CoLoS -A SYSTEM FOR DEVICE UNAWARE AND POSITION DEPENDENT COMMUNICATION BASED ON THE SESSION INITIATION PROTOCOL

Odej Kao and Stefan Lietsch
University of Paderborn Paderborn
Center for Parallel Computing
Fuerstenallee 11 33102
Paderborn, Germany
{okao, slietsch} @uni-paderborn.de

Abstract In this paper we present a new system that allows users to communicate easily and comfortably. It supports users by integrating several services such as location awareness and device independence. The user is for example not bound to one device nor must he know the exact address of the desired communication partner. He is also supported by all the information the system has or can generate from all of its services. The Session Initiation Protocol is used and extended in order to provide a scalable, secure and efficient platform for the communication system and its services. The basic functionalities are shown in a prototypical implementation.

1. INTRODUCTION

Nowadays communication means more than just talking to each other or chatting. The focus lies on exchanging information where neither the distance nor the nature of the communication partner is an issue. There are many ways to communicate with people but also with machines e.g. computers and there are more to come with growing global inter connection. Two main problems arise from this development. On the one hand a user needs several devices to use the different ways of communication and on the other hand the exact address of the opponent needs to be known to establish a connection through a certain media. In addition the user must know or at least try out the media-related address on which his partner is reachable.

To solve these problems a system with two main functionalities is desirable: On the one hand, it should act as a translator between the different types of media and on the other hand it should provide something like an extended phone book in which all communication relevant and additional data is stored. These two components enable a user of the system to communicate with every potential partner without knowing his exact address nor being bound to one specific device. Furthermore, the system can merge all the data from different types of media it has and possibly gets additional input from e.g. positioning devices. So it can generate useful information to support the user. By combining these two aspects (uniform communication and user support) we created a system that extends the existing ones presented in chapter 2 to be a modular, scalable, secure and user centered communication platform.

This paper is structured as follows: After the Introduction and Motivation in section 1, we give an overview on existing communication systems and comment on location applications (section 2). In section 3 we outline the difference between the existing systems and our concept and present some possible features. Afterwards we describe the concept of the CoLoS and its components in section 4 and subsequently we go into the prototypical implementation of our concept (section 5). Finally we present a brief conclusion and point out which future work needs to be done.

2. RELATED WORK

Recently different approaches towards personalized, device independent and user supporting communication systems were proposed. There are two main research directions: One aims at maximising reachability independent of the used device. The other direction focuses on the personalization of the system. We aim to integrate both directions and add a new aspect namely the localization of the user. In the following we introduce three representative systems and show their advantages and problems and afterwards give a short comment on the localization aspect.

The Mobile People Architecture (MPA) described in [Maniatis et al., 1999] is the base of most of the device independent communication systems. It is the first development of an open concept for a device independent communication application ensuring best possible reachability of its users. By extending the traditional layer model by a Personal Layer the reachability is mainly achieved. This layer represents the person itself by managing all owners devices and their reachability. Thus communication requests no longer go directly to the devices but to the user (resp. to his representative, the so-called Personal Proxy). This proxy decides how to proceed with the incoming request, routes it to the corresponding device and manages the communication. This so-called Personal Level Routing is the main point of the MPA, but it brings one bottle-neck namely the Personal Proxy. All the communication has to go through the proxy although there could be a better or faster way. This problem is being solved in the Iceberg Architecture [Wang et al., 2000]. The system is based on the MPA but has one major difference: it no longer has decentralized proxies for every user but concentrates many proxies in centralized units called Iceberg Points of Presence (IPoPs). These IPoPs have interfaces to many access networks (e.g. telephony, cellular, internet) and are interconnected by fast network connections. This ensures that all communications can be routed on a fast and direct way. There is also a billing unit in the system to charge the users for certain services. In conclusion the Iceberg Architecture is a highly developed system to enable device independent communication but it does not support the user beyond this functionality and is limited to communication services.

The Integrated Personal Mobility Architecture (IPMoA) [Thai et al., 2003] takes a slightly different approach to ensure the device independency and mobility of the user. It does not primarily focus on the reachability but on the mobility of its users. The user is able to access all his data and applications from every remote location and with every available device. By including communication applications a device independent communication is possible. The whole system is based on agents that commute between the home and the foreign network and exchange the data between them. Thereby a high level of personalization can be reached but since nearly all data must be fetched from the home network it may have problems especially with time-sensitive and synchronous applications, respectively.

