Intelligent Business Assistant Based on Context-aware Computing Platform

Yanshuai Zhang[1], Hong He*
[1, *]*School of Mechanical, Electrical & Information Engineering*
Shandong University at Weihai
Weihai, China
zhangyanshuai@gmail.com
hehong@sdu.edu.cn(:corresponding author)*

Abstract

The existing reminder system uses limited context information and has limited functions, so it can't meet users' needs. This paper presents intelligent business assistant based on context-aware computing platform in a grid computing environment. The platform uses the context modeling and reasoning technology and analyzes the system in the grid environment. At last, the intelligent business assistant is implemented in a SaaS environment.

Keywords: Context-aware Computing Platform; Intelligent Business Assistant; Grid Computing; SaaS.

1. Introduction

Context-aware computing obtain and store the computing-related context information about sometime, somewhere, someone and some scenarios through the sensors, computing interactive equipments, user settings and other channels. It makes analysis and speculation to users' behaviors according to these information and context of the historical information, and provide personalized service and information to users intelligently and actively [1].

Today's society is full of a variety of business activities. Travel is common enough for business people. There are so many troublesome problems that you can't order a room or get to the meeting venue in time in an unfamiliar city. Even in a familiar city, sometimes you can't find a free parking or book a passenger ticket, sometimes you will forget the meeting, the appointment and so on. So an intelligent business assistant is necessary for business people, which can remind them in time.

The current productions which can remind people mainly include CybreMinder [2], Gate Reminder [3] of the Samsung and DeDe [4] of Nokia. The process of generating information by these products is complex. And they are inconvenient for user to use. There are also some intelligent mobiles which can remind people things they saved. The GPS navigation system can show you the way and remind you the speed limitation. But the single function and the lacking context interaction of this kind of product can't meet the users' needs.

Literature [5] proposed an intelligent reminder based on context-aware computing platform. On the basis of it, this paper proposes an intelligent business assistant based on context-aware computing platform in a grid computing environment.

2. Architecture of Intelligent Business Assistant

2.1. Context-aware Computing Platform

Literature [1] analyzed the context-aware computing platform and got a conclusion that this platform can be divided into three layers, which is shown in figure 1.

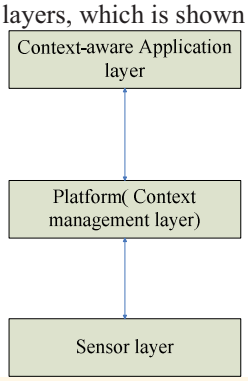

Fig.1 Context-aware Computing Platform

Interaction

Sensors are used for obtaining the context. From the perspective of pervasive computing, the context refers to any information can be used to characterize the state of entities. The entities include individual, location, objects in physical or information space. The context includes as follows [6]:

2.1.1. Computing context. E.g. the network connectivity, communication cost, communication bandwidth and nearby resources (such as printers, display and workstations, etc.);

2.1.2. User context. E.g. the user's characteristics, location, time, the nearby staff, the current relationship;

[1] Project supported by the National Natural Science Foundation of China (Grant No. 60903052). Yanshuai Zhang, Master degree candidate, research field: web service consistency. Hong He, Doctor, associate professor supervisor of the Master students, research field: algorithms design and analysis, grid computing.
978-1-4244-9142-1/10/$26.00©2010 IEEE

2.1.3. Physical context. E.g. light, noise levels, traffic conditions and temperature;

2.1.4. The history of context. Platform layer is used for managing the obtaining context, including the context storage, context reasoning, context prediction and context integration. The context is processed into needed data.

Application layer provides specific application, and meets the users' needs according to the obtained context.

In this paper, we think the context is obtained not only by all kinds of sensors, but also other ways, such as Internet. The context, which can't be obtained by sensors, can be obtained from Internet. So the Context-aware Computing Platform should be the one shown in Figure 2.

