PCSM: A Context Sharing Model in Peer-to-Peer Ubiquitous Computing Environment

Jian Ye¹², Jintao Li¹, Zhenmin Zhu¹, Xiaoguang Gu¹, Hongzhou Shi¹
¹(Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100080, China)
²(Graduate School of Chinese Academy of Sciences, Beijing, 100049, China)
E-mail: jye@ict.ac.cn

Abstract

This paper proposes the Peer-to-Peer Context Sharing Model (PCSM) which is a ubiquitous computing oriented peer-to-peer context sharing model. Owing to the distribution and limit of resources of the mobile ubiquitous network, the PCSM model constructs a context management framework based on the mechanism of registration-query. Through designing the broadcast messages for available terminals discovery and unicast messages for remote registration, the model implements remote registration of Context Database Agents, and builds up an architecture for sharing and fusing of contexts in a peer-to-peer ubiquitous computing environment to satisfy the requirements of sentient computing applications. In contrast with some current context management systems, the PCSM model partly solves the inherent problem of high cost for communication and storage redundancy through the mechanism of optimization. Besides, the PCSM model enables computing nodes to share contexts, on which the decisions depend. Based on the PCSM model, a prototype of Instant Messenger with ubiquitous computing ability is provided in the end.

1. Introduction

The peer-to-peer computing is a common architecture of computing of the ubiquitous wireless network environment. In such a ubiquitous wireless network, some applications which can change their behaviors based on a model of the environment they construct using sensor data, are called sentient computing applications[¹]. These sentient computing applications are characterized as implementation of fusing of information and physical space, context awareness and interoperation between terminals and peripheral ubiquitous system. Sentient computing applications will form a more intelligent environment for service running to provide users with the best experience of ubiquitous computing.

In the face of the requirements for decisions of sentient computing applications, network terminals in the ubiquitous wireless network should have higher intelligence of flexible change according to such contexts as network topology, user’s preference, performance of terminals, etc. At the same time, in a peer-to-peer ubiquitous environment, contexts are distributed and managed in respective terminals, which is different from traditional computing architecture with a central server. How to build up the channels between sentient computing applications and distributed context information is an awkward problem to solve. The construction of the channels involves the sharing of contexts among terminals and presentation of distributed contexts to applications.

1.1 Related work

The pioneering work in the architecture of context management is divided into centralized model and peer-to-peer model.

(1) Centralized model

In a system with centralized model, contexts are managed by a central server from which the sentient computing applications are able to get related contexts. There are many projects with centralized model such as Gaia[²], PACE[³], SOCAM[⁴], CoBrA[⁵], etc. The centralized model has an outstanding advantage of the lower cost for backup and communications of system data. However, the central server is subject to malicious attacks to reduce the robustness of the whole system.

(2) Peer-to-Peer model

In a peer-to-peer model, contexts are distributed among users’ devices. The Context Toolkit[⁶] is one of the most typical distributed computing architecture with context awareness. The Context Toolkit constructs a platform for developing applications with context
awareness to support sentient computing. It should be emphasized that the Context Toolkit uses widgets as the primary abstraction for context, which means that there are the same number of context widgets registered in a center as that of contexts. For this reason, there is a problem of high cost for communications with registration. With difference from Context Toolkit, Context Fabric\cite{7,8} takes the architectural model in the Context Toolkit and generalizes it to an event service and a query service. Besides, Context Fabric separates specifying context needs from the processing of those context needs. However, Context Fabric has the similar problem of extra cost for management because of the context distribution.

There is the other work in the infrastructure of context management, such as MUSE\cite{9}, Stick-e Notes\cite{10}, CoolTown\cite{11}, and so on. These solutions have respective features compared with aforementioned systems. For instance, MUSE takes a sensor-centric approach and put more emphasis on uncertainty inherent in all sensor-based data. Stick-e Notes provides a high level authoring tool for context-aware applications. CoolTown is reserved for the use of web-based service applications. Although these systems get great improvements on the context management, the higher cost of management still exists.

1.2 Our work

In this paper, we propose a ubiquitous computing-oriented peer-to-peer context sharing model (PCSM). Based on gathering, inference and storage of contexts in terminals, the PCSM model is focused on intercommunication, sharing and fusing of contexts among terminals. Besides, the PCSM model offers a context presentation platform with the design of context presentation engines to construct a context awareness frame-work for contexts sharing and fusing.

