HAL
open science

Using Multiple Devices to Access a Cooperative Environment: an Experience Report

Gérôme Canals, Julien Muller

To cite this version:

Gérôme Canals, Julien Muller. Using Multiple Devices to Access a Cooperative Environment: an Experience Report. HCI-IHM Workshop on Multiple User-Interfaces, Sep 2001, Lille, France, 6 p. inria-00107523

HAL Id: inria-00107523

https://inria.hal.science/inria-00107523v1

Submitted on 19 Oct 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Using Multiple Devices to Access a Cooperative Environment: an Experience Report

Gérôme Canals & Julien Muller
LORIA/ECOO
University of Nancy
gerome.canals@loria.fr

INTRODUCTION

This paper presents and discusses an experience in designing and implementing UI modules of the Motu system, a cooperative work environment that uses a multiple user interface on various devices.
We first introduce this environment, the Motu prototype, and underline the motivations for the use of various devices to access the Motu environment.
Then, we present our experience with different kinds of devices. The first device we added to the environment is a PDA, and a specific application and interface was built for this device. We thus discuss the introduction of some other particular devices and the applications we envision for them.

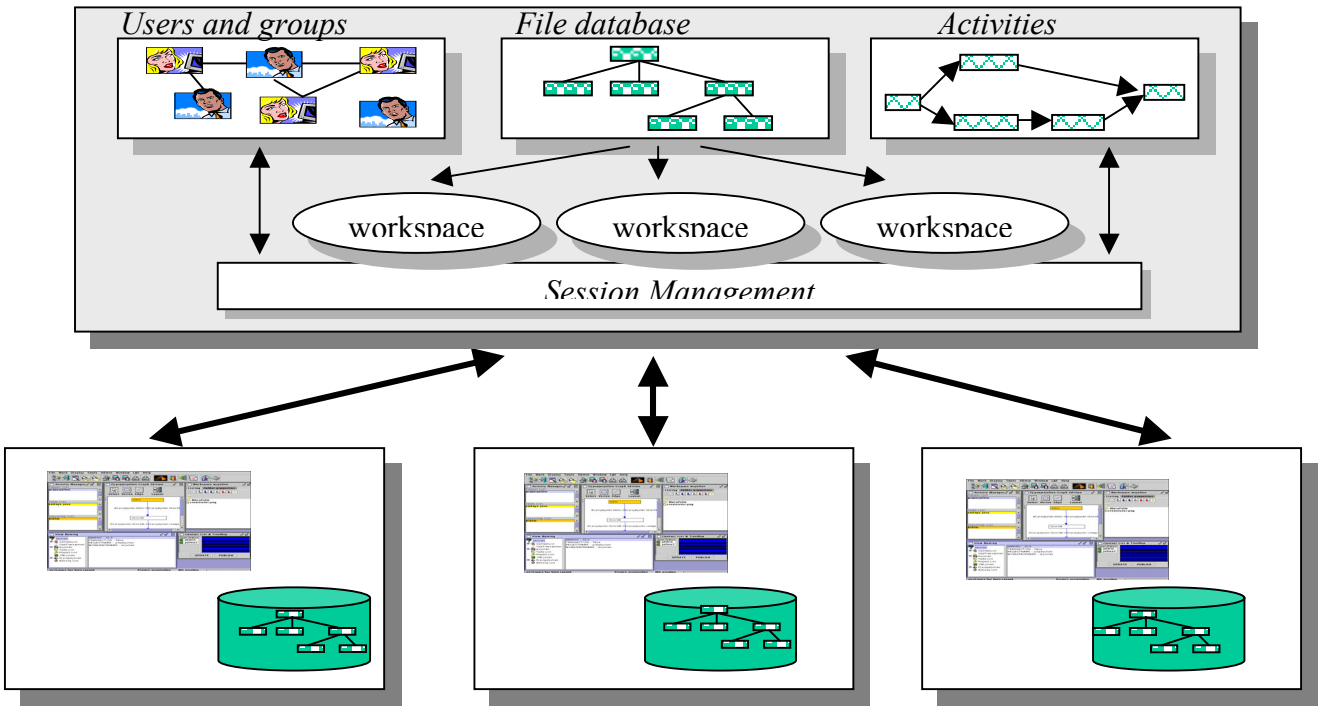
A SURVEY OF THE MOTU SYSTEM

Motu is a system prototype dedicated to the support of distributed cooperative engineering projects. It offers services and tools to help virtual teams at work in carrying out distributed activities in a project.
The core element of a Motu based project support system is a multi-versioned database that manages shared file hierarchies. Participants to a project access their files by checking out them from the file database into a private workspace. These workspaces contain mono-versioned file hierarchies and can be understood as particular projections of the shared database following some user-dependant selection rules.
Since workspaces reside on the Motu server, effective work on files is conditioned by the transfer of the corresponding workspace content onto the participant client environment. The reverse operation, checking in files back to the server, is used to publish modifications done on a file and to render them available to other participants.
Other services offered by the Motu prototype include:

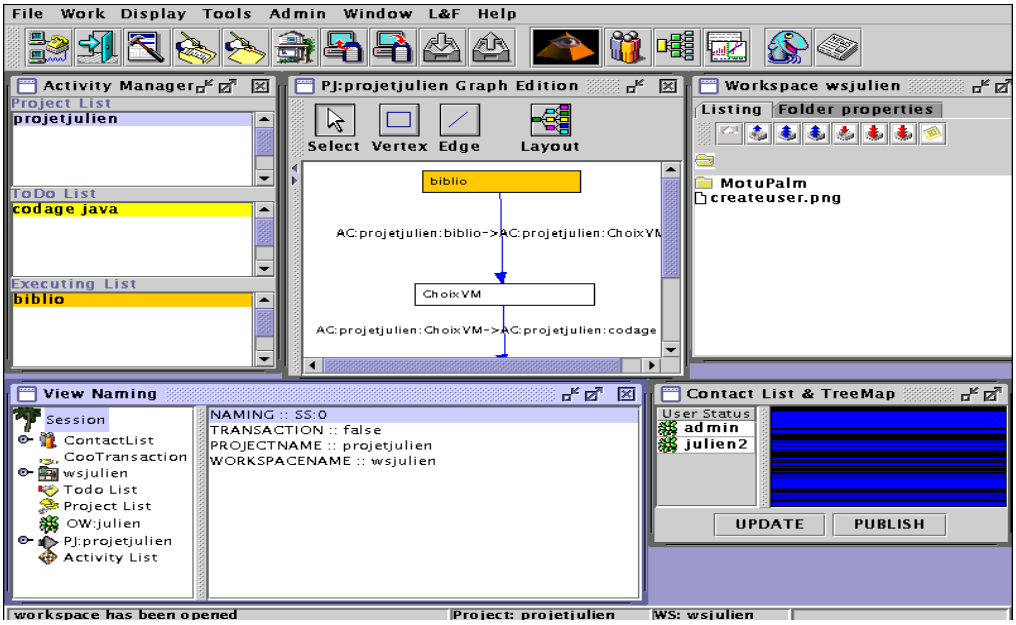
- User, session and group management: participants of a project are identified and collaborative sessions are managed. Based on that, groups of people can be defined and dynamically modified and can reflect various organisational structures. Each user has its own contact list from which he gets presence awareness and direct communication using IM, video and chat tools.
- Notification and state awareness: based on a publish-subscribe model, the Motu system includes a persistent event mechanism that allow users to be aware of what's going on in the environment. In addition, advanced awareness widgets display in real time synthesised information about the state of the shared files (up-to –date, remotely modified, locally modified …).
- Coordination and activity management: the core data management module is connected to a workflow engine based on an advanced cooperative transaction model.

This system that allows the definition and enactment of identified activities, manages dependencies between activities and computes to-do list for the participants.

The architecture of the Motu system is based on a Client/Server. The server is in charge of file and user management, event detection and notification and activity enactment. The client is mainly dedicated to user interface, and support the effective work environment for tool execution and in-use file storage.



The actual Motu server is a complete and dedicated server. However, a new ongoing implementation will be based on the Apache http server. The client can be run as a standalone application or as an applet in a web browser. All exchanges between the client and the server use an XML-based protocol.
The Motu user interface allows users to interact with the server and other participants and



includes tools for activity management, workspace visualisation and navigation, contact list management and various collaboration tools.
Working in a Motu-based project support environment consists in carrying out two main classes of tasks:

- *Production tasks* in which the effective work to be done is realized. They generally consist in creating documents using specialized software tools. These tools depend on the application domain and are independent from the Motu System. They are enacted by the participants on their client machine and operate on files residing on the client file system, but extracted from the Motu server file database.
- *Articulation tasks* in which all the additional work dedicated to cooperation and project management is done. Articulation includes communication with other participants, file and data exchange, activity management … This work is done through the Motu system, by sending commands and receiving events between the Motu client application and the Motu server.

