Circa: Communication appliances tools

Emmanuel Nars
Laboratoire de Recherche en Informatique & INRIA Futurs *
Bât 490, Université Paris-Sud XI
91405 Orsay Cedex France
nars@lri.fr

ABSTRACT

Communication appliance is a new class of communication system. Simple to use, it allows users to communicate together through different groups. Developing, prototyping and testing communication appliances is a difficult task. Existing toolkits are not designed to develop these kinds of communication systems. Moreover these toolkits are complex to use. This poster describes a toolkit designed to build communication appliances.

1. INTRODUCTION

In this poster, we are mostly interested in group-based communication between close-knit groups of people. These close-knits groups of people could contain friends, family. Although these intimate social networks are omnipresent in our daily life, they are very poorly supported by current communication systems. Many communication systems allow us to exchange with people but in a limited one-to-one communication. Exchanging with more than one user quickly becomes tricky. For example, instant messengers like iChat, MSN or Jabber are easy to setup or use to communicate with one member. However, communicating with several members is laborious to set up and reuse. For example, chatting with several users forces MSN users to initiate conversation with a single user. Next, he invites others users to this chat. Many groupware systems and toolkits such as GroupKit [2] use an alternative approach. This approach based on a central server relies on session managers. It is an easy approach for developers to set up shared spaces and manage participants. One user creates a shared space and the others connect to it with a session manager. However, each user must be connected to the same session manager server to participate in a session.

In the interLiving project, technology probes [1] have been developed. These technology probes are very simple communication devices with one or two functionalities. The feedback collected with these probes pushes us to develop a new class of devices defined as *communication appliances*. As technologies probes, these devices are simple to use and limited to one function. Development of such tools faces us to the same problems met along the technology probes development process. Creating and maintaining dynamic connections between probes in different places was a real challenge. Probes irregularly connect and disconnect to Internet and groups topology on applications frequently change. Moreover existing middleware toolkits to build groupware applications do not fit requirements of communication appliances developers. Group communication toolkits [3] are too complex and do not provide enough privacy and flexibility to build communication appliances.

2. OVERVIEW

Circa is an infrastructure design to build, use and manage communication appliances. This infrastructure is mainly based on the concept of groups communication. A group is a set of users defined symmetrically for each member. Members inside groups can exchange or communicate together with communication appliances. Communication appliances can use groups in an asymmetrically way. For example, one application could be connected to applications through groups containing different members. Therefore, all applications are not connected to each other (Figure 1)

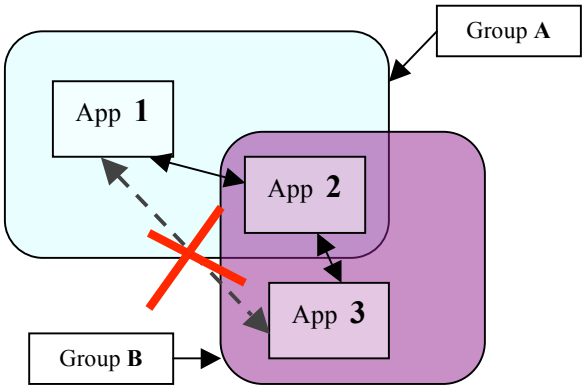

Figure 1: Applications 1 and 2 are connected because they belong to group A. Applications 2 and 3 are connected because they belong to group B. However, applications 1 and 3 are not connected because they do not belong to the same group.

The development of Circa infrastructure follows three guidelines. The first is to respect as well as possible users privacy. In many systems, users must store private information on commercial company servers starting from a mail address to your entire data (http://www.forever-safe.com). This is the reason why we choose a peer-to-peer solution. Thus, all private information or data are stored on our own server in our house. The only shared information is the groups topology. It is shared with every group's members. As they are friends or family, that is not a problem. Our second aim is to provide the developer with a simple toolkit to build communications appliances. This toolkit is designed to be quickly adopted by programmers and even by designers having programming bases. Finally, we provide users with a zero configuration installation process. Thus everyone, from a 4 years child to an 80 years grandmother will be able to use Circa infrastructure and specially communication appliances.