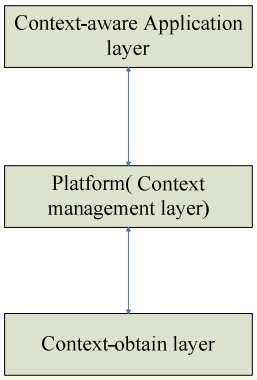

Fig.2 The Improved Context-aware Computing Platform Interaction

2.2. Architecture of Context-aware Computing Platform

This paper proposes the architecture of this platform according the three-layer computing platform. A design pattern is used here, which is open for extension but closed for modification. A variety of applications can be extended on the basis of this architecture. The architecture is shown in Figure 3.

2.2.1. Sensors and Internet. Sensors can be a variety of environmental sensors, equipment sensors and virtual sensors. In a sense, sensors include all electronic components can obtain information from other components. Sensors mainly use RFID (Radio Frequency Identification), GPS, infrared, supersonic wave, etc.

RFID is a non-contact automatic identification technology, which can work in a variety of severe environments. RFID is a very important technology in the development of current sensor network. It is considered to be the most effective way to identify objects by replacing the bar code widely used today. The RFID system usually consists of RFID tags, reader, and the antenna. Get the context from the sensor is a direct way, but through the Internet or other means to obtain the context is an indirect way. The analysis of the integrated data obtained from the Internet and the data from sensors can get better results, because the sensors alone may not be able to obtain the whole context. We need other ways to get more contexts.

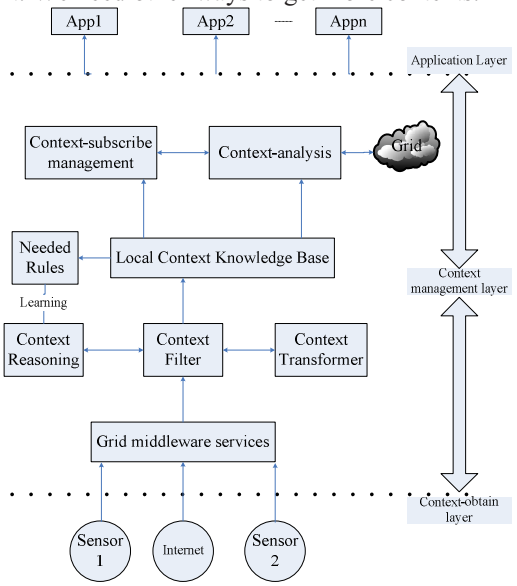

Fig.3 Architecture

2.2.2. Grid middleware services. This layer is a layer of middleware, which is used for transforming the heterogeneous, different formed context obtained by a variety of sensors into the ones which is isomorphic, same formed and handled by context filter easily.

Literature [5] uses the context agent to complete this task. One of the solutions is multi-agent-based framework. There is context manager agents, coordinator agents, ontology agents and context provider agents, etc, in this framework.

Grid middleware services are considered to be more suitable in this paper. The context obtained by the sensors sometimes needs many agents and they are usually heterogeneous and inexplicable. It is particularly important that Grid provides a powerful computing ability and the effective process of dynamic services in this case.

2.2.3. Context Filter. The context information obtained may be incorrect or conflicting, because the context information is not perfect. There are two main aspects about the inconsistent context. Firstly, the inconsistent context may be obtained by different sensors; Secondly, Context reasoning may cause inconformity. The information provided to upper layer must be correct, otherwise the reasoning and application of upper layer will be meaningless.

Literature [7] presents an inconsistency verification method, using OWL language to describe the observed context as an ontology and giving a ontology model and instance data to verify the consistency between context described by ontology model and context defined by ontology instance.

At first, context is transferred from filter to context transformer or context reasoning, and then return to filter after processing, finally, context will be added to context knowledge base after the inconsistency validation.

2.2.4. Context Transformer [5]. Completing the transformation of information is the main task of this layer, but the meaning and abstract level of context will not be changed, such as unit conversion, coordinate transformation of vector, etc.

