

POSITION STATEMENT ON PRODUCING A GROUP COLLABORATION BASED APPLICATION TO OPERATE IN A WEAKLY CONNECTED ENVIRONMENT

Keith Cheverst, Adrian Friday, Nigel Davies and Gordon S. Blair

Distributed Multimedia Research Group,
Department of Computing,
Lancaster University,
Bailrigg,
Lancaster,
LA1 4YR,
U.K.

telephone: +44 (0)524 65201
e-mail: nigel, gordon, kc, adrian@comp.lancs.ac.uk

1. Introduction

The area of Computer Supported Cooperative Work (CSCW) and the development of groupware applications has become very topical. However, to date little research has taken place on the issues raised regarding how one might design collaborative applications which are intended to operate over the weakly inter-connected network inherent in a mobile environment [Duchamp,92]. This position paper will describe our efforts in producing such an application.

Section 2 of this paper presents an overview of our application. Section 3 then describes the application's user interface. Following this, Section 4 details the structure of the application. Finally, section 5 contains our concluding remarks.

2. Application Overview

Field engineers working in the power distribution industry often have to respond to emergency situations. Currently, they are hampered by the lack of information which they have readily available whilst in the field. In order to receive the information which they require to perform a repair, the engineers either have to make a time consuming trip back to the office or try to get in contact with an often engaged central control centre. Our aim is to provide the means for field engineers to have at hand most of the information that they might require. This means providing the engineers with portable computers capable of storing large quantities of multimedia information such as network diagrams. This can now be achieved by using portable computers with CD-ROM drives attached. A further step to improving the efficiency of field engineers is to allow them to collaborate with one another. An example of this might be one field engineer being able to highlight an area of network diagram that is being displayed on a remote field engineers display while the two engineers discuss a problem. To offer field engineers such facilities we have implemented a collaborative multimedia application which should be of great benefit to field engineers in their day-to-day work [Davies,93]. The application has support for viewing geographical information (e.g. network diagrams) via a Geographical Information System (GIS), support for collaboration with other field engineers, and support for audio communications. Given the engineers' highly mobile working environment, the application needs to be able to respond to rapid and massive fluctuations in the underlying quality of service (QoS) provided by the underlying communications infrastructure [Katz,94].

3. User Interface

An example screen shot of the application's user interface is shown in figure 1. The conference manager or group co-ordinator component of the application is shown in the top left of the diagram. This component makes extensive use of icons [Gittins,86] to represent users and their applications. On the left hand side of the group co-ordinator component are icons representing the sub-applications or *modules* which are available to the user (the telephone represents audio communications and the globe GIS services) and on the right hand side are icons representing users that can participate in conferences. In the centre is a list of the current conference participants. Under each participant's icon is a column

of further icons which represent the modules which that user is currently running in conference mode (as opposed to stand-alone mode). By selecting these icons in rows the user is able to easily direct messages to a subset of the conference members. The user is able to select (by clicking on a single button) whether or not an operation (for example drawing a line in the GIS module) is propagated to those group members whose module icons have been selected.

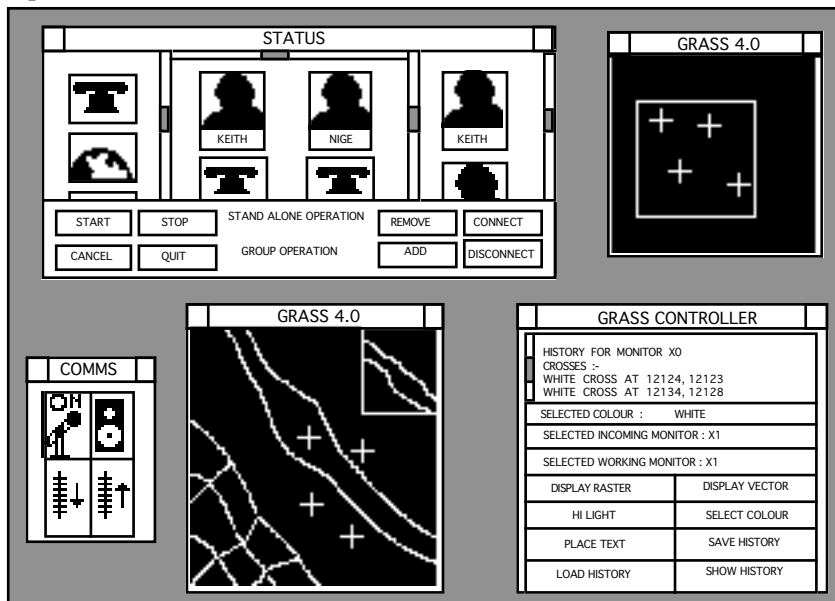


Figure 1 : The user interface

Considerable emphasis has been placed on reporting to the user the state of the communications links between local modules and remote user's modules. In our prototype this is achieved by colouring the module icons under each user in the central portion of the display to reflect connectivity. If a module icon is visible under a user icon then that user is actually running that module in group mode. The module icon's background colour indicates whether or not the user's module is currently reachable. So for example, if a communications failure occurred this would be indicated by a red background.

