A Framework of Transferring Mobile Services with Agent Based Middleware

Weidong Zhao
School of Software, Fudan University, Shanghai 200433, P.R.China
E-mail: wdzhao@fudan.edu.cn

Haifeng Wu,Weihui Dai,Wenjuan Wang and Guangxue Yue
School of Software, Fudan University, Shanghai 200433, P.R.China
School of Management, Fudan University, Shanghai 200433, P.R.China
School of Software, Yunnan University, Kunming 650091, P.R.China
College of Mathematics & Information Engineering, Jiaxing University, Jiaxing, Zhejiang, P.R.China
E-mail: 082053025@fudan.edu.cn, whdai@fudan.edu.cn, wenjuan42@gmail.com, guangxueyue@yahoo.com.cn

***Abstract*—With the development of mobile terminal devices and network communication technology, service applications in mobile environments with large potential users become more and more popular. After analyzing the particularities and obstacles of the service transferring in mobile networks, this paper introduces a mobile service transferring framework based on the agent middleware, aiming to improve the efficiency and performance of service delivery in mobile environments. As the communication bridge between mobile application service servers and mobile terminal applications, it not only can achieve efficient and timely interaction between them and reduce the service transferring workload of mobile terminal devices, but shield off heterogeneities of terminal platforms and network communication protocols. The service transferring framework offers some advantages, such as platform-independent, favorable extensibility, strong flexibility and well-behaved security. Mobile terminal users can utilize it to transfer services and resources in mobile environments efficiently, without worrying about the mobile network discontinuity and limited computing ability of mobile terminal devices. Moreover, it can provide user-centric services through building user profiles in order to improve user satisfaction and attract more users.**

***Index Terms*—mobile service, service transferring, agent middleware, data synchronization, heterogeneity**

I. INTRODUCTION

Recently, the immense popularity of mobile terminals and communication technologies has quickened the development of service applications in mobile environments, which has received much attention from academics and practitioners [1]. As a special kind of service, mobile service can enable users to make purchases, request services, access news and other information with their handheld devices. With advanced mobile communication technology, mobile services can help people enjoy user-centric and timely application services in the mobile network. However, there are still some restrictions limiting further development of the mobile application. For instance, the network bandwidth is more limited and the network status is changeable, which exert a great influence on data transmission and data synchronization mode. Besides, the processing ability of terminal devices in mobile environments is lower than that in the fixed network, which makes terminal devices very difficult to take charge of service searching and parsing. Traditional service transferring strategies, such as direct service transferring of mobile terminals, are not very effective any longer. Service applications in mobile environments are facing some great challenges because of these limitations and particularities. Flexible and effective solutions are required urgently. At the same time, the importance and efficiency of middleware and multi-agent technology has received much attention in the industry. How to improve the efficiency and performance of service delivery in mobile environment has become a hot research topic. With autonomy, reactivity, initiative and mobility, agents can be very useful for implementing application systems in the mobile network with favorable capability and expansibility [2]. While the middleware, lying between client applications and host servers, is for managing computing resources and harmonizing network communication, which is computer software that connects software components and different applications, consisting of a group of services that allow multiple processes run on terminal devices to make interaction across the network. Certainly, the agent-based middleware has been proved to be one of the feasible ways for service delivery in mobile environments, which can be the opportunity to bridge the gap between mobile terminal applications and mobile service servers. It not only can achieve the interconnection among different terminal application systems, but also it can shield off heterogeneities of terminal platforms and network communication protocols, ensuring users to call the services transparently without necessarily knowing

This research was supported by National High-tech R & D Program (863 Program) of China (No.2008AA04Z127), Natural Science Foundation of Zhejiang Province (No.Y1090264) and Shanghai Leading Academic Discipline Project (No.B210).
Corresponding author: Weihui Dai.

doi:10.4304/jsw.6.8.1578-1585

implementation details and concrete program codes. Nowadays, many researchers are paying attention to designing a high-performance middleware framework for mobile service transferring by integrating multi-agent and middleware technologies. Therefore, how to use the agent middleware to reduce the workload of mobile terminals and provide customer-centric services efficiently becomes a meaningful problem to be studied.

After analyzing the particularities of service transferring in mobile environments and the limitations of some traditional service transferring measures, this paper proposes a service transferring framework in mobile environments, which can ensure users to access the service and resources in the mobile network with their own mobile devices in time and conveniently, aiming to overcome the obstacles of mobile service transferring and then meet the personalized demands of mobile terminal users. In despite of mobile network environments are changeable and processing abilities of mobile terminal devices are restricted, the middleware based on agents can also work in mobile environments well with its flexibility and adaptability, due to the combination of mobile agents and middleware technology. And its asynchronous transferring mode can alleviate the problems posed by the occasional interruption of mobile network. Compared with WAP (wireless application protocol) gateway, there are some obvious advantages. For example, it can solve the compatibility of mobile terminal devices with different operating systems, and then realize the functions of the mobile terminal adapter to a certain extent. In addition, it can support more terminal applications and mobile network connection modes than ordinary gateway.