The PCSM model is a kind of distributed model. In contrast to the centric computing model, the distributed computing model could improve the robustness of system and prevent the system from collapse caused by breakdown of the centric node. Furthermore, a distributed system could get the load equalization of the whole system. However, the distribution of a system will introduce the higher cost of synchronization and communication as well as the higher redundancy of storage and communication. To realize the tradeoff between the distribution and the efficiency of the peer-to-peer ubiquitous environment, the PCSM model improves the performance of the system by optimizing the design of communication messages and system storage. Hence the cost for communication and storage redundancy of PCSM model is lower than any other distributed computing models.

The remainder of this paper is organized as follows. Section 2 describes the architecture of PCSM model, including the general introduction of some components of the PCSM model. Section 3 details the mechanism of context sharing and contexts presentation engine. Section 4 reports a prototype of a ubiquitous Instant Messenger based on the PCSM model. Section 5 draws a conclusion and gives a prospect of the future work on the PCSM model.

2. PCSM Architecture

The main aim of the PCSM model is at providing a general foundation of the framework of contexts sharing and fusing. As shown in figure 1, the architecture of the PCSM model comprises three components, including Registration Center (RC), Context Database Agent (CDA) and Context Presentation Engine (CPE). These three components are mapped respectively into context manager, context provider, and context query.

The Registration center is a component used for registration of CDAs. The Context Presentation Engine gets the interface of registered CDAs through sending a request of query to RC, and builds up a channel for communication with each requested CDA. To construct an intercommunication mechanism among terminal contexts, the RC plays a role of communication agent for synchronization of contexts among terminals as well as managing the interaction between the CDA and the CPEs in the same network terminal. By the intercommunications among RCs of terminals and sharing of the interfaces of CDAs in a terminal, the PCSM model archives the goal of implementing remote registration mechanism of CDAs, which makes the CPE component in the same terminal able to have access to CDAs in the other terminals. These three main components are described in detail as follows.

(1) Registration Center

The Registration Center is a critical component of management of context semantic. The RC provides the interface of registration to support the registration of CDA, the interface of query to support the query into context database from the CPEs, and the response of interface of the relative CDA component.

(2) Context Database Agent

The Context Database Agent is unique in a terminal and used to maintain the access interface of terminal context database. The terminal context database contains terminal contexts which could be modeled with RDF (Resource Description Framework) and OWL (Ontology Web Language) and relationship between these contexts. These two specifications have got widely used in the field of Semantic Web. As an access
agent of the context data-base, the CDA gets the query request from the CPEs, and looks up the context database. Then the CDA uses the outcome of query as the response to the query request.

3) Context Presentation Engine

The Context Presentation Engine involves a new method of context query which meets the requirements of query specifications. The CPEs refer to some query languages of RDF, such as RDQL, RQL or SPARQL, to construct the map between query semantic and context space. Through the RC, a CPE gets the access interface of the correlative CDA, interacts with the CDA and implements query about context semantic. Meanwhile, a CPE provides presentation interface for both application services and communication awareness to support some specific functions in a heterogeneous network, such as grouping and routing decision in a self-organizing network.

Figure 1. The architecture of context sharing and fusing

3. Context Sharing Mechanism

The mechanism of distributed contexts sharing is aimed at constructing intercommunications among terminals. An Emphasis is placed on the maintenance of the resident contexts and the peripheral contexts sharing.

3.1 Context model

Modeling on contexts has got intensively researched in recent years. A survey paper, in which some context models such as key-value model, makeup model, graph model, object-oriented model, logic-based model and ontology-based model are involved, draws a conclusion on these models[12]. These models have respective features and scope for use. As the context model of a peer-to-peer ubiquitous computing environment, the contexts should be classified as follows.

1) Resident context

Resident contexts are resident in the terminal where applications are executed. Compared with the other two classes of contexts, the resident context has a closer relation with the applications. Consequently, the resident contexts have more effects on the decision on these applications.

2) Neighbor context

Neighbor terminal node is an important factor for grouping algorithm and dynamic routing selection. In a peer-to-peer network with context awareness, the neighbor relation between two terminal nodes depends not only on the distance between two nodes but on some peer contexts of the two nodes. In addition, some sentient computing applications actually need to get both resident contexts and neighbor contexts to make changes in their behaviors.

