ISSN-L: 2223-9553, ISSN: 2223-9944
Vol. 4 No. 1 January 2013

# DECOMPOSING OF SOA ON FILE TRANSFER of WEB SERVICE ON WINDOWS AND MOBILE ENVIRONMENTS

**Abdulkarim Kanaan Jebna[1], Massudi Mahmuddin[2]**

[1]College of Arts and Sciences, Universiti Utara Malaysia(UUM),
[2]College of Arts and Sciences, Universiti Sains,
MALAYSIA.

[1]a.kanaan@msn.com,[2]ady@uum.edu.my

## ABSTRACT

*There has been a great progress in the development of miniaturized devices and their demands have increased as well. In addition, people look forward to having similar programs on their stand-alone computers to run on their mobile phones' screens. As a result, the complexity of building a program has increased in this case because same programs are needed to run on two different platforms. One of the common programs among users is the file sharing system. In this paper, we will look at how decomposition architecture of service oriented architecture (SOA) can be used for file sharing system in both desktop computer and mobile phone environments. In addition, we investigate how to decrease the complexity of building systems on stand-alone computers as well as mobile devices. As a result, we have a file sharing system which is able to run on windows and mobile environments using the decomposition approach.*

**Keywords:** file transfer, destop computers, mobile phone, SOA

## INTRODUCTION

Two billion mobile phones are estimated to be used worldwide (Rheingold as cited from Kim, 2006). The success of mobile phones lies in the fact that is that mobile phones are easily available at anytime and anywhere (Schmidt, Stuhr & Gellersen, 2001). Mobile phones are portable and people feel unhappy without them (Ventä, Isomursu, Ahtinen & Ramiah, 2008). At the same time, the demand of mobile phone applications is increased rapidly by the developments of their operating systems. In the past, transferring short text messages and making calls were the only features mobile phones had (Yang, Chen, Chen & Hsu, 2008). That is due to a few reasons such as the lack of memory size, processor speed, and restrictions of data communications. Mobile evolution can be seen from a simple voice communication device to a powerful that is able to manage many types of files including multimedia documents.

The growth of mobiles has happened on hardware and software (Palazzi, Bujari & Cervi, 2009). We can see the mobile phone operating systems (such as Windows Mobile Phone and Symbian) are much more powerful and many functions. Moreover, the mobile phone has bigger storage in various major communication mediums such as UMTS, Bluetooth, GPRS, EDGE, and Wi-Fi (Palazzi, Bujari & Cervi, 2009). We also can witness various programs available to the users in both desktop computers and mobiles. In addition, common files are compatible with the two different platforms. As a result, mobile phones have become more reliable due to the improvements discussed earlier.

At the same time, the complexity of mobile applications has also grown. Users' expectation on both platforms has increased each year. Users are attracted to an application that can be

Copyright © 2013 SAVAP International
www.savap.org.pk

deployed in both mobile phones and that similar to desktop computer (Palazzi, Bujari & Cervi, 2009). File sharing application is one of these applications and its one of the ubiquitous and common applications used in PCs (Hofeld, Tutschku & Andersen, 2005). A mobile phone has data transfer capability (Biström & Partanen, 2004). Nevertheless, a file sharing application may require a large amount of bandwidth, if P2P network approach is used.

Although, there is a great application development in mobile phones, yet there are many limitations in constructing a program that is able to run on mobile phone. In 2/2.5G network, mobile terminal must provide the IP addresses to communicate. In other words, mobile phones cannot have true end-to-end connections. As a result, the IP address will change in no time without any warnings. However, this characteristic is sustained by 3G network (Biström & Partanen, 2004).

In this work, we investigate on how file sharing application effect of downloading on Microsoft Windows platform to a 3G network mobile phone. We are also adapting decomposing technique on service oriented application (SOA) during the application development.

The rest of the paper is organized as follow: next part explains related works of mobile phone and decomposition technique in SOA. The main component of the proposed decomposition approach is described next. Result of the proposed approached is presented and finally this work is concluded after that.