The rest of the paper is organized as follows: related study on mobile services and agent-based middleware are introduced in section 2. In section 3, we briefly describe the architecture of the agent-based middleware. The integrated service transferring model is shown in section 4. Detail design and technology implementation of the middleware are discussed in section 5, and then we analyze its characteristics and advantages. In order to validate the usability and efficiency of the middleware, we have conducted a case in section 6. Finally, we conclude this paper.

II. RELATED WORK

A. Mobile Service

Due to the advancement in the mobile network technology and fast improvement of mobile terminal devices, the development rate of mobile service applications become faster and faster. How to design and integrate mobile services efficiently has attracted broad attention. Mobile services, which are defined as self-governed and modular applications, can dynamically and quickly move among different network nodes, including fixed nodes and mobile nodes, and then be published, discovered and transferred with some specific technologies, providing a standard means of communication and interaction among different software applications involved in presenting dynamic context-driven information to the end user [3]. Compared with service utilization in the fixed network environment, there are many obstacles to service searching and transferring in mobile environments. First, because of the restricted processing capabilities and memory, mobile terminal devices are difficult to deal with service information parsing, including HTTP requests and SOAP (simple object access protocol) messages parsing. Second, some mobile terminal devices do not have favorable support for direct service accessing, which should be assisted with external tools to deliver mobile services. Besides, superfluous data delivery is not easy to achieve in the low-speed and limited-bandwidth of mobile networks. As stated in [4], “there are no full-blown enterprise services being delivered to people’s cell phones today”, which shows the constraints in mobile service applications. What’s more, uniform service description and discovery are also more complicated in mobile environments by reason of the differences in terminal operating systems, supported APIs (application programming interface), and script languages [5]. In order to ensure that mobile users are satisfied with service delivery performance and enjoy these service applications, the above obvious limitations should be overcome, so some flexible and usable service transferring methods are very necessary and important. In the design process, how to ensure mobile services be fit in with users’ mobile device specifications and meet their personalized requirements should be considered carefully. The method of designing and transferring mobile services are also significantly different from that in the fixed network and mobile services should be designed with many characteristics, being consistent with mobile network environment. Therefore, how to truly implement the mobility and flexibility of services has become a hot topic.

B. Mobile Service Transferring Mechanism

Generally speaking, there are three main manners which transfer services in mobile environments [5]. The first is to use JSR(Java specification requests) 172 specification, which is defined by Java Community Process for adding two important functions for J2ME platform, one is to enable mobile terminals with J2ME applications to access services based on SOAP/XML, and the other is to provide the ability of parsing XML messages for terminal devices. However, not all mobile phones currently on the market can support JSR 172 specification. At the same time, many programmers use KSOAP (a SOAP web service client library for constrained Java environments such as Applets or J2ME applications) or Wingfoot scheme to transfer application services in mobile environment, both of which offer many easy-to-use interfaces and software development kits to implement service transferring. But, regardless of JSR-172 or KSOAP, the complex object development for service analyzing must be carried out on mobile devices, which will bring some heavy burden to ordinary mobile terminals. Moreover, if the HTTP message processing and XML documents parsing are undertaken by mobile

terminals, the speed of HTTP processing and XML documents parsing is not high [6]. In order to overcome these obstacles and improve the efficiency of service transferring in mobile environments, the service accessing measure based on the middleware was proposed. As the bridge between mobile terminals and host services, the middleware can hide the complexity of service discovery and utilization for terminal users and provide personalized and timely services for them [7]. The workload in the mobile terminals will be reduced too.