In the field of localization techniques there are many different approaches. One main research field is localization by determining the positions of all kind of mobile communication devices (e.g. GSM phones or WIFI devices) as proposed in [Youssef et al., 2003] and [Zimmermann, 2001]; another research direction is to use proprietary short range radio techniques based on bluetooth or infrared to locate the users of the system as exemplarily proposed in the Active Badge System [Want et al., 1992]. Since we want to support as many different localization systems as possible we don't want to commit ourselves to one technique or direction. We plan to integrate the positioning applications independent from the underlying mechanisms. This is why we just reference to some exemplary systems and focus on supplying an extendable communication and localization platform.

3. WHAT'S NEW?

All the systems presented above have different kinds of information about its users e.g. reachability, different addresses etc. This information is used to provide the functionalities of the communication systems. Our approach is to take all the data gathered by the different communication systems, add some additional data e.g. from location systems, and generate information that supports the user far more than possible with existing systems. Thereby we enable a platform for device independent communication and its personalization which combines the advantages of both directions presented in Section 2. Some possible services of this combined system are:

Find communication partners in your proximity. One self-evident service is to announce possible communication partners or friends which are detected near the location of the user. Possible scenarios for this service could be exhibitions where interesting exhibitors nearby are indicated to the user or the sign-posting to one specific person.

Discover the cheapest or fastest connection. Another possible service is to suggest the best and/or cheapest network connection for the users current position. Furthermore automatic connection handover mechanisms are possible.

Hints on services close to the user. The system can point nearby services out to the user. This could be communication services like a locally bounded NetMeeting conference or non-communication related services like a public printer or fax machine.

To make sure that the services above and other new ideas work properly, are accepted by the users and don't cause security problems some architectural requirements must be fulfilled: Extendable and easy to integrate, Hardware independent and portable, Transparency, Easy to use, Optionality and privacy, Security of the data.

We will introduce some exemplary functions and features of the proposed Communication and Location System (CoLoS). There are more to come since the platform is expandable and scalable. One function we already mentioned is the communication with users whose addresses are not exactly known. Our system finds the desired user by any known information (e.g. name, email, preferences) and gives choices if more than one match exists. Another feature is the communication with incompatible devices. That is that two people with different devices (e.g a mobile phone and an instant messenger) can communicate without recognizing the incompatibility. The third and very important feature of our platform is the configurable information search engine. It can access all the data gathered by the different systems and search for useful information to support the user by applying different rules on the database. In this way, all information available in the common database can be used to find new information that supports the user. This approach removes the barrier of incompatible storage used for the different communication services. We are now able to use all available resources to provide valuable information to the user.

4. CoLoS CONCEPT

Figure 1 shows the two sides of the developed system and their main components. The CoLoS client allows the service utilization and notifies the user about new incoming information. This is provided by a GUI fitted to the particular device. Already existing applications are integrated through the so-called Client Interface which can be seen as an universal interface. The data exchange between server and client is handled by the CoLoS Connection on the client and the Controller on the server side whereas the data is transmitted in packets of the CoLoS/SIP protocol. One of the main server components is the User Register where all the data is stored in a fast database. This data can be accessed by the Decision Engine to combine it following pre-specified rules in order to create useful information. The last main component is the Communication Dispatcher and its Communication and Translator Modules. It enables the system to translate between incompatible types of media transparently. In the following we describe the main components of the CoLoS server and client.

4.1 Server Modules

The **Controller** is the main server component. It decides by the type of an incoming message how to proceed with it. It passes the information of a message including a location update to the User Register in order to update the database. The Controller also decides by means of the data from the User Register by which device a user is reachable and if an incoming communication request can be fulfilled (with or without translation).

The **User Register** can be seen as an extended phone book. It stores all user data irrespective of its origin. Some exemple data are: addresses for each communication device, availability and preference of communication ways, current location, profile of the user, a buddy list, public key etc. The list of stored data can be extended arbitrarily to cover all useful information about the systems users.

The **Communication Dispatcher** is invoked by the Controller if a user requests a communication with another user who is not reachable under a compatible device. That is for example if user A wants to use an IP-Phone and user B only has his ICQ Messenger enabled. The controller recognizes this incompatibility and passes all messages concerning this communication to the Communication Dispatcher. It determines the corresponding Communication and Translation modules (in this case an IP-Phone and a messenger interface and a text-to-speech/speech-to-text translator), translates the messages and passes them back to the controller. Afterwards the messages are sent to the receiver under the address of the initial sender so that the whole translation action is transparent for the users.

The **Decision Engine** tries to find out which services can be offered to the users. This is a major functionality of the CoLoS since all available data is taken into account. It therefore applies rules on the User Register after every change in the database to look for new and useful information. One possible rule could be: "Check if a user in the users buddy list is in his proximity after he changed his position". If one or more matches are found the result is passed to the Controller. The Controller then generates messages containing this information and sends them to the correspondent users. While the database is growing and is changed more often with an increasing amount of users new mechanisms must be found to restrict the search to concerned fields of data. For example a strategy based on the current location of the user is thinkable. Additionally rules can be defined and added while the system is running.