## RELATED WORKS

### Mobile Phone Development

Mobile software have been developed and improved over the time. Since the first generation of mobile phones, all the concerns were about making calls, saving contacts' phone numbers, sending and receiving short messages (SMS). By the time, mobile phones have developed many times. They are much faster and have much more room of storage. In addition, many features have been added to them such as cameras, Bluetooth, WI-Fi and GPS. So, many things can be done. Embedding cameras of mobile phones give opportunity to capture pictures as well as videos. In addition, the pictures can be viewed to both TV by using an appropriate cable and computers. Furthermore, it can be published on the Internet.

Studies on file sharing systems have attracted many especially in the fixed network such as BitTorrent (BitTorrent, 2010; Limeware, 2010). The idea of file sharing systems is the decentralization of shared files, so we do not have to locate them on a specific computer which is called the server. As a result, the shared files will be downloaded from client computers as long as they are connected to the network. In this works, we are adopting the same idea on the mobile phones so that a mobile phone will act as a stand-alone computer on the network. In this state, shared files might be downloaded from desktops or mobile phones.

Microsoft is developing an operating system for mobile phones known as Microsoft Windows Mobile. PCs' applications cannot run on mobile phones because they have a different architecture of hardware and software. It is better if we can minimize the effects of developing a system for cross-platforms by using kind of technology or architecture that will make building heterogeneous systems easier. Consequently, a developer can keep focusing on building the system regardless of the type of operating systems. It is therefore, evident that we need the code to be more reusable in order to decrease the complexity (Selvarani, Nair & Prasad, 2009).

Copyright © 2013 SAVAP International
www.savap.org.pk

ISSN-L: 2223-9553, ISSN: 2223-9944
Vol. 4 No. 1 January 2013

Modularization program style and the object-origin program style have helped a lot in decreasing the complexity of the software. Nevertheless, the model programming and object orientation have not solved the problem forever. The size of programs has led to the complexity to be out of control. For example, the count of code lines of the main products of Microsoft, Microsoft Vista, Microsoft Exchange Server 2007 and Microsoft Office 2007 is around 200,000,000 lines (Yu & Zhou, 2010). In addition, the complexity of building software which is capable to run on multiplatform is an age-long issue in software engineering. The issue comes in terms of design, implementation, and maintenance (Wang, 2006).

Users are interested in having mobile versions for the applications that they use on their desktops. Therefore, companies and developers should be aware that they need to build at least two versions of their applications. One of the versions can be run on a desktop's platform like Windows while the other can be run on mobile phones. That demands time and funds, since the two applications have to be developed from the ground even though they are similar in their features and objectives.

## SOA DECOMPOSITION

In this part, a general architecture of the proposed file transfer is illustrated in Figure 1 below:

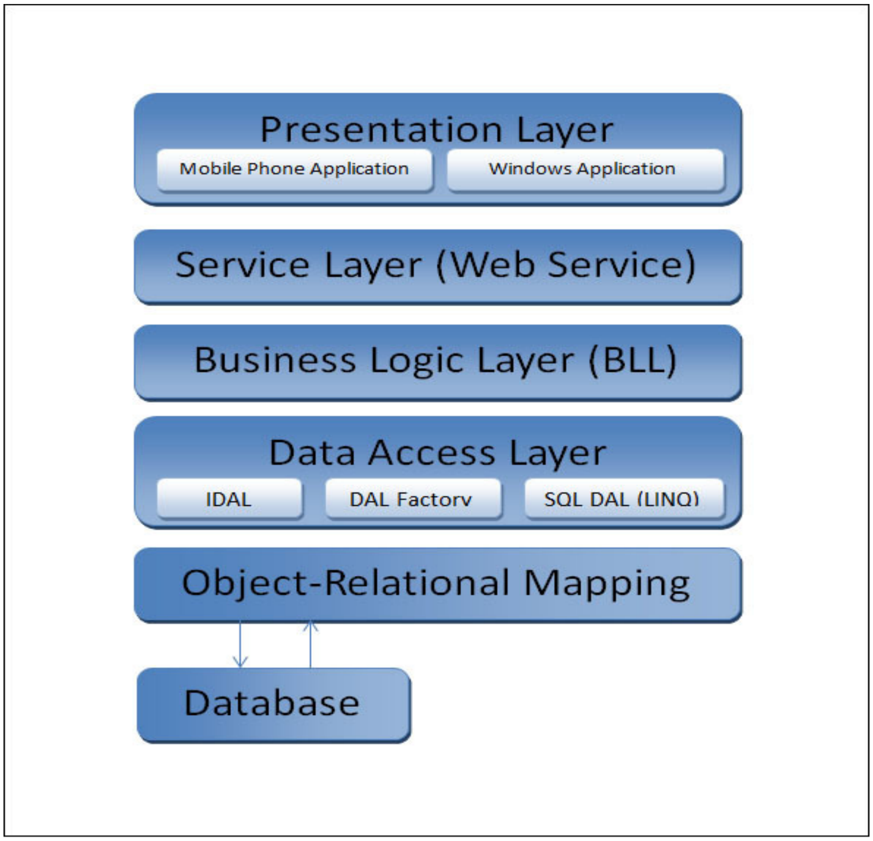

*Figure 1. Architecture of the Proposed Approach.*

## Web Services

Schmelzer et al. (2002) define Web Services as loosely coupled, contracted components that communicate by using Extensible Markup Language (XML) based interfaces. Loosely coupled refers to the dependency in a program or a service so that there is no requirement the

Copyright © 2013 SAVAP International
www.savap.org.pk

service needs to work properly. As a result, Web Services are platform independent. Contracted means what the input is that we have to pass into the service and what output variables are that get as a result on the invocation of a Web Service. Web services are used in building its components so that they cannot be accessed except by their interfaces; the methods which marked as public. As a consequence, the implementation of a component is hidden from outside. This is what it is known by encapsulation.

XML is the output representation of Web Services so it gain all benefits of XML which is a human readable, self-describing, and text-based format that is firewall friendly (Schmelzer et al., 2002). In addition, the type of network does not represent any barrier of using Web Services inasmuch as they are self-contained applications. Consequently, they can be called over any network such as the Internet (Schmelzer et al., 2002).

Web Services are considered as a high approach in distributed systems because they are similar to the concept of the remote procedure call (RPC) architecture. However, Web Services are much more powerful that RPC for the reason that its technology goes beyond what existing RPC architecture can supply (Schmelzer et al., 2002). On the other hand, both remote procedure call (RPC) and Web Services are a solution of building a program which its components are physically distributed across the network. Nevertheless, in term of complexity and costs Web Services provide better solution that RPC. Since RPC architectures (DCOM and CORBA) are too complicated to provide suitable interoperability among different systems (Schmelzer et al., 2002). This is another issue can be addressed by Web Services.

Object-orientation can help in building a less complex system. That is because its concern is about supplying reusability aspect in our complex and costly application so that our systems can be upgraded and reusable. As a result, it solves problems of legacy systems which cannot be developed into newer systems after a while because of their complexity. Consequently, it is useful in term of building less complex systems but not in term of decomposing distributed systems, so we still need the help of Web Services in addition to object orientation.

Moreover, that web services programs which may have developed on different platforms in different languages are able to have communication between each other based on standard way (Ferris & Farrell, 2003). In addition, web services supply reusability so we can say that they are reusable software components over the Internet (Thomas, Thomas & Ghinea, 2003; Emekci, Sahin, Agrawal & Abbadi, 2004). Web services are well suited technology for mobile devices because of its characteristics: heterogeneity, low processing power, low memory (Steele, 2003).

## Linq (Language Integrated Query)

Simply, LINQ introduces a uniform in object-oriented for heterogeneous sources to be accessed easily. In addition, it makes the interaction between object oriented programming and relational database something easy to be handled (Kan & Yujun, 2009).

The main achievement by using LINQ is that making the work with databases much easier than before. Since usually there is a gap between programming languages and databases (Marguerie, Eichert & Wooley, 2008). That is because developers have to work with databases by forming quires and sending them to a database engine to execute some SQL statements, then eventually to get back the result of the requests.

