Procurement Management System Design and Implementation Based on Mobile Devices

Tony Tsang, *Member, IEEE*

***Abstract*—Many domestic enterprisers not only procurement costs, labor costs remain high. Especially with the continuous development of enterprisers, procurement has expanded the traditional mode of operation and methods of procurement cannot meet the needs of modern enterpriser procurement. In order to achieve the pursuit of efficiency and effectiveness, the establishment of procurement management information system is very necessary. In this paper, analysis of the design of purchasing management of domestic enterprisers on the basis of the procurement and put forward their own views of management information systems, and we using mobile devices technology to establish an experimental implementation model.**

***Index Terms*—Procurement Management, Mobile Devices, Information System, Mobile Agents, Database**

I. Introduction

PROCUREMENT system [1] is a collective term for a range of business software solutions which utilize the latest information and communication technologies (especially the mobile networks), that can be employed to automate the internal and external processes associated with strategic sourcing and purchasing, which includes catalogue search, item requisition request, approval, purchase order, delivery, receiving, payment, identification of sourcing opportunities, supplier evaluation, negotiation, and contract. Examples of the business software solutions include procurement software, business-to-business (B2B) auctions, B2B market exchanges, purchasing consortia, tendering, auctions, procurement platform, and sourcing. The business software solutions that aid the procurement process automation utilize the latest information and communication technologies, such as electronic data interchange, Internet, mobile networks, e-mail, software and hardware, but exclude other means of communication such as telephone, fax, and so forth. Procurement technologies fundamentally aim to hammer out inefficiencies associated with manual-based procurement systems. Procurement platform are able to provide suppliers with constant, direct and inexpensive access to the global marketplace and business services, and offer buyers a friendly environment for procuring their products and negotiating terms online. This leads to: a reduction in search or advertisement cost; the identification of a larger number of prospective partners; and to the improvement in the efficiency of inter-organization business processes. For example, procurement platform can facilitate interactions between domestic suppliers and their potential overseas buyers by providing domestic manufacturers and traders with purchase inquiries and other information. It also offers local businesses maximum exposures to potential international markets and investors.

The scalability, complexity and dynamics of procurement platform present a great challenge to the system developers and business operators as procurement platform operate in open large-scale application environments in which their sub-systems are geographically dispersed. It may involve various application packages developed by different companies. Numerous efforts have been made to promote procurement platform. However, current procurement platform and business services do not have sufficient facilities for modeling sophisticated trading activities. It does not provide mechanisms for automating business transactions or the integration of services [2]. Other business services systems attempt to facilitate basic commerce such as: B2B purchases of supplies, mission critical B2B commerce transactions, such as microprocessor purchases by a PC manufacturer. The majority of the above trading processes are performed offline, and they are not fully integrated with buyers' and sellers business systems. As such, the lack of a general and systematic approach limits the adoption and usage of procurement platform by a wider business community.

Procurement platform [2] requires technologies which provide support for the adaptability, flexibility and the integration of business applications in order to meet rapid changes in business environment. Traditional web technologies are unable to meet the new business requirements which are derived from the rapid growth of e-procurement. In this research, a service-oriented framework is proposed in order to address the issues associated with the development of procurement. Service-Oriented Architecture (SOA) [3] development paradigm has emerged to improve the critical issues of creating, modifying and extending solutions for business processes integration incorporating process automation and automated exchange of information between organizations. Web services technology follows the SOA's principles for developing and deploying applications. Besides, Web services and mobile networks are considered as the platform for SOA, for both intra- and inter-enterpriser communication [4].

This paper describes a framework of web service-based business-to-business electronic procurement platform on

Tony Tsang is with the Hong Kong Polytechnic University (corresponding author to provide e-mail: ttsang@ ieee.org).

mobile agents. The rest of this paper is structured as follows. Section 2 & 3 present Procurement Management Overview as well as Service-oriented architecture. Section 4 proposes implementation architecture of the procurement services system. Section 5 expresses the conclusion.