3) Out-of-group context

The self-organization is usually the topology of a peer-to-peer network, of which a group is a fundamental unit. Since the intercommunication between two terminals which are located in different group frequently take place, application services in a distributed self-organizing environment are needed to have ability of implementing query into contexts between groups.

The context in the aforementioned context model is defined as a triplet TContext.

TContext=(group_id, node_addr, context_metadata),

where Group_id is the identifier of a group header.

Node_addr indicates the address of a terminal node.

Con-text_metadata represents the metadata structure of context. The value of these three types of contexts in context space can be defined as a set of if-then rules.

if (group_id=null & node_addr=null)
    TContext=Resident_Context;
else if (group_id=null & node_addr_ip_addr)
    TContext=Neighbor_Context;
else (group_id=ID & node_addr=ip_addr)
    TContext=Out-of-Group_Context ;


3.2 Context resident environment

Context resident environment which is the platform of interaction between CDAs and CPEs in the same terminal forms the foundation of distributed peer-to-peer terminal context management. In the environment, the RC, CDA and CPE components play different roles. Based on the mechanism of registration-query, the environment implements the registration of CDAs and the access interface of resident contexts.

The whole procedure of management of contexts in the context resident environment involves two steps. The first step is the registration of a CDA. The CDA in a terminal is registered in the RC of the same terminal with some registration messages. As shown in Figure 2, a piece of registration message is composed of the general descriptions of some CDAs and the request for contexts. After receiving the registration messages from the CDA, the RC constructs a list for all the registered CDAs which will be looked up by the CPE afterwards. The general descriptions give the identifier which represents the index of CDA in the list and the set of types which contains all definitions of types about contexts saved in the context database related to the CDA. Since the CPE sends the request message in an active way, the CDA should offer the interfaces of context data transport in the registration message.

The second step is context query. The CPE sends the context query messages to the RC for query into the relative CDAs. The CPE gets interface for interaction with these CDAs, and will send requests for contexts to these CDAs. Through the query step, a CPE will get some contexts which it get interested in and supply them for the sentient computing applications.

It should be emphasized that the CDAs registered in the RC of a terminal includes not only the CDA in the terminal but the CDAs in the other terminals. Through intercommunicating with these external CDAs, the CPE can get some wanted contexts and provide these contexts to those sentient computing applications. The construction of channels between the CPE and those external CDAs is an important factor for the implementation of the context sharing.

3.3 Context sharing messages

The context sharing is aimed at the context intercommunication among distributed network terminals. As shown in figure 1, the RCs take a role of an agent of context sharing among terminals. The context sharing message is critical to the RC. The context sharing message comprises two types of messages. One is broadcast message for detecting the peer RC, the other is unicast message for the registration of remote CDAs. The broadcast message plays an important role of detecting the available RCs. A Registration Center detects some available terminals, gets the communication interface of the RCs in the terminals and constructs the intercommunication channels with these RCs. During the registration procedure, the CDA firstly sends a registration message to the local RC, and gets the available channels for registration in the remote RCs. Then the CDA sends messages for remote registration, and implements the registration in a remote RC. The figure 3 shows how the CDA of terminal1 is registered remotely in the RC of terminal2. The whole procedure of registration is composed of seven steps. In each step, there are some messages sending and receiving between the CDA and the RC. Similarly, the CDA of terminal2 adopts the same way of registration as that of the CDA of terminal1.
3.4 Feature Analysis

There is not a central node in the peer-to-peer computing system. If a mobile terminal is designed as a center for communication, context loss will take place frequently when the terminal is shut down or unavailable. Moreover, the fault tolerance or backup of storage will have a negative effect on the performance of system. Therefore, the PCSM model is designed as a distributed architecture. Compared with the other distributed solutions, there are two features of the PCSM model.

(1) Lower cost for communication
As aforementioned, there is usually higher cost in the distributed computing model. The PCSM model focuses on the problem. It is not the context metadata but the CDA that is remotely registered in a RC. The message for remote registration is a lightweight message because there is usually unique CDA in a terminal. The lightweight message makes the cost for communication of the PCSM model lower than that of the other generic distributed model.

(2) Lower redundancy of storage
Thanks to the remote registration of CDAs in the PCSM model, those contexts related to a terminal are resident in the terminal. What is more, it is synchronization of CDAs that take place among terminals. Therefore, contexts in the context database are not in need of synchronization. The PCSM model will get a low redundancy of storage.