Standard SQL statements are not recognized by programming languages. That means their compilers deal with SQL statements as a series of characters or what is known as strings. Any

Copyright © 2013 SAVAP International
www.savap.org.pk

ISSN-L: 2223-9553, ISSN: 2223-9944
Vol. 4 No. 1 January 2013

syntax mistake or typo will not be recognized by their translators or compilers. As a result, programming languages' environments are not good testers or examiners to detect any errors in syntax at debug time.

Working with data collection objects has become easier in term of developing perspectives. Data collections are more complex structure than basic objects. Working with collections for temporary saving information into and extracting data from them are not an easy task. If we have a collection of data which contains strings, numbers, links, and etcetera, using traditional ways will exhaust developers to manage working with it. As a result, managing data are not easy and they require experience and knowledge to work with them.

LINQ addresses the above discussed problems and issues. Actually, it goes more beyond that in order to solve other problems. One of the great achievements by using LINQ is that developers do not need to worry about a data source they are working with especially in case of databases. That means programmers are write quires by using a special format (LINQ) one time only. In return, LINQ should deal, behind the senses, with the type of the database. That means LINQ will generate the appropriate SQL statements for a certain database.

LINQ technique has introduced by Microsoft as a technique of dealing with different data sources as well as allowing mixing data from different data sources (Marguerie, Eichert & Wooley, 2008). Consequently, less time and effort are required in building systems.

## Tables Mapping And Data Access Layer

Microsoft SQL Server is database management system that is used in the system. There are four tables which are users, clients, files, and messages. Users table has the registered user accounts in purpose of authorization when a user try to login into the system. Clients table stores the information of the devices that belong to a certain user. Files table has all the information of the public and private shared files. Message table which contains the messages that have been sent to a user, then they are removed when they reach their destinations. The database uses File Stream technique which has been introduced in SQL Server 2008 as a new feature. The database of the system has represented in the XML file so that any query is required in the system will be converted from object manner to the SQL statement.

For data access layer it requires three main components namely Interface Data Access Layer (IDAL), DAL and SQL DAL. IDAL represents the interface to data access layer in the system since it contains all the abstract functions of the systems. In other words, all operations of the system have been encapsulated in these components. That means any call comes from above (BBL) should use IDAL since it represents the window of this layer. By having this component, the system isolate the work of data access layer from being called and implemented in BBL, so that any changes might happen to the system. BBL will not be affected.

The system relays on DAL Factory for purpose of initializing the proper object from SQL DAL components by using reflection technique. By using reflection, the system will be able to create instance from the object in SQL DAL at run time. That gives the system the possibility of replacing SQL DAL component at any time in case that there are changes in the database type.

SQL DAL The system uses SQL Server as Database Management System (MDBS). Since the system has deployed the system using Microsoft .NET platform. All ADO.NET usage has encapsulated, so changing MDBS will be very easy. The system just needs to implement the right component that works with the desired database. In addition, all the interfaces in IDAL

Copyright © 2013 SAVAP International

has implemented in SQL DAL so that IDAL can refer to any object in SQL DAL. As a result, IDAL is just as if it were remote control device of SQL DAL. We conclude that any further replacement of SQL DAL, the new component should implements the IDAL interfaces as well, so there are no changes needed to BLL.

