Ubiquitous Metrices for Multi-platform Ubiquitous Applications

Anil Kumar Mishra, Yogita Gigras, Latika Singh

Abstract: The world has shifted from the world of content to the world of context. Instead of software, focus is more on service implementation now. As ubiquitous computing is latest trend in modern computing, all the current software will shift to ubiquitous computing environment in coming few years. Ubiquitous computing is taking a new place in everyone's life. In this paper, we will describe the general level framework on the basis of which we can define the ubiquity of any software. We will also suggest the suitable metrics for any software project to be a part of ubiquitous computing environment. We have done case study on a live project Xapper D.

Keywords: Framework, Ubiquitous computing, Ubiquitous application.

I. INTRODUCTION

Software engineering is on the matured phase now. Even pervasive and mobile computing technologies have become old. Latest trend in software engineering is ubiquitous computing. It would soon provide an environment where devices are seamlessly integrated into the environment and provide simple interfaces for human interaction to perform everyday life activities [1]. It has been identified by the researchers that, in future, the world will be characterized by computing and sensing elements embedded invisibly in the environment. The integrated devices would have the ability to interact with users to change environmental behavior according to expressions and feelings rather than forcing the user to search interfaces to perform required operations. Thus we can say that almost anything and everything we could think of (homes, shops, hospitals, roads, table, chair, cloths, etc.) will have some amount of computing capabilities. The elements of ubiquity offer opportunities for new kind of software applications [1]. It is not possible to introduce a stand-alone ubiquitous computing application or solution. The application must be able to integrate, interoperate and incorporate with the existing services and infrastructure in a seamless manner.

To make this a reality, enormous software challenges are being faced by the developers. The basic features of ubiquitous computing like context awareness, invisibility, mobility, privacy, security etc. are major software engineering challenges in moving further towards ubiquitous computing based applications development[2].

In this paper, we will describe the general level framework (characteristics) on the basis of which we can define the ubiquity of any software and also suggest metrics to improve the ubiquity of the software.

We have done a case study on a project Xapper D, which is capable of fitting in ubiquitous computing environment. For this we consulted various people; the project manager, the team lead and software developers; involved in the project. Our case study involved

- Identification of ubiquitous computing characteristics.
- Matching the characteristic with that of the software selected.
- Evaluation of the software based on those characteristics.

The rest of the paper has the following structure: section 2 describes XapperD and section 3 defines a framework for evaluating ubiquitous computing software and also describes the ubiquity level of XapperD

II. DESCRIPTION OF XAPPER D

XapperD is a multi-platform ubiquitous application that allows the business executive to capture physical application forms, PODs, receipts along with the key metadata and initiate the business process instantly. The primary challenge faced by field executives in delivering effective customer service is getting paper from the point of initiation to the point of processing centre. Typically the information gets captured on a physical application form. It then gets submitted to a branch from where it gets couriered to the processing center and then finally it gets processed. Improper filling, missing supporting documents and loss in transition add to the turn-around-time for the process. Capturing information at point of customer contact and initiating its processing in real time is a challenge both in terms of time and cost. This ubiquitous application bridges the gap between service delivery timelines and customer expectations. XapperD is used to capture documents and images at the point of customer contact and initiate its processing in real time at the back office. The tangible benefit of this ubiquitous application has already found instant applicability in Banking, Insurance, Telecom, Transportation and Logistics verticals.

XapperD directly runs on android or iOS mobile or tablet device. It has two components:

XapperD Mobile Application:

It is used to capture forms/documents/images and it runs on any android or iOS device. Application form can be configured based on the requirement. Complete application form can also be designed in XapperD, so that the user is able to enter complete form in the tablet itself. The form can be made intelligent with built-in checks and validations to increase first time right percentage.

Revised Manuscript Received on June 8, 2019

Anil Kumar Mishra, Computer Science and Engineering Department, THE NORTHCAP University, Gurgaon, India.

Yogita Gigras, Computer Science and Engineering, THE NORTHCAP University, Gurgaon, India.

Latika Singh, School of Engineering and Technology, Ansal University, Gurgaon, India.

XapperD Mobile Capture Server:
The documents and images are sent to the XapperD mobile capture server after being captured through XapperD mobile application. It is a multi-tiered platform independent solution built using robust server-side Java and J2EE technologies. Advanced image processing is done at XapperD mobile capture server for creating process ready documents. It can be integrated with any back office systems like DMS, BPM, LOBs and Core Applications.
XapperD mobile capture server can either be in premise or cloud based.