II. PROCUREMENT MANAGEMENT OVERVIEW

Procurement management is the business plan in order to achieve production or sales, from the right suppliers, to ensure proper quality of the premise, at the right time, right price, purchase the appropriate number of goods taken by management activities. That "SR" management: The appropriate suppliers (Right Vendor), the appropriate quality (Right Quality), the right time (Right Time), the right price (Right Price), the appropriate number (Right Quantity). Procurement management is to study the process of obtaining the goods, overall pre-planning, implementation and post things in control, to achieve the maintenance of normal business activities, the purpose of reducing production and marketing costs [5]. In terms of procurement planning should include setting goals, formulating plans, the establishment of systems and organization, division of responsibilities and authority, personnel selection, design processes and forms, and supervision and evaluation. Classified according to different methods, Procurement management can be divided into the following categories: Accordance with the procurement of the main different classification: corporate procurement, government procurement, public procurement units, military procurement, and the procurement of other social groups. Scientific level by purchasing Category: traditional procurement, scientific procurement (point of purchase orders, lIT purchasing, MRP procurement, supply chain, procurement, bidding and purchasing, e-procurement). Divided is according to the scope of procurement: domestic procurement and foreign procurement. Procurement authority by Category: centralized purchasing, decentralized procurement. Morphological classification according to the procurement of materials: purchasing tangible goods (raw materials, energy, auxiliary materials, semi-finished products, components and finished products), intangible goods procurement (technology, services), engineering procurement.

III. SERVICES-ORIENTED DESIGN

The procurement services system that we have developed is based on the Service Oriented Architecture (SOA) and uses Web Services. Before designing and developing the system, we have to decide that what technology is the best choice for implement the e-procurement system. After comparisons and considerations, XML web services are the best choice under our consideration because of some reasons as follows.

A. Service-oriented architecture

The Service Oriented Architecture [6] reinforces basic software architecture principles such as abstraction, encapsulation, modularization and software reuse. It provides well defined interfaces for client applications and separates the interfaces from their implementations. It allows service capabilities and interfaces to be implemented as a collection processes. Each process itself provides a service, one that offers a particular capability. Because each process is exposed through a standard interface, the underlying implementation of the individual service is free to change without affecting how the service is consumed. The Service Oriented Architecture not only encompasses the services from a technology perspective, but also includes the policies and practices by which the services are provided and consumed. Security and privacy are particularly important issues for education. Personal information is confidential, so access to such information must be restricted to authenticated and authorized users. Secure transmission of such information must be complemented with secure storage of the data. The use of the Service Oriented Architecture is critical for enforcing such policies.

B. Service-oriented architecture

Web service [6] is a technology that has built providing various types of services over a web. The main advantage of using a web service technology is cross platform communication. Recently (2005) there are two major competitor of web service technology in the market that is Microsoft and Sun. From implementation point of view both using common standards and protocols, such as Simple Object Access Protocol (SOAP), Extensible Markup Language (XML), Web Service Description Language (WSDL) and Universal Discovery Description & Integration (UDDI). SOAP is an XML-based message exchange protocol that is used to communicate between web services and their clients [7]. With the help of this lightweight protocol we can exchange structured information in a decentralized distributed environment easily. WSDL provides description of a web service. Each web service has WSDL file which is basically an XML file that describes a set of SOAP messages and how the messages are exchanged between web services and clients. UDDI is often called yellow pages of web services. A UDDI is directory of web services that have XML files describes a business and the services it offers. We will use UDDI in our architecture to expose our educational services so other can take advantages from these services.

IV. PROCUREMENT SYSTEM IMPLEMENTATION

In this paper the implementation of the procurement system will be introduced in detail from the Mobile Agent layer, the Web layer, logic processing layer and the database layer.

A. Architecture of Mobile Agent with M client P2P server

The reason for the name M client P2P server is in the fact that the architecture combines principles of peer to peer architecture and client server architecture. Role of M client was strengthened in later versions of architecture design as it is the most important part of architecture and consequently the name M client server appears even more adequate. This kind of architecture and its principles will be examined below, as shown in Figure 1.