C. Agent-based Middleware

Middleware connects software components and applications consisting of a group of services that allow multiple processes running on terminal devices to interact across network, which is regarded as a common and efficient architecture [8]. It can meet the needs of a large number of applications running on a variety of different operating system platforms, supporting different protocols and distributed computing modes. Considering the advantages of agent technology, such as autonomy, reactivity, initiative, many scholars have directed much attention to combine the advantages of middleware and agent technology to achieve better application effect and help customers acquire more satisfactory services. Therefore, agent-based middleware has aroused the universal concern in the industry, which is to achieve basic functions of traditional middleware with the mobile agent technology. Agilla is a representative mobile agent middleware in mobile networks, which allows users to deploy applications by injecting mobile agents into a network[7]. MAgNet, which is also a middleware based on agent technology, can enable social networking services for the user in mobile environments and let mobile users define and customize their social relationships with other users[8]. Those relationships would be used to plan and manage group events by corresponding agents. AMiGo-Mob, proposed to enhance group-oriented mobile service provisioning, which was modeled as three-layered architecture, consisting of physical layer, ontology layer and social layer [9]. Physical layer is responsible for observing mobile users as human physically and possessing mobile devices; ontology layer is responsible for building and analyzing semantic profiles of mobile users, while the subnets of social layer are based on the location and semantic similarity between users and service profiles. During mobile service transferring, the agent based middleware can work as a mediator between mobile terminal devices and host servers. Through some uniform application program interfaces, communications among heterogeneous systems are achieved, shielding off the complexity and difference of operating systems of mobile devices and network communication protocols, which has been proved to be a very useful method for service transferring in mobile environments. The agent-based middleware can parse SOAP/HTTP requests and responses instead of mobile terminals, and then provides satisfied services for terminal users by sending results in the appropriate format [10]. The reasoning ability of the middleware can be achieved with the self-learning ability of agents. In addition, through building a user profile library and storing the relate data and executing status of user agents, the agent-based middleware can provide user-centric services in mobile network environments.

III. AGENT-BASED MIDDLEWARE DESIGH

By combining the middleware with multi-agent technology, this paper designs an agent-based middleware for service transferring in mobile environments, aiming to improve the efficiency and performance of service transferring in mobile environments and reduce troubles brought by limited processing and poor storage capacity of mobile terminals. The core function of the middleware is to help mobile terminals indirectly to access application services in mobile networks with less resource consumption and offer user-centric services for terminal users by building user profiles. Compared with other types of service transferring measures, the agent-based middleware has many significant advantages. First, it can avoid network congestion and economize excessive energy, because of the mobility of agents and their light load. Besides, it can shield off the heterogeneity among mobile terminals and communication networks. Some measures of ensuring the integrality and security of data information and agents are also designed in the middleware. In this framework, the parsing of SOAP/XML message stream, services matching and complex object serialization are all achieved in the middleware. In this way, the obstacles of direct accessing to services are overcome and the performance of services is improved. The flexibility and extendibility of the system are also improved, because it can provide many efficient interfaces to different external applications. After analyzing the requirements, we divide the middleware into modules, including message parsing, agent reasoning, agent memory, agent management, external interface ontology library, etc. Fg.1 shows the location of the agent-based middleware in the service transferring framework. The main functions of every module are detailed as follows.

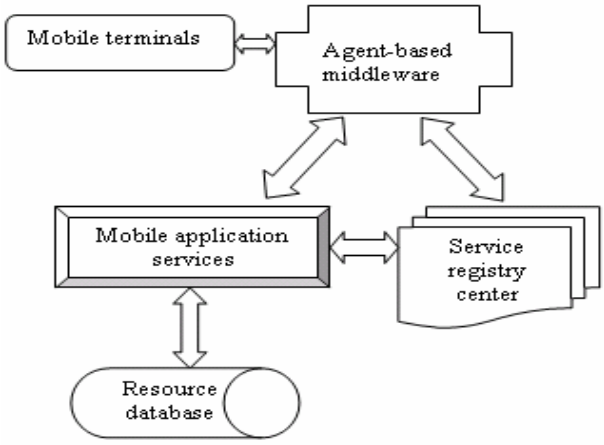


Figure 1. Mobile service transferring framework based on agent middleware

A. Message Parsing Module

In the mobile network, message transferring is achieved with XML files, which are parsed before being

displayed in terminal devices [11]. However, there are many obstacles for parsing these message files in mobile terminal devices because of their limited memory and poor processing ability. In order to address these problems, we design a message paring module in the middleware, which is responsible for paring the messages of mobile service applications as serialized byte streams instead of mobile terminals. In this way, mobile terminal devices can easily acquire responses and display results with less consumption, preventing burdensome work for mobile terminals for parsing XML messages and improve operating efficiency. After receiving the messages sent by the agent-based middleware, mobile terminals can transform the responses in the proper format for users with desterilizing function, and then display the information according to terminal traits including screen size, user preferences and so on.

B. Agent Reasoning Module

Reasoning is performed on the statements extracted from the previous step and results are inferred and then invoke the most relevant services with respect to the user profile and service descriptions in an appropriate format for terminal users [12], which can be used to derive higher-level context information from the lower-level. The reasoning module in the middleware mainly takes charge of encapsulating the reasoning into independent agents and making corresponding reasoning according to requests of users and some ubiquitous knowledge. In the module, many reasoning rules are collected artificially in advance. Besides, the module can also recommend appropriate services for users according to user registration information and service accessing logs of special users, ensuring that users can enjoy user-centric services with high satisfaction. The recommendation is stored in the agent memory module.