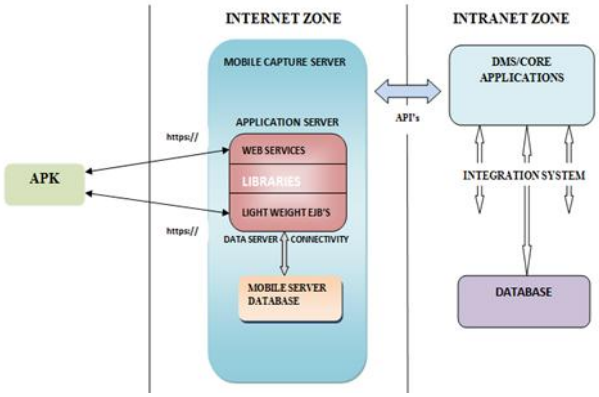

Figure1. XapperD Architecture

III. UBIQUITOUS COMPUTING CHARACTERICS

For any software to fit in ubiquitous environment, it must satisfy certain characteristics. In general there are 10 characteristics that must be present in any ubiquitous software. For software to be completely ubiquitous, all these must be fully implemented. However, we can also have ubiquitous software projects with different levels of adherence to the ubiquitous characteristics. These different levels of adherence can be a consequence of the application domain and project's requirements. Also the different levels of adherence could be a consequence of the absence of some ubiquitous characteristic on the software project. This way, it is possible to have software project with different levels of ubiquity [4].
Considering each of the ubiquitous characteristic, we will now find the level of ubiquity present in XapperD. Following are the characteristics:

1. **Service omnipresence (SO):** It allow users to move around with the sensation of carrying computing services with them. A higher number is a good indicator [5].
   XapperD is deployed on smart phones and tabs that fulfill some basic requirements. These are the objects of daily use and we no longer think them of as a computing device. They can be carried by us anywhere and everywhere.
   Hence, we can quantify service omnipresence as:
   Service omnipresence =
   OS on which software is configured / OS on which software can be configured
   = 9/5
   = 0.56
   Numerator = {android or iOS based mobile or tablet or iPad}
   Denominator = {android or iOS based mobile or tablet or iPad, Windows 8, Blackberry mobiles}
   Service omnipresence will be increased in future.
2. **Invisibility (IN):** Ability of computation service being present in objects of daily use, weakening the sensation of explicit use of a computer and enhancing the perception that objects or devices provide services or some kind of "intelligence". A higher number is a good indicator.
   XapperD is deployed in mobiles and tabs which are the objects of daily use.
   The users of these applications do not have to carry any extra computational device. Hence, we can say that the software is invisible. We can quantify invisibility as:
   Invisibility = 1
3. **3. Context sensitivity (CS):** Ability to collect information from the environment where it is being used [10]. A higher number is a good indicator.
   Hence, we can quantify context sensitivity as:
   Context Sensitivity =
   Context sensitivity present/ overall context sensitivity
   = 3/6
   = 0.50
   Numerator = {capture location, thumb impression, signature}
   Denominator = {capture location, thumb impression, signature, face recognition, sensors for sensing users' surrounding environment}
   Context sensitivity will be increased in future.
4. **Adaptable behavior (AB):** Ability of dynamically adapting to offered services according to the environment where it is being used, respecting the limitations of the environment [5]. A higher number is a good indicator.
   Hence, we can quantify adaptable behavior as:
   Adaptable Behavior =
   Services currently adaptable/services which can be adapted
   = 4/5
   = 0.80
   Numerator = {Wi-Fi printers, GPRS, 2G, 3G}
   Denominator = {Wi-Fi printers, GPRS, 2G, 3G, 4G}
   Adaptability needs to be improved.
5. **Experiences capture (EC):** Ability of capturing and registering experiences for later use.
   Experience capture is a continuous process. By capturing the experiences of the team members and the users of the application, the application has regularly been updated with new features so as to make it more ubiquitous [5]. The project has matured enough and has reached this stage through experience capture.
   Scope to capture experience is necessary.
6. **Service discovery (SD):** Pro-active discovery of services according to the environment where it is being used. In order to find new services or information, the application has to interact with

environment and allow user to do the same, to achieve some desired target.