Users are assisted when performing operations within the group module by the appropriate highlighting of both user and module icons. As an example of this, before an operation button has been pressed (e.g. add user) all such operation buttons would be highlighted whereas all user icons would be greyed. This use of greying is to warn the user that currently clicking on a user icon would have no effect. However, on clicking an operation button the operation buttons would become greyed and the user icons (and cancel button) would become highlighted.

Figure 1 also shows the user interfaces to the audio conferencing module (bottom left) and the GIS services (bottom middle, bottom right and top right).

4. Application Structure

4.1 Design Overview

The application comprises two main components. The first is the core of the application which contains generic application code (such as the X window initialisation) and code for controlling the second form of component, the application modules.

We chose a modular design for two main reasons. Firstly, from a user interface perspective the application should appear as a set of cooperating applications (or toolkit). In this way the user can configure which portions of the toolkit are running. Secondly, from an implementation point of view, we wanted to be able to integrate existing applications with the toolkit with the minimum of effort, leaving the way open to integrate new applications as and when required. The modularisation also enabled the design team to work effectively in parallel once the basic architecture had been chosen.

To assist our modular design and to overcome the problems of the heterogeneity found in mobile networks [Duchamp,92] we decided to base our application on the emerging International Standards Organisation (ISO) standard for Open Distributed Processing (ODP) [ISO,92]. This provides applications with a uniform method for accessing services via sets of well defined interfaces which are hardware independent. The distributed systems platform we chose to use was ANSAware [APM,89], a leading example of an open platform.

4.2 Structure Detail

We also decided that the application should behave as a standalone toolkit unless communication with other users was explicitly required. The group functionality was therefore abstracted out as much as possible into a separate component module, and like any other component module it can be started as and when desired. Although the edges between the core of the application and the group module became slightly blurred (as they had much in common, particularly when starting and stopping local and remote modules), all access to group facilities is still managed through the group module.

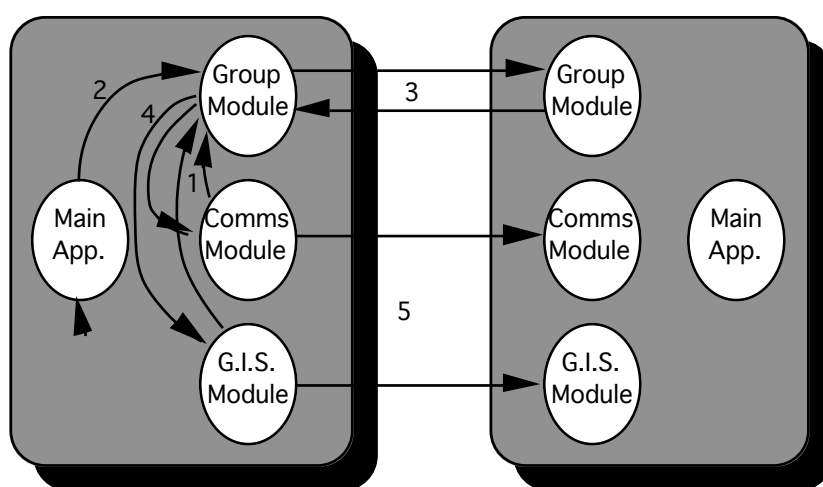
Each component module supports an ODP interface for communicating with its opposite number on other machines. Along with this external interface, each module supports an internal interface with four operations:

Operation	Effect
Initialise	The initialise operation in each module is called during initialisation of the application from the core. It should provide any module set-up and create its ODP interface. The value returned by this function informs the core whether or not the module can be used.
Show	This operation is called when the module is started to allow it to set up its user interface and communicate with its counterparts on other machines as necessary.
Hide	The converse of the previous operation, it is called to close down the user interface and shut down any communications held with other modules.
Quit	The quit operation should optionally hide the application, then clear up after the module so the application is ready to quit.