C. Agent Memory Module

Personalized registration information and accessing logs are stored in this module, which can be utilized by the reasoning module to offer personalized and customized services for special mobile terminal users. First, we build user profiles to store the user information, including user preferences and contexts, etc, which play an important role in reasoning of service transferring. When the agent-based middleware receives user request, it will check the memory module to find services and determine the most appropriate services to specific users at first. However, if there are no corresponding services in the memory module, it will go to the service registration server to search and transfer the required services. In this way, the middleware can offer user-centric services.

D. Agent Management Module

The agent management module is responsible for maintaining the basic information of agents, including agent status management, agent registration, and agent resource allocation, which is designed into one container of agent development platform with its integrated management measures. Generally speaking, the agent management module can provide a favorable working environment for agents existing in the middleware. The directory facilitator management and agent communication management are also integrated in this module. After a function agent registered into the management contain, it will acquire a unique identification, including host address and agent name [13]. Moreover, agents communicate with each other with the common agent communication language.

E. Ontology Library Module

In order to solve semantic heterogeneity and message parsing, an ontology library is designed in the middleware. In the ontology library, many concepts relating to the function agent are used to describe the multi-agent middleware for transferring services. As a public vocabulary, ontology library can achieve knowledge sharing and reusing [14]. Herein, knowledge contains ordinary agent models and basic communication message between agents. And it is also useful for the communication and inter-operability among agents in the middleware. The aim of building the module is to shield off the heterogeneities of different operating systems and network communication protocols.

F. External Interface Module

The interface module takes charge of providing usable APIs for different mobile terminals and different application in order to assist them in transferring required services in mobile environment. In other words, mobile terminals can utilize the functions of the agent based middleware with different network accessing standards. Moreover, the module is responsible for connecting with the host server to transfer corresponding services and data files. The most important point is that these interfaces are platform-independence and flexible, having relatively strong expansion. In this way, the heterogeneity of mobile terminals and communication networks can be relieved [15].

After introducing the function module of the agent-based middleware, the process of service searching and transferring in mobile environments is presented: first, mobile terminal users have access to the middleware with the standard communication mode, and then send service requests with legitimate accounts, which can be filled in the form. And the middleware checks user validation and analyzes the user request, and then determines the required services according to some rules. Second, the agent-based middleware arouses one user agent to help terminal users find the most relevant services with respect to their requirements in the memory module. If it is successful, the middleware transforms the service messages into serialized byte streams, and then sends concrete service content with the proper format to mobile terminals. However, if the required services are not found in the memory module, the user agent aroused by the agent-based middleware will move to the service server to parse the service registration files in the server, in order to search the required services. Next, the middleware will transfer the corresponding services from host servers and convert them into the suitable format for mobile terminals. After that, the agent-based middleware

will return some serialized information to mobile terminals with asynchronous transfer mode, returning the results to terminal users. As mobile terminals, the de-serialize function is achieved to recompose java objects and display the corresponding response for users. In this way, a simple service transferring is over.

IV. SERVICE TRANSFERRING MODEL ARCHITECTURE

By combing web services and agent-based middleware technology, we design a service transferring model for mobile terminal users. In order to use the advantages of these technologies, we choose some mature standards and stable development platform, whose specific contents are discussed in the succeeding section. As demonstrated in fig. 2, there are three main components in our architecture: mobile terminals, agent-based middleware and service server. In the following, we have a brief description of each component:

1) Mobile terminals: In the model designed in this paper, mobile terminals can be any devices such as cellular phones, PDAs, laptops and so on, which are responsible for connecting to the middllware with the asynchronous mode and creating service requests according to the requirements of terminal users. Certainly, user application interfaces have been developed in the mobile terminal devices. Considering maybe operating systems of mobile terminals are Windows Mobile, Symbian, and Linux and so on, we choose the Java langauge to develop applicaions in mobile terminals because of its platform independence. When users determine their requests, they fill in the appropriate form of mobile terminal applications and submit information to the agent middleware. Then, the user-agent is aroused in the remote agent-based middleware, searching for required services and delivering service information to the end users. Mobile terminals can disconnect with the communication network temporarily, but independently, asynchronously and autonomously communicate with network servers, which are implemented with the asynchronous communication mode. It is proved that the asynchronous communication mode is indeed more suitable for applications in mobile networks. Moreover, there is a local record management system in mobile terminal devices for storing much temporary information for users.

2) Agent-based middleware: As mentioned above, the agent-based middleware acts as a mediator between the mobile service server and mobile terminal applications, which is responsible for discovering and transferring the most relevant services with respect to search keywords and user preferences, and then it transfers the corresponding services for users with the proper format, shielding off the complexity and technology details for service transferring. In other words, terminal users can make use of the middleware to transfer mobile network services transparently. The detailed function modules of the middleware have been discussed in section III.

