire group or specific individual group members must acknowledge receipt of a shared operation,

iii) the cost which the client is prepared to pay in order for either the entire group or specific group members to evaluate the shared operation.

Ability to Provide Feedback

In order to enable feedback to the application, the group service enables application programmers to selectively break group transparency by enabling them to associate specific QoS-based requirements with group updates. If, when propagating a group update, one or more of these requirements cannot be met then the group service can provide appropriate feedback to the application.

Ability to Adapt

The group service is capable of performing intelligent adaption i.e. tailoring its behavior based on changes in the underlying communications infrastructure [Davies,94]. For example, the group service can save resources by not attempting to propagate a group operation to any group members which are known to be currently disconnected and unreachable. To give a slightly more sophisticated example, consider a situation where a user has requested that his or her next group operation needs to be completed within five seconds. However, one of the group members has network connectivity provided by a GSM handset with a call set-up time of fifteen seconds. In this situation, the response of the group service depends upon whether or not the group member equipped with GSM is connected at the time when the user issues the group operation. If the group member was not connected at the time then the group service should not attempt to propagate the operation to that member because the propagation could not occur within the specified time limit.

The group service also provides application developers with a convenient service for handling group based issues, such as: group broadcast, dynamic group membership and the enforcement of membership policies. By using the service the development effort for building distributed mobile-aware groupware is greatly reduced.

5. Summary

This paper has examined the problems associated with using distributed groupware in a mobile environment. The fundamental problem is that current groupware tends to assumes a reliable and constant quality of communications which does not exist in a mobile environment. The implications of making this false assumption are two-fold. Firstly, systems which enforce certain ordering and reliability semantics across all operations can suffer drastic performance penalties. Secondly, users will not be given sufficient awareness of (and control over) the effect of the unreliable communications environment on their collaboration. A new class of mobile-aware groupware is required in order to address these issues. However, the problem with building mobile-aware distributed groupware using existing development tools and services, such as those based on RM-ODP, is that they mask the programmer from low-level networking and group based details. In order to address this need, a novel RM-ODP QoS-based group service has been developed which enables programmers to build mobile-aware groupware systems that provide users with an awareness of the current state of group communications and the ability to control the performance/consistency tradeoff.

6. References

[APM,89] APM Ltd. (1989). “The ANSA Reference Manual Release 01.00”, Architecture Projects Management Ltd., Cambridge, U.K.

[APM,92] APM Ltd. (1992). “An Introduction to ANSAware 4.0”, Architecture Projects Management Ltd., Cambridge, U.K.

[Cheverst,94] Cheverst, K., Davies, N., Friday, A. and G. S. Blair. (1994). “Moving the 'Desktop' Into the Field”, IEE Colloquium on Integrating Telecommunications and IT on the Desktop, Savoy Place, London, U.K.

[Cheverst,96] Cheverst, K., Davies, N., Friday, A. and G. S. Blair. (1996). “Services to Support Consistency in Mobile Collaborative Applications”, 3rd International Workshop on Services in Distributed Networked Environments (SDNE), Macau, China, IEEE Computer Society Press.

[Davies,94] Davies, N., Cheverst, K., Friday, A. and G. Blair. (1994). “Supporting Adaptive Services in a Heterogeneous Mobile Environment”. Workshop on Mobile Computing Systems and Applications (MCSA), Santa Cruz, CA, U.S., IEEE Computer Society Press.

[Dix,95] A. J. Dix (1995). “Cooperation without (reliable) Communication: Interfaces for Mobile Applications”. *Distributed Systems Engineering,* 2(3) pp. 171-181.

[Dourish,92] Dourish, P. (1992) “Awareness and Coordination in Shared Workspace.” in Proceedings of the ACM CSCW'92 Conference on Computer Supported Cooperative Work. p107-114, Toronto, Canada, November.

[ISO,95a] ISO/IEC 10746-2 | ITU Recommendation X.902, (1995). “Open Distributed Processing - Reference Model - Part 2: Foundations”.

[ISO,95b] ISO/IEC 10746-3 | ITU Recommendation X.903, (1995). “Open Distributed Processing - Reference Model - Part 3: Architecture”.

[Lauwers,90] Lauwers, J.C. and Lantz, K.A. (1990). "Collaboration awareness in support of collaboration transparency." in Proceedings of the ACM SIGCHI'90 Conference on Human Factors in Computing Systems, p303-211, Seattle, Washington, April.

[MOST,95] Davies, N., Cheverst, K., Friday, A. and G. S. Blair (1995). "Mobile Open Systems Technologies For The Utilities Industries", Remote Cooperation - CSCW for Mobile and Tele-workers. A. Dix, Springer Verlag.

[OMG,98] "The Common Object Request Broker: Architecture and Specification", Revision 2.2., February 1998, http://www.omg.org/corba/corbiiop.htm.