2.2.5. Context Reasoning [8]. Context reasoning has two main tasks, one is speculating users' current activity, and another is speculating users' next activity according to the current one. For example, when a user brakes his car in a travel, the system needs to speculate the user's current activity is running into a traffic light or taking the initiative to stop, then determine that he is lost or try to find a way to parking.

Context reasoning deals with the context as a kind of knowledge by drawing on machine learning.

There are two main reasoning technologies [9].

Rule-Based Reasoning

One of the main representatives is GAIA. Defining reasoning rules is generally adopted by ontology-based context reasoning. Usually RDF language is used for describing the model of metadata, XML and OWL language are used for modeling to describe logic and FOL is used for writing some rules to do the reasoning.

Machine learning-based Reasoning

There is Bayesian networks, neural networks and learning based on Markov model in this reasoning. Bayesian network is widely used in recent years. Neural network is widely used in context aware systems, including context recognition and prediction. Speculating users' current activity can use rule-based reasoning. Users can develop custom reasoning rules. Speculating user's next activity can use Bayesian neural network. Both of the reasoning is needed to know the dependencies between contexts. They can use the ontology reasoning engine technology.

2.2.6. Local Context Knowledge Base and Grid. Local context knowledge base is used for storing the context knowledge through reasoning and rules defined by users and systems. A lot of knowledge can be stored in the grid server in the current grid computing environment because of the large amount of data. Local Knowledge only save the most commonly used knowledge and rules. After establishing the mapping to the server, knowledge base can communicate with the server when it needs data. This measure can reduce storage requirements of knowledge base.

2.2.7. Learning mechanism. The platform draws on the learning mechanism in machine learning. Not all the knowledge in the knowledge base meets the users' needs, or there is sometimes inconsistency between the users' needs and knowledge. Adding the rules, which meet the users' and system's need, back into the context reasoning module will improve the accuracy of reasoning. The accuracy will be higher and higher by studying. Different users can modify reasoning rules according to their custom, so the system can be customized.

2.2.8. Context Analysis. As knowledge stored in the knowledge base is encoded in different languages, such as OWL, DL language, so it needs to be resolved to a format which can be identified by context subscription management.

2.2.9. Context Subscribe Management [5]. Context subscription management is the interface to handle the subscription of the applications. This interface allows the edit (add, delete, change) subscription of the applications. When the subscription is resolved and the syntax and semantics are appropriate, the subscription manager needs to check that the interaction is response-request mechanism or event-driven mechanism. The subscription manager will immediately implement subscriptions when the response-request mechanism works; otherwise the subscription manager needs to constantly check whether the event subscription is true or not in order to notify the appropriate information to the application.

2.3. Create instantaneous service for pervasive computing

Context-aware platform will obtain a lot of context via sensors. A large number of data sources access to the grid application through Internet. Pervasive computing requires us to deliver grid computing power to the place where computing needs, so that computing is everywhere at all times[12], as is shown in Fig.4. To achieve this goal, grid services should be readily adaptable to the users' mobile in different environments, so the mobile device carried by users can interact seamlessly with the services provided by grid platform. Take GPS for example, a person wants to know his location or nearby parking via mobile phone rather than his personal computer. It requires that grid services of GPS can monitor these needs, and communicate with the remote digital devices. During the query the user may move from one place to another, so the new connections generate continuously to replace the old ones, which calls for new instantaneous services replacing the old ones to response users' needs.

Although OGSA and the perfect pervasive computing platform also have some distances [13], it is the most feasible architecture which we can get to support pervasive computing until now. Because the OGSA's goal is complete consistent with the network demand of pervasive computing. In the basic web services model, if the creation and revocation of services uses the mechanism beyond the standards of

web services, using web services to meet the needs of pervasive computing will be less flexible and convenient. Because the dynamic and random characters of pervasive computing often need to initialize new instantaneous services to meet users' demand, such as processing management and interaction related to a particular activity. When the state of the activities is no longer necessary, corresponding services can be removed to save resources.