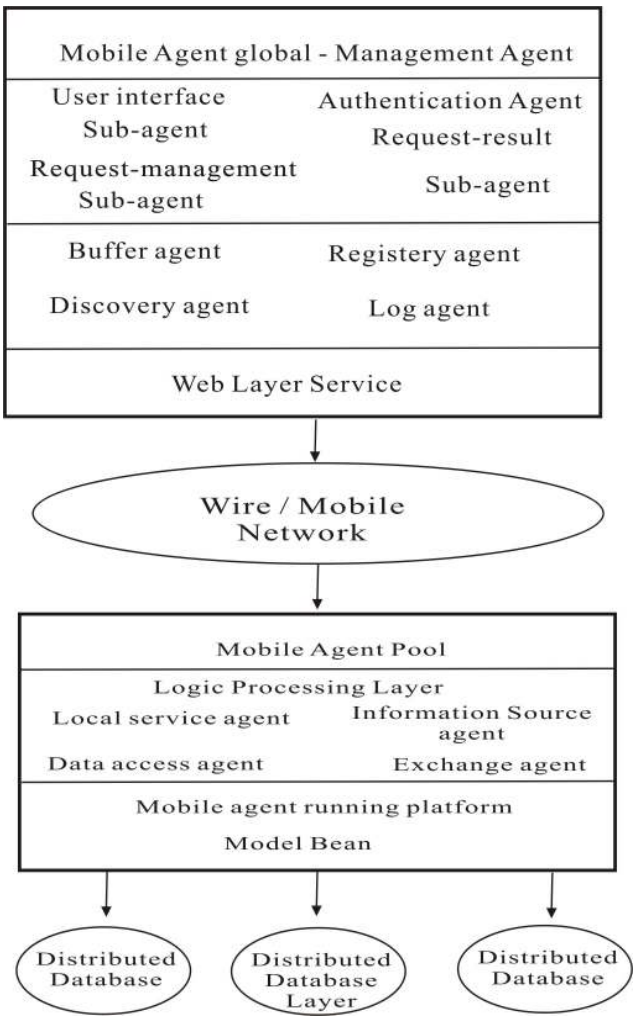

Fig. 1 The physical structure of Procurement System Framework

The architecture solves the problem of connection / collaboration of mobile devices in the network. The central part of the software system is a server (main server). Each mobile device while registering into network also registers to this server. Subsequently the device obtains the list of all other connected devices, description of the services and information necessary to the initialization of the connection with these devices from the server. The list may even include information about computation performance. This information may be of use in analysis of performance, of data transfer etc. Each device provides server the same information about itself during the process of registration. The first client obtains the minimalistic list having one device only as no other device has ever been connected to server. The list of devices is updated in some periods or in random moments of time when device asks server to up-to-date list of connected devices from the server.

The list has to be marked by a time stamp of some kind, that enables to decide if the list is up to-date or not. The just described solution combines features of client server architecture and peer to peer architecture. Consequently it is called M client P2P server architecture or M client server architecture, as mobile client is denoted as M client. The client may also act as an autonomous client in the scope of this architecture.

Similarities between M client server architecture and agent based architecture will be found and examined in this section. Main emphasis will be on the way of communication. Hybrid client is similar to one software agent or set of software clients being in one node. Multi agent paradigm (approach) is based on some autonomy of the agent that depends on the power and ability of application layer implemented in the client [8]. Thin client is more dependent (less autonomous) than the hybrid client. The agent may meet its own aims or it may follow the aims of another agent. The first mentioned agent is analogous to simple client and the second one is analogous to hybrid client. In spite of the fact that the architecture include server as essential element it can be considered decentralized and similar to multi agent system. Decentralization is enabled and manifested by many mutually replaceable clients (nodes) possessing the role of server when needed. Architecture can be further generalized and brought closer to multi agent paradigm. We can comprehend it as the network of hybrid clients analogous to clients that make use of other clients and at the same time offer some service for them. Mobile agents are software agents whose code and data can migrate among nodes or among locations. Typical purpose and result of such arrangement may be data volume reduction transferred among nodes. Many similarities between just described architecture and multi agent architecture can be found and used. Consequently it is possible and helpful to implement the architecture using an available framework developed initially for multi agent systems. One good example of such framework is JADE for Java that satisfies all important requirements, mainly possibility of multi-platform runtime support and mobility support. This open source framework is well maintained and clearly documented in literature and on the internet. Agents in the JADE environment are placed into so called container that is adequate to JVM process. The communion is established and kept in the scope of so called platform (set of containers). One container starts connection to other container. Special DG agent has the role of the mediator and its role is to gain target address of some container. This agent is placed in special main container. Consequently this architecture is not fully decentralized, but its weakness can be reduced by replication of the functionality of main container to several containers. Basic implementation relations are in the following table.

Mobile environment and similarly more general environment of wireless networks brings some specific problems related to lower reliability of network (transmission media), limited bandwidth and limited performance of client devices. These problems and the way of their solution may substantially influence the architecture of the resulting communication infrastructure. Internet networks were initially designed as transparent and enabling mutual communication of any two nodes. But the present situation is substantially different. Firewalls and translators of addresses (NATs) form two main

barriers of direct communication.

