

the application is actually running and processing, with other participants active at other locations.

The user has some flexibility over the scenario, for example the control panel of a remote camera can be activated to simulate a zoom even if this is not explicitly defined in the scenario. The activity of other users is indicated in the scenario.

#### **EXPERIMENTATION**

This environment has been designed and implemented at the French Navy. It is currently being used to prototype collaborative applications within the control room of a navy ship, a technical teleassistance application and a cooperation between different vessels.

#### **REFERENCE**

1. Roseman, M. and Greenberg, S. GroupKit: A Groupware Toolkit for Building Real-Time Conferencing Applications, Proceedings of CSCW '92 (Toronto, November 1992), 43-50.
2. Cortès Mauricio and Mishra Prateek. DCWPL : A Programming Language For Describing Collaborative Work, Proceedings of CSCW '96 (Boston, USA, November 1996), 21-29.

# An Environment for Prototyping CSCW Applications

**PONS Marie-Chantal**

Ministry of Defence  
Direction des Constructions Navales CTSN/TIRN  
Av. de la Tour Royale, BP 28  
83800 Toulon Naval France  
+33 4 94 16 24 23  
mcp@newton.ctsn.dga.fr

**MERIALDO Bernard**

Eurecom Institute  
2229 Route des Crêtes, BP 193  
06904 Sophia-Antipolis Cedex France  
+33 4 93 00 26 29  
merialdo@eurecom.fr

## **ABSTRACT**

CSCW applications are difficult to design and to implement. In this poster, we present an environment which allows to rapidly create a simulation of a collaborative application and to show it to an user. This environment is based on a high level description language which describes the application and on a simulator which allows to interpret this description according to predefined scenarios. The user can observe the dynamic behaviour of the application as would be seen by one of the participants. He may then rapidly provide comments and suggestions to improve the current design. This environment has been implemented and is currently being tested.

## **Keywords**

Rapid prototyping environment, single-user simulation of multi-user applications, description language of cooperative applications, simulation.

## **INTRODUCTION**

CSCW applications are becoming more and more popular, yet designing new applications to fit in a given workplace is difficult because users are generally not familiar with this new way of working. It is often necessary to prototype - at least partly - the intended application so that they can understand its possibilities and are able to provide comments and suggestions.

Even if toolkits (such as GroupKit [1]) exist to facilitate the development of such prototypes, a working multi-user computer application still requires substantial development effort to be implemented and demonstrated. Languages are a other way to describing applications [2].

We propose an environment where we describe a CSCW application in a high-level language, which is interpreted by a simulator to rapidly provide a view of the application appearance. This simulation can be run on a single machine, according to one or several predefined scenarios. A scenario is the description of a particular sequence of interactions involving a group of people using this simulated application. The simulator is able to show the user the dynamic behaviour of the application as seen by one of the participants.

Since the simulator follows predefined scenarios, no actual computation, data exchange, synchronisation etc.. takes place. Prototyping new applications is thus specially easy because they only concentrate on interface appearance and dynamic behaviour.

In the next section, we briefly describe the LDSC language. Then we present the SISCOP simulator. Architectural designs and interface snapshots are presented in the poster. We conclude with a description of the experiments that are currently being performed.

## **LDSC: A LANGUAGE TO DESCRIBE COLLABORATIVE APPLICATIONS**

LDSC is a scripting language to describe collaborative applications. The language contains seven basic classes: session, activity, actor, role, node, equipment and document. A session is a collection of activities. An activity is a particular type of situation involving actors. Actors are participants of the activity, and their prerogatives are defined by roles. Node, equipment and document are used to describe the properties of physical and electronic objects. A cooperative application will be described by defining one or more instances of these classes. The characteristics of these objects and their relations will be written as values of their attributes (for example the role of an actor) or as rules which define their dynamic behaviour (reaction of the object to a request).

When the application is started, the main session object is created, which may cause other objects (activities, actors, roles ...) to be created. The application evolves as new requests are generated by users through their interface, or by system components (such as timers).

## **SISCOP: A PLATFORM TO SIMULATE APPLICATIONS**

The SISCOP runs on a stand-alone PC and is able to interpret a LDSC script. It creates a view of the user interface as seen by one participant of the application. The application can be 'played' according to a predefined scenario, which consists of a sequence of requests simulating the activity of the participants to the application. Interface widgets simulating audio-video connection, control panels, shared documents, existing applications etc... are used to give the user the feeling that