4.2 Client Modules

The **CoLoS Connection** is the correspondent to the Controller on the server side. It gets commands and communication data from the GUI or the Client Interface, packs them into CoLoS Protocol packets and sends them to the Server. When a new packet is received the CoLoS Connection decides by its type what to do with it. For example an incoming message with the type "User Information" is passed to the GUI where it is displayed correspondingly.

The **Client Interface** is the bridge between existing communication and location applications and the CoLoS. The Interface notices which of the registered applications are started and announces this through the CoLoS Connection to the server. Furthermore it intercepts connection requests to pass them to the server for further processing. Incoming messages are handed over to the corresponding application and the whole communication process is monitored for faults and interruptions to enable a quick solution. Since the Client Interface acts transparently the user can continue to work with his applications as usual but also has the advantages of the CoLoS. The interface can also be utilized to get Location data from positioning applications. This functionality must be controllable by the user at all time to avoid an unwanted surveillance.

By the **GUI** a user can access all functionalities offered by the CoLoS easily and quickly. It for example alerts him on incoming requests or offers a "friend list" where he can save contacts he often uses. Also searching and security functions are integrated. All settings made by the GUI are sent to the server and stored in the user Registry.

4.3 CoLoS Protocol

The data exchanged between the CoLoS server and the clients is, as mentioned above, encapsulated in packets of the CoLoS Protocol. This ensures an efficient and secure data transmission. The CoLoS Protocol is implemented as an overlay protocol based on TCP/IP / SIP. It has several control fields (e.g. sender and receiver address, type of the message, additional options) and one optional data field.

4.4 Exemple CoLoS Interaction

For a better understanding of the system and the interaction of its components we describe one example process in the CoLoS. The scenario is that user A has the CoLoS client and a positioning device enabled and changes his location. An other user B, who is on As friends list, is nearby As new location. User A is notified and can chose between different options.

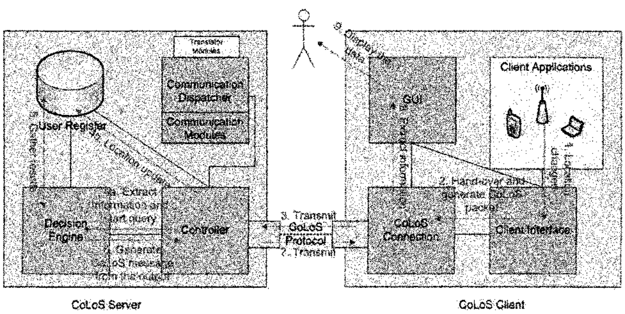

Figure 1. Processing of location updates

This process is depicted in Figure 1 and takes place as follows: The localization device realizes the change of positions and passes this information to the Client Interface (1.). The Client Interface gathers it and generates location update messages in fixed intervals (2.). This avoids an overload of the system due to too fast or too many location updates. The messages are passed to the CoLoS Connection where they are packed in CoLoS Protocol packets and transmitted to the Controller on the server side (3.). The controller receives the packet and extracts the type and the information included. Due to the type of the message the Controller decides to initiate a query through the Decision Engine (4a.). Simultaneously the position update is send to the User Register (4b.). The Decision Engine selects the rules corresponding to the received information and applies them to the User Register to search for helpful information (5.). In this case the rule "Find friends of User A that are closer than 50 m" could be selected and used to start a query. If the query has one or more results, this information is passed back to the Controller (6.). There a CoLoS Protocol packet is generated and transmitted to the client (7.). The ColoS Connection extracts the data and analyzes the type (8.). Afterwards the information is passed to the GUI where it is visualized and different choices of actions are presented to the user (9.).

This process is already implemented in the Prototype, as described in the following paragraphs, and it works well in a closed environment. It is ideal for the interaction in the CoLoS system.

5. CoLoS PROTOTYPE

To realize the components and functions of the presented concept we decided to base our system on the Session Initiation Protocol (SIP). This gave us the chance to use some of the well-tested and approved functions of the SIP and extend it to our needs. The message transmission mechanisms for example fully satisfy our requirements and already have security and recovery functions built in. Moreover, many communication applications already implement the SIP and can therefore easily be integrated into our platform. More information about the Session Initiation Protocol can be found in [Rosenberg et al., 2002].