---

* projet In Situ, Pôle Commun de Recherche en Informatique du plateau de Saclay, CNRS, Ecole Polytechnique, INRIA, Université Paris-Sud

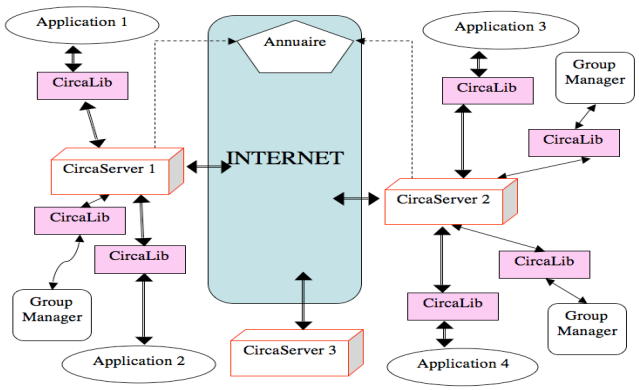

Figure 2: Sample configuration of Circa Infrastructure

Circa infrastructure (Figure 2) is implemented in C#. C# language allows us to provide the developer with a single library (*CircaLib*) for PC or mobile device like PDA or Smartphone in order to prototype communication appliances of different sizes. The *CircaServer* is the heart of the infrastructure. CircaServers allow users to find each other and exchange data through applications. CircaServers forward applications requests to other CircaServers and notify local applications when new data are available for them. Application or communication appliance is the tool that users use to communicate. During the infrastructure development, we have developed some applications such as a white board, a chat room or a videoconference system in order to test the infrastructure. A specific application called *GroupManager* has been developed too. It allows users to manage their groups and their applications. A centralized *meta server* acts as a name server for all the CircaServers, so they can locate each other. It is the only part of the infrastructure that is not distributed (Integration of a distributed name server [4] is also being investigated).

3. DEVELOPER'S VIEW: CIRCALIB

Building a communication appliance is a frustrating experience. Setting up links between applications, managing links between applications and taking into account network issues is tricky and is not the goal of communication appliance developers. Circa manages interconnections between communications appliances and provide developers with a limited set of services. CircaLib allows storing files, sending or getting data without taking into account connected applications and groups topology. In order to show the simplicity of CircaLib, we will illustrate through an example how to use CircaLib in the building process of a videoconference application (Figure 3). This example obviously shows how to exchange data between applications without any knowledge in network programming. Developers have just to initialize CircaLib with the name and the type of the application and Circa infrastructure undertakes everything.

```
CircaLib CL = new CircaLib("VideoConf", "Circa::Video");
CL.Initiates_MultiStream();
CL.send(webcam_picture);
for (int i=0; i<CL.nb_Apps_connected; i++)
{       byte[] pict_buffer;
        CL.Recieve(pict_buffer, i);
        Display(pict_buffer,i)   }
```

Figure 3: Example of code to create a videoconference application with CircaLib

4. END-USER'S VIEW

Circa infrastructure provides the user with a personal server, a set of communication appliances and a personal application called GroupManager. Each user possesses his own GroupManager. It is a kind of remote control for his communication appliances. User can use GroupManager to create groups and to invite others users in his groups. Several groups and users can be set on any communication appliances. This means that two users with two GroupManager can use the same communication appliance with different groups at the same time. GroupManager is a software version of group managing interface. In another paper submitted to Soups'05 (FamilyNet: A tangible interface for Managing intimate Social Network), we describe how to use a tangible interface based on cards containing RFID tags to manage groups on communications appliances.

5. PERSPECTIVE & CONCLUSION

In this poster, we have presented Circa, a software infrastructure designed to facilitate programming and the use of communication appliance. One of our guidelines is to preserve users privacy and to create a secure infrastructure to store and exchange users' data. In a first time, we focus on creating a stable and easy-to-use infrastructure. This infrastructure is not yet secure because communication between different modules is not crypted. However, we think that it will be easy to integrate asymmetric cryptosystems because each object of our infrastructure is tagged with an unique id. These ids could be the keys needed by these kinds of systems. In a second time, we want to extend Circa services to provide developers with a set of basic tools (i.e. webcam tools, whiteboard tools, chat tools ….) which could be reuse to build communication appliances.

6. REFERENCES

[1] H. Hutchinson, W. Mackay, B. Westerlund, B. Bederson, A. Druin, C. Plaisant, M. Beaudouin-Lafon, S. Conversy, H. Evans, H. Hansen, N. Roussel, B. Eiderbäck, S. Lindquist, and Y. Sundblad. Technology probes: Inspiring design for and with families. In Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems, pages 17-24, April 2003. ACM Press

[2] M. Roseman and S. Greenberg. Building Real Time Groupware with GroupKit, A Groupware Toolkit. ACM Transactions on Computer-Human Interaction (TOCHI), 3(1):66-106, March 1996. ACM Press

[3] Ohad Rodeh, Ken Birman, Danny Dolev. The Architecture and Performance of the Security Protocols in the Ensemble Group Communication System. Journal of ACM Transactions on Information Systems and Security (TISSEC).

[4] Xiaozhou Li, C. Greg Plaxton. On name resolution in peer-to-peer networks. In proceedings of the second ACM international workshop on Principles of mobile computing, pages 82-89, October 2002. ACM Press