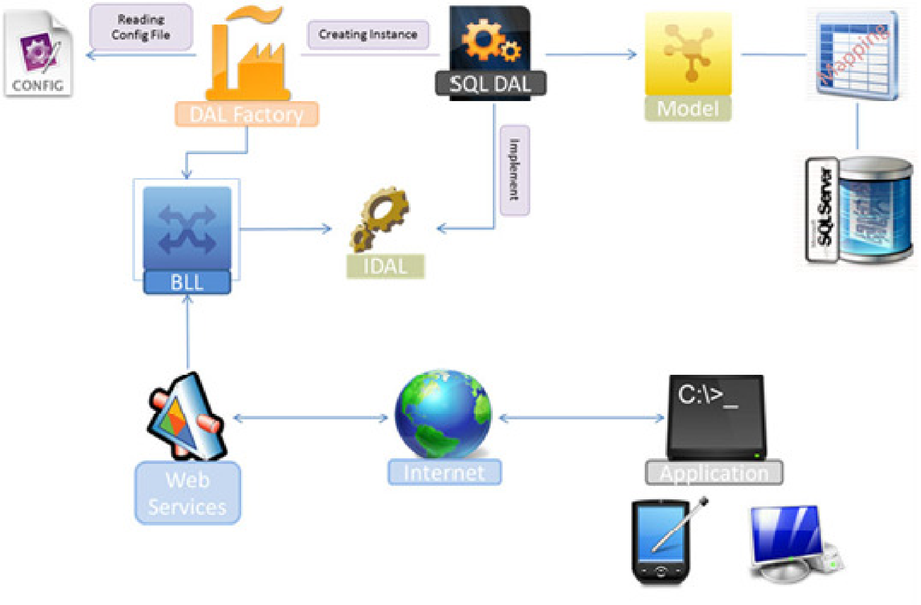

*Figure 2. Overall Proposed Approach at Run-time*

**Error! Reference source not found.**shows how the system work at run-time It starts from application of the client side for either windows or mobile program. After the program has been hit, the program will try to call a web service through the Internet. For example, the end user application wants to start downloading a file, so it will try to invoke the proper web service. After the request reaches the web service which resides on the web server, the web service will call the appropriate function from BLL. BLL at the beginning will ask DAL Factory to create an instance which supports the proper interface from IDAL. As discussed in DAL Factory that its purpose to initialize the object from SQL DAL, which has been stated in XML configuration file, by using reflection. Then, BLL will send a request to the created object by DAL Factory through the interface (IDAL). After SQL DAL has been called, SQL query will be formed by using LINQ and mapping the database into objects. Finally, the formed SQL query will be send into the database engine to be executed.

## RESULT

The files sharing system has tested on both desktop computers and mobile devices. The test has done on different size of files in different size of transferring packets. The size of files is 256 KB, 512 KB, 1 MB, 2 MB, 3 MB, 4 MB, and 5 MB. These files have been uploaded and downloaded with different size of transferring packets which are 32KB, 64KB, and 128KB. These both tests have been done on the same device. Emulator device has been used as a mobile emulator for testing. The desktop device has Core Due CPU P9600 with 2.66 GHz. The RAM is 1.22 GB while the emulator is Windows Mobile 6 Professional Emulator with its standard features. Figure shows the time taken in seconds for uploading the files to the database from a stand-alone computer. Figure 4demonstrated the time spent to download the files from the server into a mobile device.