As shown in Figure 2 we integrated the CoLoS components into the SIP architecture. Our system uses SIP to transport its messages, takes advantage of its routing algorithms and ensures the safety of the message transmission through its security mechanisms. The server side of the CoLoS is combined with the SIP-Proxy-Server and the Client uses the SIP-User-Agent to send and receive its messages. In both, the client and the server parts of the SIP, messages concerning the CoLoS are recognized and forwarded to the corresponding CoLoS component. This is where the content of the message is processed and if needed a response is generated and sent back using SIP mechanisms. Since the SIP-Proxy-Server is designed only to forward requests, we developed an additional component called SIP Client Simulator that simulates the server to be a client. This allows the CoLoS server to send unsolicited messages to the users device to announce, for example, a friend in his proximity.

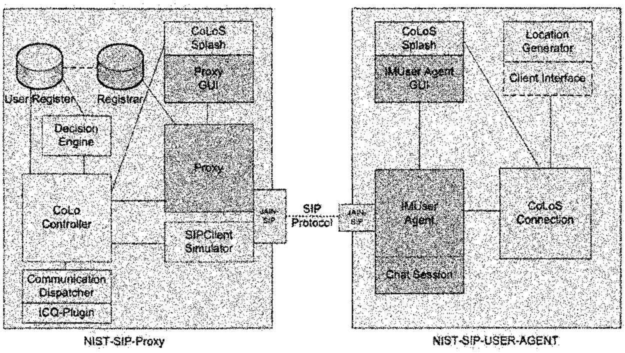

Figure 2. Realization of the CoLoS using the SIP

Our prototype is designed to show the basic features of the CoLoS. It already has an interface for ICQ/AIM Messaging, the SIP Messenger and a simple Location Generator that simulates a localization device connected through the Client Interface. On the server side we implemented the Communication Dispatcher with modules to translate SIP Messenger messages to ICQ messages and reverse, a Decision Engine that can generate information and pass it back to the Controller and a User Register utilizing a MySQL database which interacts with the registrar service of the SIP Proxy. To exchange data between client and server we developed a data format called CoLoS Protocol that is embedded in the SIP message. This encapsulates all CoLoS relevant data. The prototype was tested in different environments and gives a glimpse on what a complete system is able to do.

6. CONCLUSION AND FUTURE WORK

In this paper we proposed a design for an integrated and user supporting communication system. We analyzed the requirements and came forward with proposals for possible services. Regarding the current, fast development in communication technologies we designed a system that is not limited to existing applications. In fact we developed an open platform which is able to integrate existing and future technologies and to link them seamlessly. This is achieved with a modular structure and the use of standardized and application independent protocols. Thus the system can be extended by arbitrary applications to serve its users as a transparent, easy to use and secure communication base.

The prototype we presented was implemented to demonstrate the main features of the CoLoS. In a limited surrounding we can claim the CoLoS prototype works well and efficient. Since the SIP is already tested and approved in large-scale network environments we only have to test and scale the CoLoS specific features.

In the future the simplification and personalization of communication will gain more and more importance since the number of ways to communicate increases steadily and many users do not want or simply cannot take care of the maintenance of all the media. Additionally, negative aspects of the expanding communications world, such as unwanted spam, could be effectively fought by intelligent communication platforms as proposed in this paper.

References

Maniatis, P., Roussopoulos, M., Swierk, E., Lai, K., Appenzeller, G., Zhao, X., and Baker, M. (1999). The mobile people architecture. *Proceedings of the USENIX Symposium on Internet Technologies and Systems, October 1999.*

Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and Schooler, E. (2002). Rfc 3261 - sip: Session initiation protocol. Standard, IETF.

Thai, B., Wan, R., Seneviratne, A., and Rakotoarivelo, T. (2003). Integrated personal mobility architecture: A complete personal mobility solution. *ACM Mobile Networks and Applications (MONET) Special Issue: Personal Environment Mobility in Multi-Provider and Multi-Segment Network, Vol 8, Iusse 1, Feburary 2003.*

Wang, H., Raman, B., Biswas, R., Chuah, C., Gummadi, R., Hohlt, B., Hong, X., Kiciman, E., Mao, Z., Shih, J., Subramanian, L., Zhao, B., Joseph, A., and Katz, R. (2000). Iceberg: An internet-core network architecture for integrated communications. *IEEE Personal Communications (2000): Special Issue on IP-based Mobile Telecommunication Networks.*

Want, R., Hopper, A., Falcao, V., and Gibbons, J. (1992). The active badge location system. *IACM Transactions on Information Systems, Vol. 10, No. 1, January 1992, pp 91-102.*

Youssef, M., Agrawala, A., and U.Shankar (2003). Wlan location determination via clustering and probability distributions. *IEEE International Conference on Pervasive Computing and Communications (PerCom) 2003.*

Zimmermann, R. (2001). Lokalisierung mobiler geraete. *Seminar Mobile Computing ETH Zuerich 2001.*