3) Mobile service server: In order to ensure services in mobile environments to be discovered and transferred conveniently, we design and implement the service framework with SOA (service-oriented architecture) . In the framework, services are described to be a set of messages in XML according to WSDL (Web Services Description Language). At the same time, SOAP and UDDI are respectively applied to be corresponding standards for parsing messages and service registration. Using these uniform standards, the description, discovery, invocation of services are well supported. In general, service discovery module, service parsing module and service registration module are the core components. These components are separately implemented with the corresponding technology and standards in the SOA architecture. Certainly, the resource database is also built to store the data related to services.

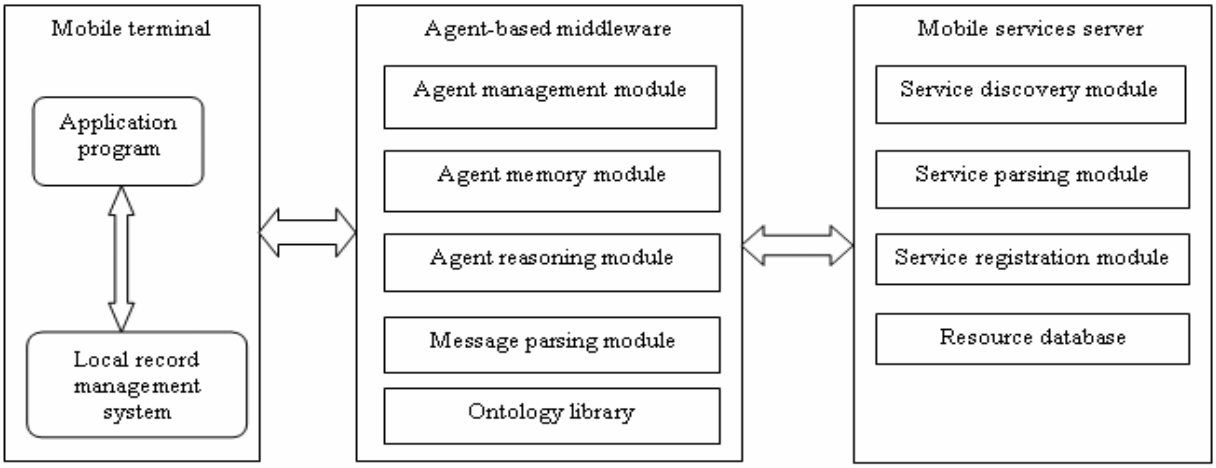


Figure 2. Components in service transferring model

V. MODLE ANALYSIS

As mentioned above, the service transferring framework combines the advantages of middleware and agent technology, which can provide efficient and user-centric services for mobile users. In order to implement the mobile service transferring framework in this paper with advanced technology, we also decompose the

framework into three main parts to achieve, including mobile terminal design, agent-based middleware and mobile service server deployment.

A. Mobile Terminal Application Design

In order to achieve platform independence, we choose J2ME to develop the mobile terminal application. For the sake of brevity, we put all codes of the mobile terminal application in one MIDlet, which is a basic executing unit of mobile terminal applications. At the same time, we also design favorable graphical user interfaces and basic storage units for terminal users, prompting them to interact with the middleware conveniently and efficiently. Users can submit service requests by filling some corresponding information in the form. Considering message sent by the agent-based middleware is byte stream, we implement de-serialize function in the terminal development, in order to ensure the service information can be displayed accurately and clearly in terminal devices, leaving the heavy parsing work and service transferring requirements to the middleware. For the connection between mobile terminals and the agent-based middleware, we designed one function of making connection in the middleware. First, one HTTP connection is opened for the communication between mobile devices and the agent based middleware with the standard communication protocol, and then the service request of terminal users is sent to the agent-based middleware. After the request is sent, mobile terminals can disconnect to the network temporarily and then re-connect to the middleware to receive responses. In order to achieve the asynchronous transfer mode, we use socket programming to transmit the information between mobile terminals and the middleware. In this way, data can be easily organized to comply with the specified protocols, and the transferring data in the connection is regarded as the input/output stream. With this communication mode, the workload of terminal devices can be lower than direct service accessing and so does the required processing capacity of terminals.