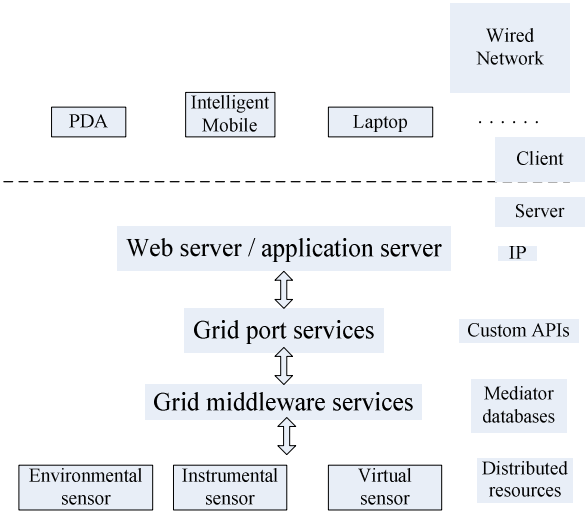

Fig.4 A variety of context accesses Internet via grid

The instantaneity of services has great influence on how to create, name, use, and find a service. A direct result is separating the definition of grid services from its instance. Grid service description consists of defining of the grid service interface and its corresponding semantic WSDL (and OGSI) expansion. Grid service instance is an addressable instantiation service of corresponding description. It can have a state and be instantaneous. Examples of transient service are such as database queries, data mining operations, network bandwidth allocation, ongoing data transmission and the reserve of processing capabilities, and so on. It indicates that services can be used to manage it of a large number of short active super lightweight entities.

3. Implement of Intelligent Business Assistant

The implement of intelligent business assistant is shown in Fig.5. There are three layers. Users log in the assistant via client. The assistant implement the management to context. Context-obtain layer transfers context to SaaS.

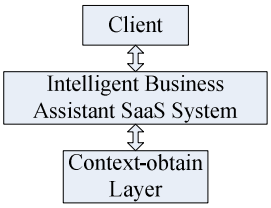

Fig.5 Implement of Intelligent Business Assistant

3.1. Client

Client mainly implements the interaction between users and the SaaS system. It need to be convenient to access Internet, has good maneuverability and UI, some intelligence and so on. Intelligent phones meet these demands well, so it is the best choice for client. Current intelligent phones mainly include Symbian S60 of Nokia, Windows phone of Microsoft, iphone of apple and Android of Google.

Smart phones have a small size, low power consumption, and is easy to carry and highly popular. All of these advantages make it a best choice for the client.

3.2. Intelligent Business Assistant SaaS System

Intelligent business assistant has a irreplaceable advantage in processing context because of the powerful computing, resource sharing and collaborative working of Grid. But grid computing is not really built till today. Therefore, SaaS is adopted to replace grid computing in this paper. SaaS can be said to a specific product merged by development of grid computing and the current social reality. Its core concept is the service-oriented software. SaaS treats software as a service, not a commodity.

Users pay to use the services provided by SaaS platform on demand. They only need to use the genuine software from SaaS instead of spending a lot of money to buy software themselves. Thanks to it, some small and medium enterprises reduce their costs to accelerate their development, and genuine software is supported effectively.

The topology of SaaS system infrastructure is shown in Fig.6.

The Implementation of infrastructure adopts virtualization, four-way and four-core server and FC-SAN. Virtualization technology is used for building a HA cluster with several servers. In the virtual cluster, the whole cluster can be divided into different parts, or a server is divided into different regions. According to the type of resource, cluster or server is divided into the same number of virtual servers. The appropriate operating system and application software are installed in each virtual machine, which work for different resources and services. Virtual server is transparent to users, who seem to have an individual server. Virtual

machine needs to achieve the logical isolation between each other internally.