Table 1: Implementation concept of basic of M client features in JADE

The Component of architecture	Implementation in JADE
M Client	Agent in the container
Internet, GSM Network	Internet, GSM Network
Server Sources of data	Agent in the container, source of data
Server source of information about nodes	DF agent in the main container

Communication between two mutually shaded nodes has to be mediated by another node, which is often called the active node, the super node or the server. This solution reduces decentralization of the system. Other mediators are necessary to improve / keep reliability and performance. Mobile devices cause additional problem. The address of client may change during the process of its shift from one to another location. Again, this problem is often solved by mediator, special home node of mobile node that mediates communication with it. JADE architecture solves this problem by special extension LEAP that enables to split the container into two parts: the Front End and the Back End. The Front End part of the container is kept on mobile device (this is light weight part of the container) due to limited system means of the device. Back End part of the container is its server part that runs on stationary device accessible from the network by front end part of the container and by other communicating containers. Connection between the Front End and the Back End part of the container is established by the Front End part. It enables reliable transit (gateway) through firewalls and translators of addresses, NATs. The Front End also possesses the functionality for restoring the connection in case of interruption. In addition to it both Front End and Back End are equipped with data structure for storing messages directed to the other part in case the messages cannot be delivered due to failure of connection. The main disadvantage of this solution is a direct dependence of mobile device on its Back End, which cannot be replicated. Interesting possibility for transparent communication may be the use of some VPN (Virtual private network) based solution. Open source implementation of L2 VPN supporting peer to peer communication called N2N is one of such possibilities. This VPN is used for mediating mutual communication of nodes. Server node (super node) mainly gains information about other nodes. Communication between nodes is established via server only in case it is impossible to establish any direct node to node connection between them. From the application point of view given node communicates with other nodes through L2 or L3 layer. Consequently it is possible to use broadcast mechanism for finding nodes that can be reached. Broadcast can be also used for global distribution of information for all nodes. N2N mechanism supports increasing reliability by replication of functionality of super-node into two different nodes. The solution using participation of each node in more VPN is also verified. Each VPN has one super node or two nodes. But N2N solution is not prepared for more common situation, in which each node could mediate communication. We should note that in this architecture super node doesn't necessarily need to directly correspond with the hybrid client. In case we want to fully implement our architecture using agent approach, it is necessary to combine two technologies, for example JADE framework and N2N VPN technology. In this arrangement VPN ensures transparent communication for JADE. This is shown in the next Table.

Table II: Communication Layer in case of more Technologies

Application logic (m-client)
Agent framework (JADE)
Transparent network (ensures N2N)
Wireless network

JADE and other multi agent frameworks support mobility of agents i.e. possibility to transfer code and data from one node to another node. For this purpose JADE framework uses standard Java mechanism of serialization/de serialization and transfer of classes through network. The code of agent may also contain call back methods for solving (handling) the events of opening and closing the migration that typically serves for releasing of resources in initial location and allocation of resources in the new location. JXTA for Java that implements transparent peer to peer networks may serve as another possible communication platform. In comparison with N2N JXTA has more complete and more complex platform (it includes for example more types of nodes) and enables greater degree of decentralization. In contrast with N2N which provides the virtual network card JXTA has native API for communication. It also provides functions for detecting other nodes and their services, for transparent fronts of messages etc. JXTA may also serve as a complete platform for implementation M client server architecture without necessity to use other agent-based framework.

B. Web Layer Implementation

Servlet can be considered a server-side Applet, and it has been loaded and implemented is equal to that Applet be loaded and implemented [9]. When the form defined in JSP was submitted, this request was sent to a Servlet which will find service Bean and entrust the task of data checks and entity creation to it. This Servlet code is stored in file Createproduce.java. When requesting drive, Servlet request may contain multiple data items. And when Web container received a request to the Servlet, it will be packed into an Http Servlet Request object, and then this object will be sent to the corresponding service metho*d.*