Service discovery can be discovering another device in the environment to interact with or another software service for the users in that environment [6].

Hence, we can quantify service discovery as:

Service Discovery =

Services in use/ total services discovered

= 2/3

= 0.67

Numerator = {Banking, Insurance}

Denominator = {Banking, Insurance Transportation and Logistics}

Aim is to discover more services.

7. **Function composition (FC):** Based on basic services, the ability to create a service required by the user [6]. A higher number is a good indicator.

   Hence, we can quantify function composition as:

   Function Composition =

   Services available to user/total services which can be made available

   = 1/3

   = 0.33

   Numerator = {finance sector}

   Denominator = {finance sector, hosting Business Use Cases online}

   More business use cases need to be planned and implemented to increase function composition.

8. **Spontaneous interoperability (SI):** Ability to change partners during its operation and according to its movement [7].

9. **Heterogeneity of devices (HD):** Provides application mobility among heterogeneous devices. That is, the application could migrate among devices and adjust itself to each of them.

   The different parts of the application i.e. mobile capture server, mobile application and the integration adapter, runs on devices with highly varying specification.

   Tested devices =

   Tested true/total devices tested

   Supported devices =

   Supported/total devices tested true

   For **Mobile Capture Server**

   Tested devices = 11/28 = 0.39

   Supported devices = 15/16 = 0.94

   For **Integration Adapter**

   Tested devices = 15/31 = 0.48

   Supported devices = 19/20 = 0.95

   For **Mobile capture client**

   Tested devices = 4/4 = 1

   Supported devices = 4/4 = 1

   Therefore we can say that the application is heterogeneous.

10. **Fault tolerance (FT):** Ability to adapt itself when facing environment's faults (for example, on-line/off-line availability). A higher number is a good indicator.

XapperD Mobile Application captures images which are then transferred to XapperD Mobile Capture Server. We can upload one application at a time or bulk of applications can also be uploaded. In case of network outage, upload resumes from the last place. The software can also be used in no network coverage areas.

Hence we can say that the application is fault tolerant. Fault tolerance should always be high.

Also in any ubiquitous environment, privacy of users' content is of utmost importance. With mobility and context sensing, we have a great threat to security of users' information.

11. **Privacy (PR) -** Availability of users' information to other user or to third party [9]. A high number is a good indicator.

XapperD is a highly secure application. Only registered devices (Mobile/Tablet) are allowed to communicate with the XapperD Mobile Capture Server ensuring safety and security of customer information. The IMEI/UDID number of any device (tablet/mobile/iPad/) is registered first before using the application. This is to prevent any unauthorized access of the application. User credentials are associated with the device id to prevent unauthorized access of the application i.e., login to the device is authenticated. We also have retention policy definition for data stored on devices (For e.g. automatically delete device data after successful upload or delete device data after acknowledgement from centralized team etc). Metadata and images are encrypted (DES Algorithm) before transferring over the network. The images cannot be opened directly from the file system on the mobile device and no data is accessible outside the application. Even transferring data to a laptop will not work. Data is transferred to the network using secure https protocol. Audit trails are maintained for successful upload cases to create transparency and accountability in the system.

Hence, we can quantify privacy as:

Privacy (Security) = 7/7

= 1

IV. RESULT AND ANALYSIS

A. Figures and Tables

Applied Metrics

Metrics	Values
SO	0.56
IN	1
CS	0.50
AB	0.80
EC	Present
SD	0.67
FC	0.33

SI	
HD	HD is present in MCS, IA and MCC values for which for tested and supported devices are 0.39, 0.94; 0.48, 0.95; 1,1.
FT	High
PR	1

Note: Ubiquity measure is done on a scale of 0 to 1.

From above results the ubiquity level of the software application can be clearly seen on different ubiquitous characteristics. It can be clearly known about the pervasiveness of the application

V. CONCLUSION

Ubiquitous computing is the latest trend in software computing which involves advancement in Mobile computing and Pervasive computing to present a global computing environment. This environment will combine processors and sensors with network technologies (wireless and otherwise) and intelligent software to create an immersive environment to improve life. But lack of appropriate software engineering approaches is the major obstacle in creating such environments. In this paper, we first identified all the potential software engineering challenges in the ubiquitous computing era and then proposed a general level framework for all the ubiquitous computing applications. This framework will help us to evaluate ubiquity of any software and also shift them to the ubiquitous computing environment. We evaluated software XapperD using this framework to find out the level of ubiquity of that software.