B. Agent-based Middleware

The agent-based middleware is the core component for the service transferring framework, which takes charge of many tasks instead of mobile terminals, such as XML parsing, service discovery and service transferring. First, we design some agents in the middleware and lots of rules in the reasoning module, including the parsing agent, delivery agent, serialization agent and search agent etc. Herein, every agent has its own function, for example, the parsing agent takes charge of XML parsing and converting the SOAP messages of mobile services to byte stream. Besides, we gather many reasoning rules artificially in advance. In order to manage agents effectively, we use Java Agent Development Framework (JADE) to construct the framework. The main container in JADE is responsible for maintaining the directories of agent identifiers and offering a working environment for agents [16]. In the agent manager module, we define a normal Java class to manage agents for mobile service transferring and set up channels for communications among different agents with ACL (agent communication language) messages. What's more, we utilize JADE-S to support the message integrity and confidentiality, which guarantee that information is not tampered by others during transmission. It is achieved by supporting data signature and encryption on the ACL messages [17]. In the memory module, we set up a user profile to store personalized information and service accessing records of users as shown in Figure 3, in order to provide user-centric services for mobile terminal users, and then improve user satisfaction. Moreover, we also utilize many measures integrated in the JADE platform to build ontology library. As for external interfaces, we encapsulate different APIs for external applications. All the functions are implemented in Java.

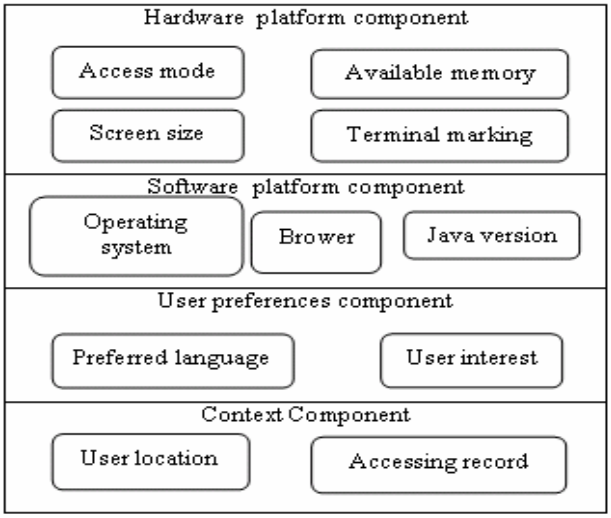


Figure 3. Mobile user profile

C. Mobile Service Server

For the sake of describing and discovering mobile services conveniently and efficiently, we design services in mobile environments. In this way, some independent function units are packaged as different service units. Specifically, all the mobile services are described in WSDL, which describes the functions, necessary parameters and returning value of special services. Meanwhile, SOAP is used to be the communication protocol for mobile services. And the parsing of HTTP requests, SOAP messages is implemented in the middleware. Furthermore, UDDI (Universal Description, Discovery, and Integration) is used to enable service providers to register and find mobile services. All the services in mobile applications need be registered with the registration server. When users want to make use of these services, they may obtain the registration information and parse the services with the middleware. Furthermore, in order to store the data and other resources for mobile applications, a database is built with Microsoft SQL Server, and the agent-based middleware system can invoke data by establishing JDBC (Java Data Base Connectivity) connection. Mobile agent technology is used to encapsulate the services for the mobile service

mobility. Transferring the state and execution stacks of agents is utilized to achieve the mobility of service.

VI. CASE STUDY

In order to validate the practicability of the solution, we have made a case study, which is helping mobile terminal users search for logistics information in mobile networks. First, One simple service is set up in mobile environments, which can help mobile terminal users search for logistics information. Using the service, terminal users can find out cargo and vehicle information between two different places. Certainly, the agent based middleware takes charge of parsing messages and transferring services from the host server. Next, we store related data into SQL server database, containing the origin, destination, and vehicle-no and some other data. In the middleware, we encapsulate service transferring and information parsing functions into two separate agents. Users can determine some querying conditions according to their own requirements and submit the query request to the agent middleware, analyzing requests and parsing service messages are finished in the middleware, and the middleware is also responsible for passing user requests and transferring corresponding services. Finally, user terminal devices display the response results in an definite format for terminal users. As previously described, the accessing traces are stored in the agent memory module in order to provide more accurate and user-centric services for specific users in the future.

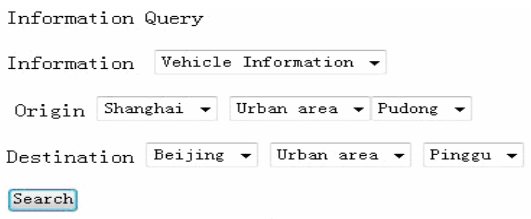


Figure4. User information query

Origin	Destination	Vehicle-No	Loadage	Go-off	Particular-Information
Shanghai Pudong	Beijing Pinggu	HuA12345	20ton	2009-10-16	View
Shanghai Pudong	Beijing Pinggu	HuA43527	18ton	2009-10-19	View
Shanghai Pudong	Beijing Pinggu	HuA25893	16ton	2009-10-21	View
Shanghai Pudong	Beijing Pinggu	HuB13864	20ton	2009-10-25	View
Shanghai Pudong	Beijing Pinggu	HuD18475	18ton	2009-11-16	View
Shanghai Pudong	Beijing Pinggu	HuA25867	16ton	2009-11-19	View
Shanghai Pudong	Beijing Pinggu	HuB54782	22ton	2009-12-16	View
Shanghai Pudong	Beijing Pinggu	HuC14558	18ton	2009-11-30	View

