A Context Based Data Management System for Pervasive Computing Collaborative Applications

Sharat Khungar and Jukka Riekki

Department of Electrical and Information Engineering and Infotech Oulu
P.O.BOX 4500, 90014 University of Oulu, Finland
{firstname.lastname}@ee.oulu.fi

ABSTRACT

We present a contextual storage system for pervasive computing collaborative applications that facilitates applications to tag contextual information such as user's activity, location, and time to the documents. Unlike previous efforts, our system provides explicit support for group activity and access rights of data to support cooperative work. By using the context, the system provides greater flexibility in organizing documents. It allows the sharing of information between different users of a group by using the context related to them.We describe file browser and instant messaging that we have developed on top of our context based storage.

Keywords

Context-aware, Data Management, Pervasive Computing

INTRODUCTION

Context-aware systems are computing systems that provide relevant services and information to users based on their situational conditions [1].In order for computing to be invisible to users while supporting more and more applications, the data required for these applications must be reliably and efficiently stored, queried and delivered.To address above issues we have created a Context Based Storage (CBS) system as part of CAPNET [2], an architecture that supports building context-aware cooperative applications for mobile users.

The design and implementation of the storage system presented in this paper contributes to the research of ubiquitous computing by integrating contextual information into the data storage and distributing this information so that it can be accessed on a mobile device, wherever and whenever needed. The CBS provides provide a simple way to store documents using the context explicitly provided by the user linked to the specific document or to his own current context. It then allows users to retrieve documents from a ubiquitous storage using the context related directly to the document or context related to the user.

RELATED WORK

The ParcTab [3] allows access to files that were linked to a particular location. As users moved between different locations, the file browser would change to display relevant data. While they only considered location in their file system, this work was important in establishing the relevance of context in data access.

The Stick-e document framework [4] describes information in SGML format that includes data and context information. When a specified context matches an available stored document, a trigger makes a data available. However, our approach is different in that we don't use proactive retrieval based on contextual changes in our system, but rather use interactive retrieval to allow user to choose what action to take.

Gaia CFS [5] bears close resemblance to our system but the focus in Gaia is to simplify locating data important for automatically launched applications. They have emphasised more on the data adaptation on the device and organizing data in directories and file system. In our system special emphasis is given to group activity and access rights of data.

CONTEXT BASED STORAGE

The above systems use context about the user, context linked to specific documents and time. None of the above systems, however, directly support all three forms of context with group activity and access rights of data. We have created a Context Based Storage (CBS) that combines these systems into one coherent system. We provide attributes linked with documents, to create a storage space along with a facility to log context history.

CBS consists of a logical context data model and a physical data storage space. The *logical context data model* is a way of representing context information. The context information itself is represented using four concepts: *Entities* are simply people, places, and things. *Attributes* describe some property of an entity. *Relationships* are special kinds of attributes that point to other entities. *Groups* are a way of grouping existing entities, and represent one way in which more sophisticated representations of context can be modelled.

The structure of the data storage in CBS comprises (in a logical view), parameters, files, directories, types, users and groups, and has the following features: (i) Two kinds of data: parameters (for context) and files, (ii) Parameters and files are owned by users, (iii) Users can be grouped in Groups, (iv) Groups can have parameters, (v) The set of

parameters and files owned by user can be organized into directories and on the basis of their types.

Documents are retrieved with a query that can contain the document attributes, information about time as well as context history. For example, retrieve all files for location == TS387 && activity == meeting && time == 9:30-10:30. While the above examples illustrates that queries support AND Boolean operations, OR queries are supported by attaching different contexts to the same document.

The access rights mechanism for the user's data deals with the rights to read or write to CBS user files or parameters. It works similarly to UNIX, with three kinds of access zones: owner, groups and others. Each piece of information is owned by a single user and this user can belong to several groups (each one with independent access rights).

PROTOTYPE

We have implemented a set of applications on top of CBS as part of CAPNET system. Here, we describe only two applications in detail, file browser (FB) and instant messaging (IM).

FB is a graphical data explorer used to manipulate documents with functions to open, read, write, close, delete and rename. If the user is a member of certain group then he is granted access to the files of that group, this makes it easier for users to share files while storing them at one place. To share a file between many users in a group, the owner grants access rights to that file to other member of that group. The users can view these files based on the context attached to them or by performing a query without context-aware option that lists all the documents accessible to the user (Figure 1). FB application supports storage of documents at one ubiquitous storage instead of transferring the same file to many users.

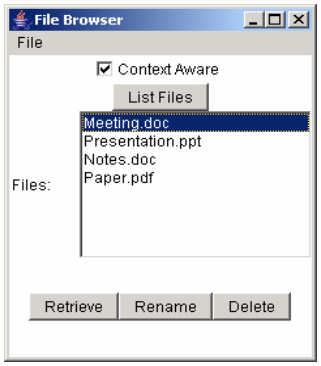

Fig.1. View from the File Browser application

The context-aware instant messaging (IM) system [6] developed on the existing CAPNET architecture utilises the group activity (Figure 2) and access rights mechanism of CBS. It allows the users to communicate with other users based on their context. The key idea in IM is to utilize both the communication initiator's and receiver's context in inferring presence. The users organize their contacts in groups and their presence is updated automatically. A user can be added as 'co-worker' in one group and 'family' in another based on the relation between the users.

This prototype has been tested by researchers, and during the experiments a mobile device was used (Compaq iPAQ PDA). All mobile device components are implemented according to Personal Java 1.2a specification. Context-based storage utilizes the MySQL database, which is running on a Sun Solaris server. The university's premises covered by WLAN and positing system are used as the test

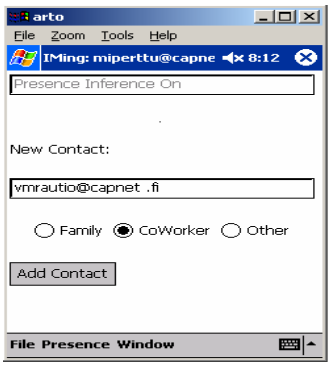

Fig.2. View from the IM application

environment. The graphical user interfaces are described as extended XUL scripts that are rendered as AWT components in the PDA.

CONCLUSION AND FUTURE WORK

We have presented a contextual storage system and two applications built on it that use context to support user's work. Unlike other systems, our system provides explicit support for group activity and access rights of data to support cooperative work.

Now, we are improving the system based on the evaluation such as user's satisfaction about context-aware collaborative applications and system faults. Future work involves developing more applications to support collaborative work and proactive retrieval of data.

REFERENCES

1. Salber, D., Dey, A., Abowd, G.: The Context Toolkit: Aiding the development of context enabled applications. Proc CHI 99 Conference on Human Factors in Computer Systems, Pittsburgh, PA, 1999, pp. 434-441.
2. Sauvola, J.: Capnet Project. http://www.mediateam.oulu.fi/projects/capnet/
3. Schilit, BN., Adams, N., Want, R.: Context-aware computing applications. Proc. IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, 1994, pp. 85-90.
4. Brown, PJ.: The Stick-e document: a framework for creating context-aware applications. Proc Electronic Publishing 1996, pp. 259-272.
5. Hess, CK., Campbell, RH.: An application of a context-aware file system. *Pervasive Ubiquitous Computing*, Vol. 7, No.6, 2003, pp. 339-352.
6. Perttunen, M., Riekki, J.: Inferring Presence in a Context-Aware Instant Messaging System. To appear, INTELLCOMM 04, Bangkok, Thailand, November 2004.