FUTURE WORK

The software today follows Service Oriented Architecture (SOA). They are implemented Software as services (SaaS). We need to have refined characteristics to evaluate the ubiquitous computing software. As developers are still facing a lot of problems in making software for ubiquitous computing environment, a lot of research is required in this area. The future work in this area will be to find new ubiquitous computing characteristics and fine-tune the existing ubiquitous computing characteristics as the next era is ubiquitous computing. Also there is a need of quality model to evaluate ubiquitous computing software so as to make it fit in ubiquitous computing environment.

REFERENCES

1. Ema Kusen, Mark Strembeek, "Security Related Research in ubiquitous computing-Results of a Systematic Literature Review "June 2016, International Journal of Pervasive Computing and Communications.
2. Muhammad Usman Ashraf, Naveed Ahmed Khan, "Software Engineering Challenges for Ubiquitous Computing in Various Applications", 2013, 11th International Conference on Frontiers of Information Technology.
3. Carlos Machado, Eduardo Silva, Thais Batista, Jair Leite, Elisa Yumi Nakagawa, "Architectural Elements of Ubiquitous Systems: A Systematic Review", 2013: The Eighth International Conference on Software Engineering Advances.
4. Dania Abed Rabbou, Abderrahmen Mtibaa, Khaled A. Harras, "SCOUT: Social Context-Aware Ubiquitous System", 2012 IEEE.
5. Ricardo Mejia-Gutierrez, Gilberto Osorio-Gomez, David Rios-Zapta, Damiel Zuluaja-Holguin"Ubiquitous Conceptual design of a ubiquitous application: A Textile SME Case Study for Real Time Manufacturing, Monitoring "Design Engineering Research Group (GRD), University, EAFIT, Columbia, 2011.
6. Rodrigo O. Spínola, Jobson Massollar, Guilherme H. Travassos, "Checklist to Characterize Ubiquitous Software Projects", SBES 2007 XXI Simpósio Brasileiro de Engenharia de Software.
7. 7.Jean Scholtz and Sunny Consolvo, "Towards a Framework for Evaluating Ubiquitous Computing Applications", Published by the IEEE CS and IEEE ComSoc, 2004.
8. Brad Johanson, Armando Fox, Terry Winograd, "The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms", Pervasive Computing Magazine Special Issue on Systems, Version #2, 2002.
9. 9.T. Kindberg, A. Fox, "System Software for Ubiquitous Computing", Pervasive Computing, IEEE, Volume 1, Issue 1, 2002, pg 71-80.
10. Manuel Roman, Christopher Hess, Roy Campbell, "Building Applications for Ubiquitous Computing Environments", First International Conference, Pervasive 2002 Zurich, Switzerland, August 26–28, 2002 Proceedings, Volume 2414, 2002, pg 16-29.
11. M. Satyanarayanam, "Pervasive Computing: Vision and Challenges", IEEE Personal Communications, 2001.
12. Friedemann Mattern, "The Vision and Technical Foundation of Ubiquitous Computing", UPGRADE Vol II, No.5, October 2001.

AUTHORS PROFILE

Anil Kumar Mishra is Ph.D. Scholar in THE NORTHCAP UNIVERSITY in the department of Computer Science and Engineering. His research area is ubiquitous computing and software metrices.He has published many research papers in above areas. He has done B-Tech in computer science and M-Tech in software Engineering.

Yogita Gigras is currently working as Assistant Professor in the Department of CSE & IT, School of Engineering & Technology of The North Cap University, Gurugram, Haryana. She is a committed researcher in the field of Soft Computing and has completed her PhD in the same area. She has done her M.Tech in Computer Science and Engineering from Banasthali University, Rajasthan She has published various papers in peer reviewed reputed International Journals and in IEEE/Springer International Conferences. She is the reviewer and assistant editor of various international journals.

Dr Latika is a Professor and Dean SET at Ansal University. She has done her PhD from National Brain Research Centre, Manesar.
Her research interests span using computation and machine learning tools in various applications including diagnosis of neurodevelopmental disorders, detection of android malwares, blind steganalysis etc. She has published more than 30 papers in various Journal and conferences of international repute.