The Motu system has been particularly experienced in the domain of Architecture and Building construction. Projects in this area link together many peoples playing different roles (architects, engineers, quality officers ..) in different organisations, and sharing common data and documents (plans, budget prevision, technical documentation …).
An important point to note is that in such projects people generally spend a lot of time *outside of their office*, being thus unable to access their work and collaboration environment using their habitual desktop.
Thus, an interesting enhancement of our system is to include mobile and specific devices to offer a more continuous access to the project support environment. These devices will support activities carried out outside of the office.
Different devices have been identified: PDAs, cellular phones and meeting support devices (e.g. eBoards) . The first one that has been included is a PDA. Next section describes MotuPalm, the PalmOs version of the Motu client.

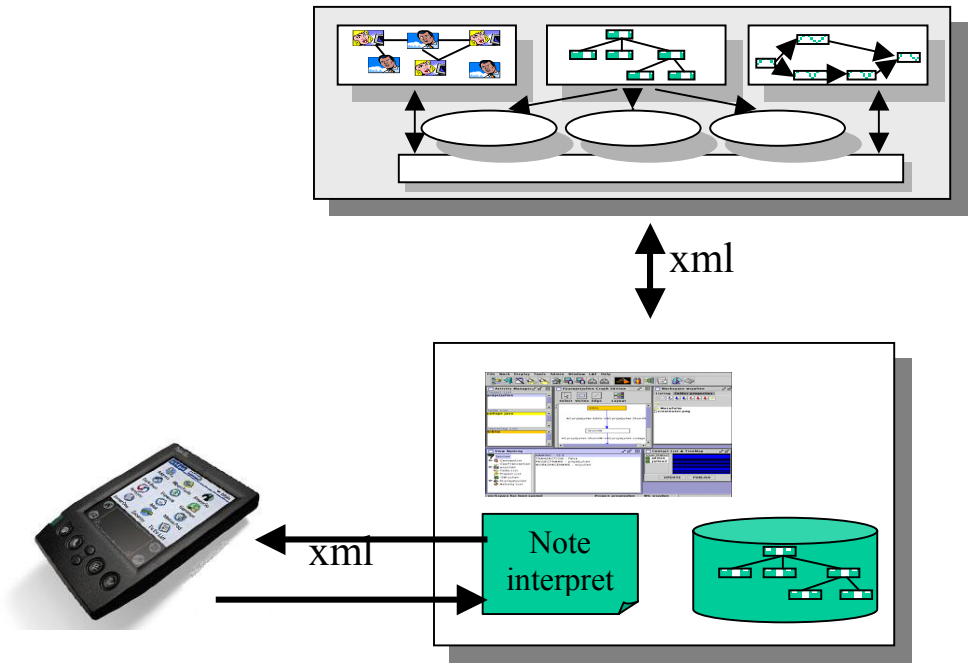
ACCESSING THE MOTU SYSTEM FROM A PDA

A PalmOs based application has been designed and implemented that offers to participants the possibility to interact with their work environment while being away from it.
The following points have considered when designing the functionalities of this new access mode:

- Work done outside the office is rarely devoted to production tasks: there is no real need to effectively edit design documents on a very small device. In contrast, this work is more often dedicated to articulation tasks. Translated in the Motu context, this means that a complete client residing on a PDA is not required. An interface for sending Motu commands may be sufficient,
- A PDA is essentially a *disconnected* device that can be synchronised from time to time with a desktop environment. It should be used in this way.

Considering these points, the application that was designed is basically an advanced annotation system. This mechanism allows users to edit comments, memos and notes and to rely them to elements in is client environment, e.g. files, contact elements, activities …
These memos can in addition contain commands with their arguments.
The Motu client is extended with a note interpreter which is executed when the PDA is synchronized: effective annotation are created on environment objects, memos are displayed and commands are executed.

All exchanges between the Motu client and the PDA are based on an XML protocol.
The architecture of the extended system is the following.