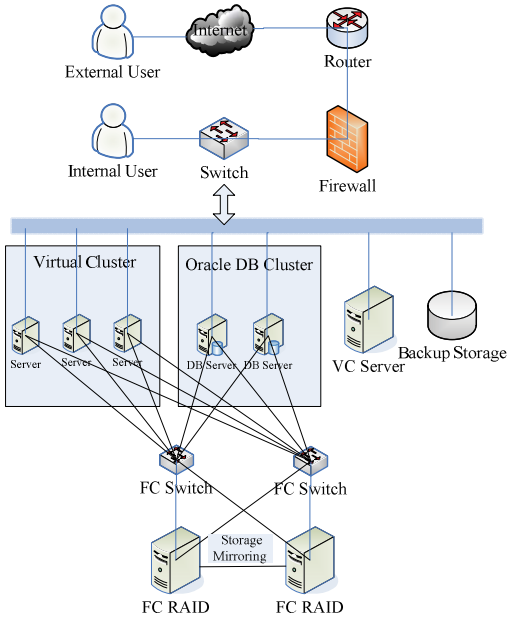

Fig.6 The topology of SaaS system infrastructure

Oracle database software can be deployed on four-way and four-core servers to build the database cluster. Oracle's powerful features can improve the reliability and continuity of system, and ensure a good performance in dealing with large-scale users. Data isolation must be achieved in database cluster. Because users' data stored in the SaaS platform often involves commercial secrets. A user's data must be invisible to other users. Taking the scalability and cost into consideration, increasing the number of servers to ensure that every user has a server is not a reasonable option. The reasonable solution is the data isolation. Different users can use the same server, but one's data isn't accessed by others.

Storage is implemented in FC RAID using FC-SAN technology. FC-SAN is more stable, mature, open and stable than 10GbIP-SAN. Mirrors are created in two FC RAIDs. When a working RAID crashes, the spare RAID will replace it. This method ensures the continuity and service of the system. Users can use SaaS platform normally even an accident happens. Meanwhile, crashed server will be recovered from disaster to be a spare server.

After the virtualization, the topology of SaaS system infrastructure is shown in Fig.7

Intelligent business assistant SaaS system module is shown in Fig.8.

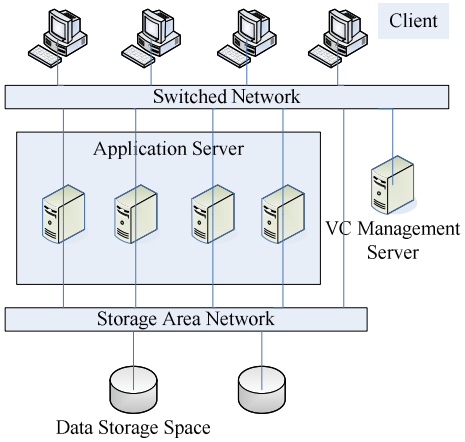

Fig.7 The topology of SaaS system infrastructure

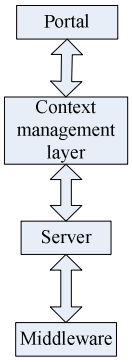

Fig.8 System module

3.2.1. Portal. Users of intelligent business assistant SaaS system browse the portal to know the list of services provided by the system. They get the permission to customize the corresponding service by paying for registering. As long as users access to the Internet, they can log on the portal to customize services.

Portal is a B/S system which has multi-layer, distributed data and distributed applications.

To meet the users' growing demand and keep good scalability, website will reserve corresponding interfaces for possible additional services. If we want to add some services, we only need to extend rather than modify the design of system. Portal uses role-based access control (RBAC) to manage users and user authorization.

The access to web is limited, deciding by user's own authority. Greater authority means more resources. The system achieves secure hierarchic access to the system and avoids the management to a large numbers of users one by one by dividing users into different groups. Roles and users is many-to-many relationship. User's role will be updated when his authority changes. Web site also provides plug-in for users to customize the interface and the services they need.

Role-based access control is in accordance with the user's authority to manage system. It is easy to control,

reduce the burden and costs of rights management, and enhance the security of the system.