Copyright © 2013 SAVAP International
www.savap.org.pk

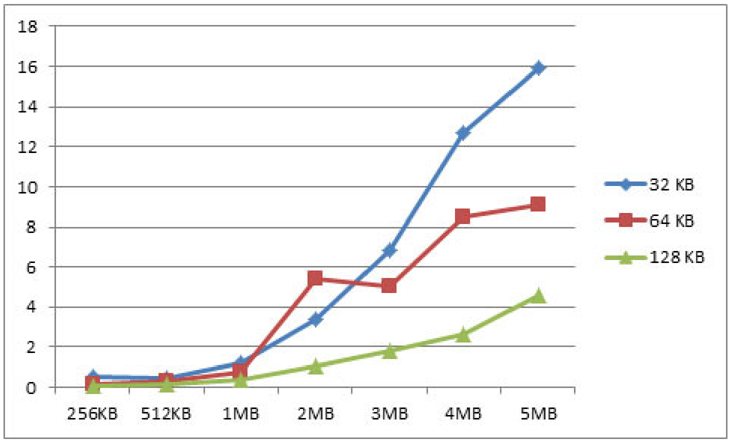

**Figure 3. Time (in sec.) of downloading files from Windows environment to the database.**

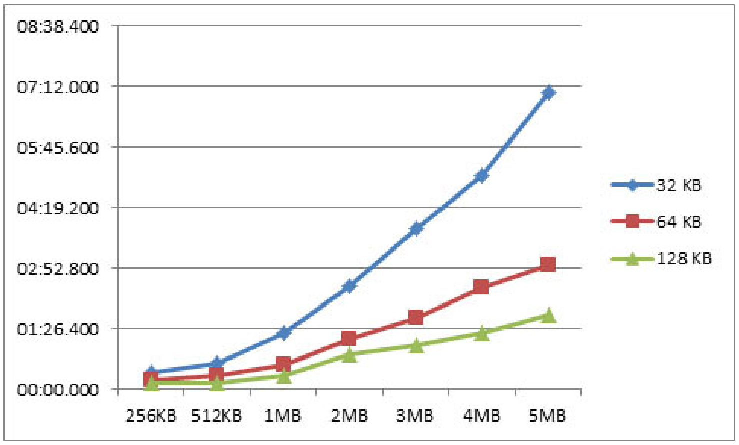

**Figure 4. Time (mm:ss:ms) of downloading files from mobile environment.**

Obviously, Figure 3 and Figure 4 show that mobiles devices consume more time compare to standalone computers.

## CONCLUSION

The proposed approach provides interoperability and decrease the complexity of the system by using SOA and web service. SOA architecture simplifies the architecture of systems. Web services provide interfaces to SOA. In addition, web services rely on XML and HTTP as transport protocol. Using XML and HTTP have the role of decreasing the complexity of building cross-platform systems. Moreover, we have noticed that by increasing the size of transferring packet, the time needed of file transmission becomes less. We also found that a mobile device consumes more time compare to a standalone computer. That is because there are limitations of capabilities of mobile devices which are lack of memory and processing speed comparing to standalone computers.

## REFERENCES

Biström, J. & Partanen, V. J. (2004). Mobile P2P - Creating a mobile file-sharing environment. *Telecommunications Software and Multimedia Laboratory*. Helsinki University of Technology.

Copyright © 2013 SAVAP International

BitTorrent. (2010). BitTorrent. Retrieved June 21, 2010, from: http://www.*bittorrent.com/*

Curbera et al., (2002). Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI. *Internet Computing, IEEE,* 6(2), 86-93.

Emekci et al., (2004). A Peer-to-Peer Framework for Web Service Discovery with Ranking. Web Services, 2004. *IEEE International Conference* (pp. 192-199). IEEE.

Esposito, D. & Saltarello, A. (2008). Architecting Microsoft® *.NET Solutions for the Enterprise.* Washington: Microsoft Press.

Ferris, C. & Farrell, J. (2003). What Are Web Services? *Communications of the ACM,* 46(6), 31.

Gehlen, G. & Pham, L. (2005). Mobile Web services for peer-to-peer applications. *Consumer Communications and Networking Conference, 2005.* CCNC. 2005 Second IEEE, (pp. 427-433).

Hammad-ul-Hasnain. (2005). Building Mobile Application with *.NET Compact Framework. Engineering Sciences and Technology, 2005.* SCONEST 2005. Student Conference, (pp. 1-6). Karachi.

Hassan, M. (2009). Mobile Web Service Provisioning in Peer to Peer Environments. *Service-Oriented Computing and Applications (SOCA), 2009 IEEE International Conference* (pp. 1-4). Taipei: IEEE.

Hofeld, T., Tutschku, K. & Andersen, F. U. (2005). Mapping of file-sharing onto mobile environments enhancement by UMTS. *Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference* (pp. 43-49). IEEE.

Limeware. (2010). Limeware. Retrieved June 21, 2010, from: http://www.limeware*.com/*

Jamil, M., Shaikh, S. P., Shahzad, M. & Awais, Q. (2008). 4G: The Future Mobile Technology. TENCON 2008 - 2008 IEEE Region 10 Conference, (pp. 1 - 6). Hyderabad.

Johnsrud et al., (2008). Efficient Web Services in Mobile Networks. *ECOWS '08. IEEE Sixth European Conference (pp. 197-204).* Dublin: IEEE.

Kan, W. & Yujun, Z. (2009). Using LINQ as an Instructional Bridge Between Object-Oriented and Database Programming. *Proceedings of 2009 4th International Conference on Computer Science & Education,* (pp. 1464-1468). Nanning.