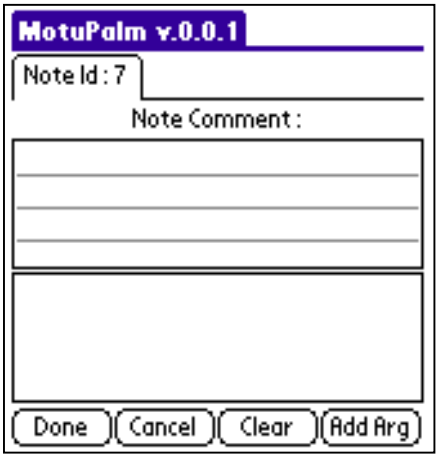
From the user point of view, a PDA session is made of 3 successive steps:

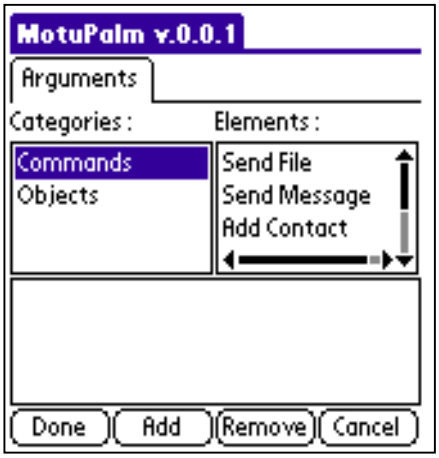
1. *Preparing the session*: the user prepares the PDA session from the Motu client by selecting the objects on which he wants work. These objects cans files, contact elements, activities … These object are requested to the server which sends back an XML description of them (except file contents). This XML document is then transferred to the Palm device.
2. *Carrying the sess*ion: the MotuPalm interface is used by the participant to create notes and reminders. These notes and reminders can be linked to their related objects. They can also contain commands. Example of notes that can be created include the following :
   - A simple annotation on a given file: this is used to store comments on documents,
   - An annotation on a file linked to a sendFile command with an identified recipient extracted from the contact list and the annotated file as arguments.
   - An annotation linked to an ongoing activity and an identified user and including
     - an addActor command to add that user to the activity casting,
     - an sendMessage command to inform the user.
3. *Synchronization*: when the user come back to its habitual work environment and resumes the PDA session, the xml description of the new notes are transferred to the Motu Client and processed by the Note Interpret. This results in the creation of effective annotations, reminders and the execution of the commands.

The MotuPalm user interface is very different from the Motu Client standard interface. However, it uses the same Xml description of the objects in the environment than the Motu Client interface. Some example use of the MotuPalm interface are depicted in the following snapshots.

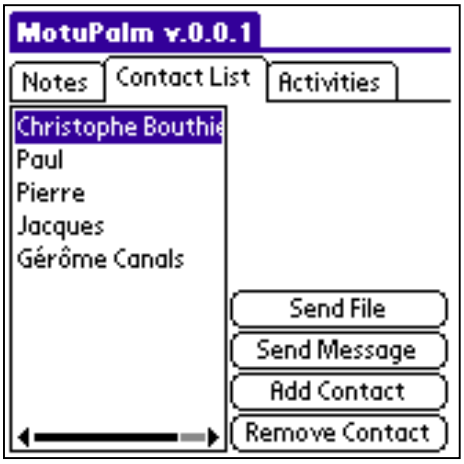
The main MotuPalm window offers a note record that contain all the created notes, and buttons to create new ones, and delete or edit existing ones.

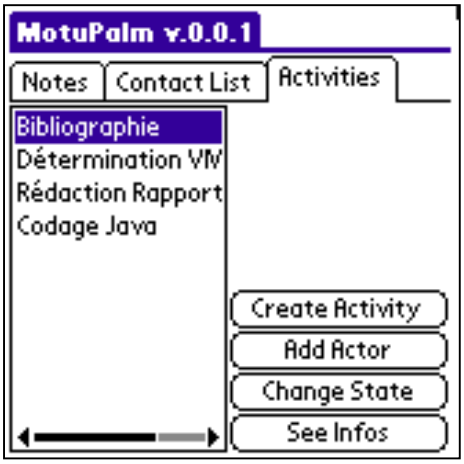
The creation of a new note consist first in editing a short text to annotate an object, comment a command or that will serve as a reminder. Then, an arbitrary number of arguments are added. These arguments can be any Motu command or any Motu objects accessible on the PDA, including the existing notes. The complete structure of the note is displayed as long as the note is being edited.





Although any motu command can be included in a note, the MotuPalm application provides a specific interface for some frequently used commands related to activity management and contact list manipulation :





CONCLUSION

The MotuPalm application has been completely implemented and integrated to the Motu environment. A first experimentation has been conducted. Some lessons have been learned from this experimentation:

- The actual user interface is sometime confusing and need some additional work to simplify it and render it more understandable.
- The fact that the MotuClient and the MotuPalm interfaces are very different seems not to be a problem, since the services offered by the two applications are quite different.
- The important thing is that the services offered should be adapted to the characteristics of the device and should fit the common use of the device.