Figure5. Service transferring resul

Figure 4 and Figure 5 shows the search process, in Fig. 4, terminal users determine some query conditions and click the ‘search’ button to submit the request to the middleware. Fig. 5 shows the query result.

By combining the mobile agent with web services, the service transferring framework discussed above has favorable flexibility and extendibility. We use Java and the JADE platform to develop the middleware and design the mobile services. With the platform independence of these technologies, the middleware has well-behaved scalability and agility. Through building user profiles in the model, the middleware can provide user-centric services for terminal users, which can make users more satisfied. Because burdensome work is taken charge by the middleware, so the workload of mobile terminal devices is reduced, in order to prevent the troubles brought by limited processing and storage abilities of devices in mobile environments. As for the information transferring security, we use JADE-S tool to ensure data integrity and reliability. Overall, transferring mobile services based on the agent middleware can provide a flexible and efficient way for service transferring in mobile environments, and help mobile terminal users acquire more targeted and timely services

VI. CONCLUSION

In this paper, we introduce a service discovery and transferring framework in mobile environments with the agent-based middleware. This framework can enable us to deliver the most relevant services to terminal users, being compatible with user preferences and device profiles. Users can utilize it to transfer services in mobile networks conveniently without worrying about the processing abilities of mobile terminal devices. With the advantages of agent and middleware technology, the service accessing model proposed in this paper can improve the performance and efficiency of service transferring for mobile terminals. We discuss the architecture of the proposed framework and construct a concept implementation using Java and the JADE platform. It can eliminate the issues caused by the diversities of mobile terminal devices with different capabilities and other constraints. In the future we also hope to apply mobile grid technology to add the flexibility of the middleware and integrate some other mobile agent security mechanisms into our framework. How to import grid technology into the framework is also worth further study.

ACKNOWLEDGEMENTS

This research is supported by National High-tech R & D Program (863 Program) of China (No.2008AA04Z127) , Natural Science Foundation of Zhejiang Province (No.Y1090264) and Shanghai Leading Academic Discipline Project (No.B210).

REFERENCES

[1] K. Siau and Z. Shen, "Mobile communications and mobile services," International Journal of Mobile Communications, vol. 1, pp. 3-14, 2003.

[2] E. Sanchez-Nielsen, S. Martin-Ruiz, and J. Rodriguez-Pedrianes, “An open and dynamical service oriented architecture for supporting mobile services,” In Proceedings of the 6th international Conferenceon Web Engineering , California, USA, pp. 121- 128, 2006 .

[3] Q. H. Mahmoud, and Z. Wang, “Customizing and Delivering MobileServices using Software Agents and CC/PP,” In Proceedings of Consumer Communications and Networking Conference(CCNC), Las Vegas, USA, vol. 2, pp. 1114-1118, 2006.

[4] M. Wagner, W. Balke, R. Hirschfeld, W. Kellerer, “A Roadmap to Advanced Personalization of Mobile Services,” In Proceedings of

the 10th International Conference on Cooperative Information Systems (CoopIS) Industry Program, Irvine, CA, US, 2002.

[5] N. Elkarra, A Web Service Strategy for Mobile Phones, Available from http://webservices.xml.com/pub/mobile.html;November 18, 2008.

[6] M. Basuga, R. Belavic, A. Slipcevic, V. Podobnik, A. Petric, and I. Lovrek, "The MAgNet: Agent-based Middleware Enabling Social Networking for Mobile Users", Proceedings of the 10th International Conference on Telecommunications (ConTEL'09), Zagreb, Croatia, 2009, in press.

[7] C. L. Fok, G. C. Roman, and C.Y. Lu, "Agilla: a mobile agent middleware for self-adaptive communicationsensor networks," ACM Transactions on Autonomous and Adaptive Systems, pp.1-26, 2009.

[8] M. Baguga, R. Belavic, A. Slipcevic, and V. Podobnik, "The MAgNet: Agent-based Middleware Enabling Social Networking for Mobile Users," Proceedings of 10th International Conference on Telecommunicatons, pp. 89-96, 2009.

[9] L. Vrdoljak, I. Bojic, V. Podobnik, and M. Kusek, "The AMiGO-Mob: Agent-based Middleware for Group-oriented Mobile Service Provisioning," Telecommunications, vol. 8, pp. 97-104, 2009.