C. Logic processing layer implementation

Logic processing layer is responsible for processing any operation, including EJB. EJB is a server software component can be distributed in multi-layer environment. The interface between EJB component and its container is defined in EJB specification. Generally, the server vendors achieve the development tools of the server, EJB container and distributing the EJB components to the EJB container. EJB containers provide service for the EJB examples they contained, and the services include the control of affairs, management safety, the use of thread or other resource connection pool technology, processing persistent data, as well as other high-level system tasks. Application developers only need to compile the disposal of the EJB components in business functions, rather than considering the problem of container complexity realization. EJB technology is the main basis of J2EE. It provides a scalable framework for implementing the application logic in distributed calculation environment. EJB components can be divided into Session Beans, Entity Beans and Message-driven Beans three categories. Session Bean represents the request of the client to the server, and it implements the processing flow of the business logic. The life cycle of the session Beans is the same as the duration of the session at the client, which represents a short conversation with the client. Entity Beans is used to deal with the persistent data, rather than business logic, and it represents the actual data object and the method affected in the data object. Entity Beans provides an object oriented view of memory in the data storage layer and a mechanism which combines the data persistent function and the object encapsulation. In addition, it corresponds to the data storage layer of multi-application architecture. Message driven Beans is similar to the Session Beans in behavior. The difference between them is that it can be transferred only through sending message to message-driven component. And message-driven components can also transfer other EJB, Beans. A typical EJB structure is composed by three parts, namely EJB type, EJB object and Home object. Firstly, EJB client find Home objects through the directory service, and then find or create EJB objects through Home interface, and return to the citing of the EJB objects. EJB Object provides a way the client accessed the EJB container. The EJB client accesses the EJB component object through EJB container, and all the requests to the EJB component objects are took over by the EJB container. EJB clients can be a Java Servlet, Java Applet, or other applications. Logic processing layer provides a type of component model to access the distributed system services and persistent data. Independent clients and the Web application in the Web layer all can use the EJB components remained in the logic processing layer. The logic processing layer separates the business logic related to the application from the implementation detail of the system level services, which also separates the roles of application developers (interest in application function) and publisher, this is because that the container manages all the system level problems, such as durability, re-entrant, affairs and remote access, logic processing layer also simplifies the application component development.

D. Database layer application

Here Microsoft's SQL Server 2000 is used as the database, which is a relational database management system with two dimensional tables for the basic management unit. Therefore, the object model is ultimately described by two-dimensional forms and the relationship between them. Object model needs mapping to the database model, that is, the object is required to be transformed to the database. The multi-tables mapped out from an application objects after standardization disposal are regarded as a database object. When some application requests are changed, the system will firstly modify the part which does not involve the part without request change. Secondly, the modification of the amendment only limited to add or delete program module, or add new database table, and basically do not have to modify the original code or the original database table definition, which greatly reduces the workload and reduces the work difficult. Database middleware is a middle layer between the frontend client and back-end database, which completes all the data logic operation, simplifies the client and plays a role as a bridge between the client and the server. After received the user's data request, the client will find different database middleware by using JNDI according to different data requests, the corresponding database middleware will find the corresponding data source database according to JDBC, and then the middleware will use SQL language to extract the corresponding data of the database for various logic processing which then will be returned to the client to display. In this model the most used mode is Session Facade design pattern, which can mostly reduce the dependence between the client and the server, and it stresses that the relevant example should be completed in one network transmission and one affair. Logic processing layer can be divided into two levels: data access logic part and data access interface part. Domain Model is the core of the system accessing the database, and according to the design object, the part is achieved through using the BMP entity Bean. The advantage of the BMP entity Bean is the persistence of the data it managed itself, so that greater flexibility. Data logic layer serves for data access, and in accordance with the mode definition of Session Facade, the layer will develop the corresponding Servlet and Session Bean according to the system affair division and specific business logic. The design object of Domain Model is to achieve common interface with the database, while the layer design should meet specific design and implementation independent of the database. BMP entity Bean is a type of entity Bean need to achieve their own data access logic, in the Bean JDBC is usually used to maintain the persistent EJB field. The basic idea of the Domain Model design is to use the BMP entity Bean in two management levels to complete the database management, and one of the Beans manages each user table in the management database, the other manages each record in the corresponding tables. EJB loading process is as follows: First of all, obtaining the object (java.sql.Connection) connected with the database and using which to obtain the DatabaseMetaData class object of designated database, and then using DatabaseMetaData class object to obtain the primary key of the table EJB primary key designated. In addition to constructing EJB components to achieve the business rules according to the structure of

functional module and processing flow, the logic processing layer main uses object agent and service locator to build and improve the framework. Object agent is achieved by using an ordinary class. And the session appearance is achieved through session bean, when the business representatives and session appearance are used together, there is one-to-one relationship between them, the reasons are as follows: the logic packed in one business representative may interact with a number of business logic services, thus, between business representatives and business services there will be one-to many relationship, but thus multi business services will sometimes assembled in one session appearance, so between the business representatives and session appearance there is also one-on-one relationship. Business representative is a logical abstraction, rather than physical abstraction, when in collaboration with the presentation layer, the business representative is actually in the presentation layer. However, these components are an extension of the logic processing layer and closely linked with the session appearance. Therefore, the business service layer and the presentation layer can use a same service locator, and the whole system only needs one server, which can be achieved through using a single mode.