4. Prototype

Based on the PCSM model, we have developed a Ubiquitous Instant Messenger (UIM) to support office collaboration and automation. The UIM has an ability of context awareness. The UIM client which is installed on a wireless terminal enables users to chat with the others by providing means of short texts. These information intercommunications always take place between two peer users. As compared with those existing IMs, the UIM provides users the ability of observing the current whole map of the state of the collaboration space make the reasonable decisions for the next step in addition to information sharing and collaboration.

4.1 Application study

As the sentient computing application, the UIM provides several ways of applications in an actual scenario.

(1) Observations of the profile of all participants

As an existing solution for observation problem, the IMs provide a manual way of showing whether people are available or busy and which activities people are engaged in, which often leads to misrepresented statuses. The users, terminal devices and the IM construct a distributed peer-to-peer ubiquitous computing environment. As for the UIM, the state of a user depends on the place where the user is staying, the time when the session is active and the other peripheral contexts around the user. At the same time, these contexts are collected and maintained by the respective terminal device the user hold.

(2) Different levels of intercommunications
The UIM provides a switch from written text to verbal communication to form diversified channels for information intercommunications between two peer users. The decisions for the choice between written and verbal communication rely on the current situation of the participants. In many cases, the UIM allows for communication patterns that are biased towards one or the other according to convenience of users.

(3) Reasonable plan for collaborations
Collaboration is the kernel feature of an IM system. It should be emphasized that creating, maintaining and deleting collaboration relationships among users to support such collaborations as whiteboard coordination, group conference and schedule, and so forth. In contrast to some common IMs, the UIM collects the contexts but the state of peer users to get the whole optimal decision for a plan through pulling some related contexts from peer terminals.

4.2 A case of the UIM

The office automation of a development group is a case of the UIM. As far as a development group is concerned, the group is composed of three subgroups. The group as well as subgroups is assigned a leader responsible for the development activities of respective group. It is an presumptive regulation of the group that the higher-level member could get some public contexts of the lower-level ones and make them as the founda-tion of management of the whole group.

It is assumed that the leader of group (LG) wants to hold a meeting to make a plan for development. He should confirm when the three leaders of subgroups (LSG) are available to present in the meeting. Therefore the LG may get the distributed schedules recorded in the respective terminals of LSGs. Once the conflicts between schedules of LSGs and the designated date by LG take place, the LG should decide whether change the date or inform those LSGs according to the urgency of the meeting and the schedule of LSGs. The conference room of the UIM is shown as figure 4. None of the LSGs is available and
two LSGs are available are represented respectively by (a) and (b).

As the PCSM model shows, the LG looks up the schedule contexts in the context database of each LSG which have been registered through the Register Centers in each terminal. By making a comparison between the designated date and schedule, the LG is able to find out whether the aforementioned conflicts take place or not. If there are no conflicts, the LG will inform the three LSGs of the date to hold the meeting. However, if there are come conflicts, the LG should weigh the different urgency level between the meeting and the schedule, and send messages of schedule changing or date informing. The choice of channels through which the messages are sent should comply with the current situations of the receivers. It is supposed that a LSG is making a presentation in another meeting when the LG wants to sending a message to him. The LG are required to observe the current situations of the LSG through the approaches shown in figure 1 before sending a message to the LSG.

![Figure 4. The conference room of the UIM](image)

5. Conclusions

The PCSM model, a ubiquitous computing-oriented peer-to-peer context sharing model, is put forward in this paper. To fill the requirements of sentient computing applications on distributed contexts in a peer-to-peer ubiquitous network, the PCSM model constructs the channels for remote registration of Context Database Agents through the mechanism of Registration-query. We have developed a prototype of UIM to verify the validation of the PCSM model.

The Context Presentation Engine is another critical component of the PCSM model. The CPE looks up the context database for interested contexts. Besides, the CPE presents these contexts to sentient computing applications. In general, the role of the CPE lies in providing the interface between sentient computing applications and contexts. Consequently, researching on specifications for context query and mapping from query logic to context space are involved in the future work of the PCSM model.

Acknowledgement

This effort is supported by Foundations of R&D Infrastructure and Facility Development (No. 2005DKA63905) and Project of National High-tech R&D Program (2006AA01Z112) of Ministry of Science and Technology of the People’s Republic of China.

References