Kim, J. (n.d.). Peer-to-Peer Overlay Networks. Retrieved September 10, 2010, from FSU Computer Science: http://ww2.cs.fsu.edu/~jungkkim/P2P.html

Kim, R. (2006). The world's a cell-phone stage. Retrieved June 21, 2010, from SFGate: http://www.sfgate.com

Lawton, G. (2005). *What Lies Ahead for Cellular Technology?* Computer, 14-17.

Liu, F., Wang, G., Li, L. & Chou, W. (2006). Web Service for Distributed Communication Systems. *Service Operations and Logistics, and Informatics, 2006. SOLI '06. IEEE International Conference (pp. 1030-1035).* Shanghai: IEEE.

Marguerie, F., Eichert, S. & Wooley, J. (2008). *LINQ in Action. Greenwich,* United States of America: Manning Publications Co.

Copyright © 2013 SAVAP International
www.savap.org.pk

ISSN-L: 2223-9553, ISSN: 2223-9944
Vol. 4 No. 1 January 2013

Mehta, V. P. (2008). *Pro LINQ Object Relational Mapping with C# 2008*. United States of America: Apress.

Microsoft. (n.d.). FILESTREAM Overview. Retrieved October 1, 2010, from TechNet: http://technet.microsoft.com/en-us/library/bb933993.aspx

MSDN. (2010, August 4). Device Emulator for Windows Mobile. Retrieved October 1, 2010, from MSDN: http://msdn.microsoft.com/en-us/library/bb158519.aspx

Neable, C. (2002). *The .NET Compact Framework*. Pervasive Computing, IEEE, 84-87.

Newcomer, E., & Lomow, G. (2004). *Understanding SOA with Web Services.* United States of America: Addison Wesley Professional.

Palazzi, C. E., Bujari, A. & Cervi, E. (2009). P2P file sharing on mobile phones: Design and implementation of a prototype. *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference (pp. 136-140).* Beijing: IEEE.

Schmelzer, R., Vandersypen, T., Bloomberg, J., Siddalingaiah, M., Hunting, S. & Qualls, M., et al. (2002). *XML and Web Services Unleashed (1st ed.).* United States of America: SAMS Publishing.

Schmidt, A., Stuhr, T. & Gellersen, H. (2001). Context-Phonebook - Extending Mobile Phone Applications with Context. *In 3rd Mobile Human-Computer Interaction Workshop,* (pp. 1-6).

Selvarani, R., Nair, T. & Prasad, V. (2009). Estimation of Defect proneness Using Design complexity Measurements in Object- Oriented Software. *2009 International Conference on Signal Processing Systems,* (pp. 766-770). Singapore.

Steele, R. (2003). A Web Services-based system for ad-hoc mobile application integration. *Proceedings of the International Conference on Information Technology: Computers and Communications (ITCC'03)* (pp. 248-252). IEEE.

Thomas, J. P., Thomas, M. & Ghinea, G. (2003). Modeling of Web Services Flow. E-Commerce, 2003. *CEC 2003. IEEE International Conference (pp. 391-398).* IEEE.

Ventä, L., Isomursu, M., Ahtinen, A. & Ramiah, S. (2008). "My phone is a part of my soul" – How People Bond with Their Mobile Phones. *The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (pp. 311-317).* Valencia: IEEE.

Wang, Y. (2006). Cognitive Complexity of Software and its Measurement. *Cognitive Informatics, 2006. ICCI 2006. 5th IEEE International Conference (pp. 226-235).* Beijing: IEEE.

Yang, C. T., Chen, C. J., Chen, H. Y. & Hsu, C. H. (2008). A Peer-to-Peer File Resource Sharing System for Mobile Devices. *Grid and Pervasive Computing Workshops, 2008. GPC Workshops '08. The 3rd International Conference, (pp. 275-280).* Kunming.

Yu, S. & Zhou, S. (2010). A Survey on Metric of Software Complexity. *Information Management and Engineering (ICIME), 2010 The 2nd IEEE International Conference (pp. 352-356).* Chengdu: IEEE.

Copyright © 2013 SAVAP International