E. EJB deployment

The EJB components must be configured and deployed in the J2EE platform after they were developed. EJB mainly achieves business logic, which can be stored in different application servers [5]. When achieving the business logic, these business logics should be transferred through the application server on the client. To transfer the business logic, the EJB which achieved the business logic must be firstly deployed in the server to transfer. Here Ant tool is borrowed to deploy EJB. Ant is an open-source tool with Java-based structure, is a part of Apache Jakarta project. Ant uses XML as its structure description language which composed by some target with finite state. The developers or structure managers define the attributes, transform the various tasks into a series of target in Ant configuration file, and define the dependencies between the targets. The developers of Ant provide a series of tasks which can structure and deploy EJB, and after the completion of the deployment, the EJB generated packets can be used not only to the web logic, but also to other application servers. Only through appropriate configuration of the ant file of one project, the EJB can be structured by using the XML sentences with similar elements.

V. CONCLUSION

In this paper we describe the design of a distributed procurement management system that uses an SOA as a model for deploying, discovering, integrating, implementing, managing, and invoking supply chain service system. Such a model could help the procurement, area to develop cost efficient and dependable supply chain services. Our architecture provides scalable environment where you can you're your services easily without interrupting the core architecture. You can add new procurement web service and register your service using mobile agent or other directory service. Maintainability of procurement services is easier than before because procurement management services are divided with respect to category and, functions. Mobile agents are always flexible as they can move in a network to find information; our web service can communicate with other web services in a network using standard-based protocols such as Java systems. Our research continues in the field of architecture implementation. Our purpose is to develop a prototype that allows the implementation of the architecture shown.

REFERENCES

[1] Zhai Dongsheng, Liu Lancui, ``A Framework for B2B E-Procurement Platform Based on Agents and Web Services", International Conference on Computational Intelligence and Software Engineering, CiSE 2009, Pages 1-5, 11-13 Dec. 2009.

[2] Yinsheng Li, Xiaohua Lu, Kuo-Ming Chao, Ying Huang,Muhammad Younas, ``The realization of service-oriented e-procurement platform", Springer Science, pages 307-319, 2006.

[3] Giner Alor-Hernandez, Ruben Posada-Gomez, Juan Miguel Gomez-Berbis, and Ma.Antonieta Abud-Figueroa,``a web service-based brokering service for e-procurement in supply chains", Springer Science, pages 686–693, 2007.

[4] Stanley Oliver, Kiran Maringanti, ``Application of Web Services in the Context of E-Procurement: SME", Foci. E-Procurement in Emerging Economies: Theory and Cases, pages 265-305, 2007.

[5] Guo Jun-wei, Li Lei-jun, ``Procurement management information system research and implementation based on Asp.net", IEEE 3rd International Conference Communication Software and Networks (ICCSN), pages 634 - 637, 2011.

[6] Parichat Pasatcha, Komrhon Sunat, ``A distributed e-education system based on the service oriented architecture", IEEE International Conference on Web Services, pages 791-794, 2008.

[7] Tsang Tony, ``An XML-based architecture for distributed real-time multimedia systems", IEEE Global Telecommunications Conference, GLOBECOM '01, pages 2005 - 2008 vol.3, 2001.

[8] Maly Filip, Kriz Pavel, Slaby Antonin, ``Mobile oriented software architecture M client M client server",33rd International Conference on Information Technology Interfaces (ITI), Proceedings of the ITI, pages 109 - 114, 2011.

[9] Zhu Zhao-you, Dai Sheng-hui,``J2EE-based enterprise ERP system design and implementation", 2nd IEEE International Conference on Computer Science and Information Technology, pages 509 - 512, 2009.

Tony Tsang. (M'2000) received the BEng degree in Electronics & Electrical Engineering with First Class Honours in U.K., in 1992. He received the Ph.D from the La Trobe University (Australia) in 2000. He was awarded the La Trobe University Post-Graduation Scholarship in 1998. He is a Lecturer at the Hong Kong Polytechnic University. Prior to joining the Hong Kong Polytechnic University, Dr. Tsang earned several years of teaching and researching experience in the Department of Computer Science and Computer Engineering, La Trobe University. His research interests include mobile computing, networking, protocol engineering and formal methods. Dr. Tsang is a member of the ACM and the IEEE.