HiLCoE Journal of Computer Science and Technology

Volume 1, Number 1, December 2012



HiLCoE

School of Computer Science and Technology

Editor-in-Chief

Mulugeta Libsie, Department of Computer Science, Addis Ababa University
E-mail: mulugeta.libsie@aau.edu.et

Editorial Board

Ahmed Hussien, **HiLCoE** School of Computer Science and Technology
E-mail: ahuseid@yahoo.com; hilcoe@ethionet.et

Mesfin Kifle, Department of Computer Science, Addis Ababa University
E-mail: kiflemestir95@gmail.com

Nassir Dino, **HiLCoE** School of Computer Science and Technology
E-mail: nassir@hilcoe.com.et

Tibebe Beshah, School of Information Science, Addis Ababa University
E-mail: tibebe.beshah@gmail.com

Copyright©2013 HiLCoE, Addis Ababa. All rights reserved. This Journal, or any part thereof, may not be reproduced in any manner without the written permission of HiLCoE.

HiLCoE Journal of Computer Science and Technology is planned to be published twice a year by HiLCoE. Individual articles can be accessed and downloaded at http://www.hilcoe.com.et.

All articles published in the *Journal* represent the opinion of the authors and do not necessarily reflect the official policy of HiLCoE, the Editorial Board or the institution with which the author is affiliated, unless this is clearly specified.

Address all correspondences to: *HiLCoE Journal of Computer Science and Technology*, P. O. Box 33465, Addis Ababa, Ethiopia; e-mail: hilcoe@ethionet.com.et; Tel. +251 114162121/ +251 114 673585; Fax +251 114162474.

HiLCoE, one of the first higher education institutions in Ethiopia, was established in July 1997. It is a specialized Computer Science and Technology College that adheres to computing and quality of education. It is now widely recognized that HiLCoE prepares students to become industry and academic leaders who can apply technology and computer science principles across a wide variety of fields.

HiLCoE, since its inception has a vision of being Center of Excellence in ICT Education, Research and Development (ER&D). To realize this vision, in addition to its role in academia, it offers cutting-edge consultancy in the areas of Information Systems Security, Enterprise Architecture, Strategic and Solution Architecture, Service Engineering, Software Development, Project Management, Distributed Systems and Mobile Applications and many more.

As a result, and particularly to encourage prospective graduates to become part of the research community in computing, this Journal is launched. HiLCoE Journal of Computer Science and Technology is a journal for the development of the theory and practice of computer science and technology. Its primary objectives are the following:

- To serve as a forum for the advancement of computer science and technology.
- To contribute to a rapid information and knowledge exchange between researchers in computer science and technology.
- To serve as a forum for postgraduate students in computer science and technology to publish their thesis/project results.

Papers for publication are also welcome in future issues from researchers regardless of institutional affiliation. Future publications will be peer-reviewed by at least two reviewers who are active researchers in the respective area of each topic of the paper unlike the first two issues which are neither peer-reviewed nor the responsibility of the affiliated institutions.

Contents

Interoperable Payment Systems: Requirements Driven Architecture for Ethiopian Banking Sector
Abebe Girmay and Fantahun Abera....1

Enabling Seamless Sharing of Data among Organizations Using the DaaS Model in a Cloud
Addis Mulugeta and Abrehet Mohammed Omer....7

Unit Selection Based Text-to-Speech Synthesizer for Tigrinya Language
Agazi Kiflu and Tibebe Beshah....13

ERP Implementation Framework: The Case of Ethiopia
Alemu Samuel, Mesfin Belachew, and Mesfin Kifle....22

Domain Knowledge Based Requirements Change Management Framework: Case of Airlines Systems
Almaz Tsegaye and Mesfin Kifle....31

Achieving Organizational Interoperability for Agricultural Research with Cross Organizational Business Process Modeling
Amanuel Tekalegn and Fantahun Abera....37

Mobile Banking Framework in the Ethiopian Context
Ayenew Wudu and Fekade Getahun....45

Sensor Based Context-Aware System Architecture for Improving Yields in Agriculture
Bereket Assefa Hailemariam and Dejene Ejigu....51

Web Based Cultural Heritage Inventory System
Dereje Tadesse and Fikreyohannes Lemma....57