3.2.2. Context management layer. Context management layer is designed as it shown in Fig.3. The difficulty of the whole system is completing the function of context analysis. This layer also identifies the data delivered from another layer and abstract it into the appropriate service, for example, the data reflecting roads and buildings can be abstracted out of GPS services and the service of finding a hotel or a parking. How to abstract as much service as possible from a lot of data is urgent to be solved. System also packages a variety of services based on users' roles, and provide service packages for different users for their customization.

3.2.3. Server. Servers store processed context knowledge. Different types of knowledge can be separately stored to be convenient for management and use. Server also stores users' data, and makes the data isolation. The design proposal generally divides into the complete isolation type, complete sharing and the database share Schema isolation [11].

The design of the complete isolation type combined with the database share Schema isolation is adopted to store data in this paper. User who has higher authority and more customized services will use the complete isolation type, and general users will use the database share Schema isolation type.

4. Shortage of Intelligent Business Assistant

4.1. Data security and personal privacy

Intelligent business assistant will communicate with a lot of sensors, servers or even satellites. How to ensure that data is not intercepted, altered and forged, the system is not compromised, and users will not get the wrong message, are the very important issues. We should also consider how to avoid personal privacy or even national secrets aren't violated.

4.2. Man-machine interaction

We need to develop a more natural way than the existing way of human-computer interaction to reduce users' operational difficulties, so users will be easier to interact with the system and focus on solving problems. We should also show the transparency of pervasive computing in this field.

4.3. Context Obtain

Although sensor technology developed rapidly, how to obtain context accurately and completely needs to be solved completely. In consideration of the distribution, heterogeneity and polymorphism of sensors, there is a great challenge than how to obtain sensor data and transfer them into useful context in a unified way.

5. Conclusion

In summary, the paper designs a distributed intelligent business assistant by obtaining the context in the pervasive computing and grid computing environment. The learning mechanism is added to the process of context in the system. It enhances the robustness and communication with the server by storing the context in the SaaS server. However, the system is in the preliminary study stage, there are some problems for further studying.

References

[1]Wang Wei, Xie Zhenglong, Xu Qing. Intelligent Reminder Based on Context - aware Computing Platform. Journal of Changchun Normal University (Natural Science). 2009, 28(5).

[2]Dey A K, Abowd GD. CybreMinder: a context - aware system for supporting reminders[C]. In Proceeding of HUC, 2000, Bristol, UK: 172- 186.

[3]Kim S W, Kim M C, Park S H et al. Gate reminder: a design case of a smart reminder[C]. In Proceeding of the 2004 conference on Designing Interactive Systems :processes ,practices ,methods ,and techniques ,Cambridge ,Massachusetts ,USA. 2004: 81 - 90.

[4]Jung Y, Persson P, Blom J. DeDe: design and evaluation of a context-enhanced mobile messaging system[C].In Proceeding of CHI 2005, Portland, Oregon, USA.2005: 351-360.

[5]Wang Wei, Lin Peng. A Survey of Context-aware Computing Platform. Journal of Xianyang Normal University, 2009, 24(4).

[6]Xu Guangyou, Shi Yuanchun, Xie Weikai. Pervasive/ Ubiquitous Computing. Chinese Journal of computers, 2003, 26(9).

[7]Wang Wei, Qian Yi, Tang Shancheng. Research on context reasoning in pervasive computing environments. Computer engineering and design, 2009, 30(14).

[8]Lu Ping, Zhong Liang. Research on healthy smart home based on context-aware computing. Computer engineering and design, 2009, 30(21).

[9]Wu Zhaohui, Pan Gang.Pervasive Computing. China Computer science and technology development report 2005, China Computer Federation Proceedings CCFP-0002, pp.175-187, Tsinghua University Press, Beijing, 2006, 8

[10]He Hong, Gong Peng, Lin Zongkai. A Method of Creating Grid Service and Its Application in Pervasive Computing .Journal of Software，2006, 17(Suppl.):192-199.

[11]Frederick Chong, Gianpaolo Carraro. Multi-tenant data architecture [OL]