Amongst the component modules, the group module plays a particularly important role. It provides the user interface for conference management, allowing the user to specify with whom they wish to talk and with which modules. Internally, the group module also provides a form of brokerage service for ODP interfaces to the component modules of the other users. Each component can register an interest in group membership changes, so that when members join and leave they receive an interface to the equivalent module in the member's application. From this interface components can then provide application specific group configuration. This is achieved through another internal interface which has the following operations:

Operation	Effect
TestGroupMode	Check to see if the user has specified group operations such that any local action should propagate to other members of the group.
SetGroupMode	Allows modules to set whether or not operations are to be propagated.

RegisterJoin	Registers a call-back function with the group module so that the component can be contacted when someone joins the conference. This allows the component to reconfigure in order to incorporate the new member.
DeRegisterJoin	Removes the call-back supplied by the previous operation.
RegisterLeave	Registers a call-back function to inform the component when a member leaves the conference.
DeRegisterLeave	Removes the call-back supplied by the previous operation.
ReportFailure	Allows a component to inform the group module of problems with other member's modules, allowing it to update the status display informing the user of any problems.



1. Component modules register their interfaces with the group module
2. Another user is added to the conference from the X interface
3. Group module asks foreign module for interfaces of its modules
4. Group backcalls components with interfaces of new member
5. Components can now talk and reconfigure

Figure 2 : Reconfiguring for a new member

Figure 2 shows a stylised diagram of how the modules interact internally in order to add a new member, the platform on the left initiates the operation.

4.3 Group Issues

Currently our prototype application has been engineered over a high speed reliable network (Ethernet) and does not have fully implemented support for operating over a weakly connected network. In order to operate over such a network a number of issues need consideration and these are mentioned below.

Currently there is no distributed systems platform available which offers reasonable group handling mechanisms over a weakly connected communications infrastructure. Because of this, we have implemented our own group management functions on top of ANSAware. However, there is still an outstanding need for low level support to report network failure.

Owing to the frequency with which network failures are expected to occur we have designed our application to maintain a high degree of state information which allows new group members to be brought up to state. This mechanism still needs to be properly integrated with network failure detection in order to detect those members who, through temporary disconnection, do not receive group updates. It is important that such members are brought back up to state upon reconnection.

One solution which allows the application to be aware of network state is the extension of the underlying RM-ODP platform to make use of quality of service bindings. These refinements will allow applications to *react* [Davies,94] or *adapt* [Katz,94] to changes in their communications infrastructure.

5. Concluding Remarks

To date there has been little work on how one can produce collaborative applications which expect usage over varying qualities of network including weakly connected as would encountered in a highly mobile environment. We have considered the problems involved in producing such applications and have gone some way towards implementing some of our proposed solutions.

This position paper has described our approach regarding three main issues. The first issue is how to devise a user interface which would allow group members to visualise the current state of the group. The second issue concerns how one might construct an application infrastructure which allows the inclusion of new modules to be straightforward and which offers primitives for events such as modules being lost from the group. The final issue was how to provide the application with feedback on the QoS currently being offered by the underlying communications infrastructure.

References

- [APM,89] A.P.M. Ltd. "The ANSA Reference Manual Release 01.00", APM Cambridge Limited, UK. March 1989.
- [Davies,93] Davies, N., G.S. Blair, A. Friday, A.D. Cross, and P.F. Raven. "Mobile Open Systems Technologies For The Utilities Industries." Proc. IEE Colloquium on CSCW Issues for Mobile and Remote Workers, London, U.K, March 16, 1993
- [Davies,94] Davies, N., S. Pink, and G.S. Blair. "Services to Support Distributed Applications in a Mobile Environment." Proc. 1st International Workshop on Services in Distributed and Networked Environments, Prague, Czech Republic, June 1994.
- [Duchamp,92] Duchamp, D. "Issues in Wireless Mobile Computing." Proc. Third Workshop on Workstation Operating Systems, Key Biscayne, Florida, U.S., 1992. IEEE Computer Society Press, Pages 2-10.
- [Gittins,86] Gittins, D. "Icon-based human computer interaction" International Journal for Man-Machine Studies 24, 1986, Pages 519-543.
- [ISO,92] ISO. "Draft Recommendation X.901: Basic Reference Model of Open Distributed Processing - Part1: Overview and Guide to Use", Draft Report 1992.
- [Katz,94] Katz, R.H. "Adaption and Mobility in Wireless Information Systems." IEEE Personal Communications Vol. 1 No. 1, Pages 6-17.