[10] J. Cao, D.J. Kerbyson, and G.R. Nudd, "HighPerformance Service Discovery in Large-scale Multi-agent and Mobile-agent Systems," Software Engineering and Knowledge Engineering, vol. 11, pp. 621-641, 2001.

[11] M. H. Raza and M. A. Shibli, "A mobile agent middleware for multimedia services," Advanced Communication Technology, pp. 1109-1114, 2007.

[12] A. Ornicini and G. Rirnassa, "Towards seamless agent middleware," Enabling Technologies: Infrastrusture for Collaborative Enterprises, pp.417-422, 2004.

[13] K. Jurasovic and M. Kusek, "Verification of Mobile Agent Network Simulator," Lecture Notes In Computer Science, vol. 4496, pp. 520-529, 2007.

[14] S. Mundle, N. Giri, A. Ray, and S. Bodhe, "JADE based multi agent system for mobile computing for cellular networks," Proceedings of the International Conference on Advances in Computing, Communication and Control, pp.467-473, 2009.

[15] A. Shemshadi, J. Soroor, and M. J. Tarokh, "Implementing a multi-agent system for the real-time coordination of a typical supply chain based on the JADE technology," System of Systems Engineering, vol.2, pp.1-6, 2008.

[16] Y. H. Fu, R. Piplani, R. D. Souza, and J. R. Wu, "Multi-agent enabled modeling and simulation towards collaborative inventory management in supply chains," Simulation Conference Proceedings, vol. 2, pp.1763-1771, 2007.

[17] L. S. Yu, E. Masabo, L. Tan, and M. Q. He, "Multi-agent automated intelligent shopping system (MAISS)," *Young Computer Scientists*, pp.665-670, 2008.

Weidong Zhao was born in Xuzhou, China, in December 1971. He received his Master degree from Hefei University of Technology, China in 1998 and Ph.D. in system engineering from Southeast University, China in 2001. He worked as a post-doctor at School of Management, Fudan University from June 2001 to May 2003.

He is an associate professor and the director of electronic business centre, School of Software, Fudan University, China. In recent years, he has published more than 80 papers in software engineering, information management, electronic business, etc in some important journals and international conferences. His interests include electronic business and business intelligence.

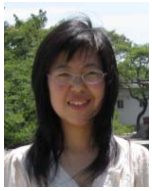
Dr Zhao is also a member of Shanghai E-business Expert Service Team.

Haifeng Wu received his B.S degree in network engineering in 2008 from Yunnan University, and he is current a master student at School of Software, Fudan University, China, whose major research fields are mobile commerce, business intelligence and supply chain management. He has published about four papers during the post-graduate.

Weihui Dai received his B.S. degree in Automation Engineering in 1987, his Master degree in Automobile Electronics in 1992, and his Ph.D. in Biomedical Engineering in1996, all from Zhejiang University, China.

He worked as a post-doctor at School of Management, Fudan University from 1997 to 1999, a visiting scholar at Sloan School of Management, M.I.T from 2000 to 2001, and a visiting professor at Chonnam National University, Korea from 2001 to 2002. He is currently an Associate Professor at the Department of Information Management and Information Systems, School of Management, Fudan University, China.

Dr. Dai has published more than 110 papers in Software Engineering, Information Management and Information Systems, Complex Adaptive System and Socioeconomic Ecology, Digital Arts and Creative Industry, etc. Dr. Dai became a member of IEEE in 2003, a senior member of China Computer Society in 2004, and a senior member of China Society of Technology Economics in 2004.



Wenjuan Wang received her bachelor degree in information security in July 2008 from Yunnan University, China. She started her master degree in system analysis and integration in September 2008, at School of Software, Yunnan University, China. She is current a joint student at School of Software, Fudan University, China. And her research direction is electronic business.

Guangxue Yue obtained his master in Hunan University. He is a professor at the College of Mathematics & Information Engineering, Jiaxing University, China. His main research interests include Distributed Computing & Network, Network Security, and Hybird & Embedded Systems. In these areas he has published above 30 papers in leading journals or conference proceedings, above 20 papers are indexed by SCIE, EI. He served as many workshop chairs, advisory committee or program committee member of various international IEEE conferences, and chaired a number of international conferences such as ISECS'08, ISIP'09, ISECS'09 and ISISE'08. He have taken as a guest researcher in Jiangxi University of Science and Technology, Jiangsu Polytechnic University, State Key Laboratory for Novel Software Technology at Nanjing University, Graduate School of Chinese Academy of Sciences, Guangdong Province Key Lab of Electronic Commerce Market Application Technology.

.