Wave Process Model for Open Source Software Development
Ermias Kebede and Oliver Petzoldt....62

Semantic Web-Based Digital Business Ecosystem: The Case of Amharic Semantic Blogging
Eyoss Girma and Workshet Lamenew....68

Data Exchange Using Web Services: The Cases of Ethiopian Revenue and Customs Authority and Transport Authority
Firesew Mikreyohannes and Abrehet Mohammed Omer....75

A Framework for Integrating Software Usability into Software Development Process
Hayat Dino and Rahel Bekele....81

Test Case Generation from UML Activity Diagram: The User Perspective Approach
Henok Bekele and Mesfin Kifle....88

An MVC Approach to Design Content Management System
Kibreab Gizatu and Nassir Dino....94

IPv6 Migration Framework – Case of Institutions in Ethiopia
Kidist Mekonnen and Taye Abdulkadir....99

Quantitative Usability Measurement for Commercial Off-the-Shelf Products: the case of Core-Banking System
Lemlem Berhanu and Rahel Bekele....104

A Methodology for Implementation of Function Dimension in Zachman Framework
Meron Bekele and Tibebe Beshah....109

Automated Heuristic Evaluation for Usability of Multiple Web Interface
Mihirtab Zewdu and Dagmawi Lemma....113

A Web Based School Networking System with ICEfaces MVC in the Ethiopian Context
Pineal Terefe and Fikreyohannes Lemma 120

SOA Business Process Interoperability with TOGAF - The Case of Microfinance Customers
Rahma Hussein and Fantahun Abera 126

Reengineering Open Source CMS using Service-Orientation: The Case of Joomla
Tagel Gutema and Dagmawi Lemma 132

Usable User Identification Technique - The Case of European Union Remote Web Access
Temechu Girma and Dagmawi Lemma 138

Quantification of Software Requirements Specification and Software Quality: Quantitative Approach
Wesen Tegegne, Mesfin Belachew, and Mesfin Kifle 144

Semantic Knowledge Oriented Electronic Health Record Integration System
Winta Tesfaye and Fekade Getahun 151

Integrated Approach for Software Project Management Methodology Selection
Wondwossen Elssa and Fekade Getahun 156

Interoperable Payment Systems: Requirements Driven Architecture for Ethiopian Banking Sector

Abebe Girmay
Dashen Bank S. Co., Addis Ababa, Ethiopia
abebegirmay@yahoo.com

Fantahun Abera
PhD Candidate, Addis Ababa University, Ethiopia
fanetsa8@yahoo.com

Abstract

Modern Payment Systems have numerous advantages over traditional retail payment systems. At present it is common to use cards on banks' ATM and POS terminals as well as other terminals deployed within and outside the country. Card based payment systems in Ethiopian banks are not interoperable. Banks only offer services for cardholders opened accounts and registered businesses in each respective bank branch. This limitation reduces the payment card system utilization and accessibilities which hinder the financial transformation of the country in moving forward from cash based to cashless economy. It also creates duplication of investment across the country with unilateral deployment of card payment systems, ATMs and POS terminals in one place instead of scattering them for ease of use and convenience of the cardholder.

Using the existing international available industry models for interoperability in its present condition has its own problems, caused by highly regulated banking practice in Ethiopia. Adapting the international practice without any local standard affects the business in terms of controlling and operational business setup. Most available models are not bank-led models; and mixed in nature comprised of mobile operator, third party processors, international, regional as well as national banks, which are difficult to implement on the existing infrastructure and environment. Payment system practice and technology implementations outdo long progression on what it reached today; hence, adapting the model as-is for existing newborn environment without the internal background assessment may affect existing operational business implementations.

To alleviate the existing issues, this paper proposed a new architectural model that supports interoperability payment card system in relation to the country's current organizational and environmental uniqueness via the method of literature and document review, questionnaire and interview with technical and operational personnel as well as cardholders to get the current practice and limitations on the existing card based payment systems.

Keywords: Interoperability; Payment System; Card-Based Payment System; ATM; POS

1. Introduction

Payment system can be described as the systems and communications mechanisms put in place by commercial banks to provide their customers with the facilities and alternative channels to effect payments without the limitation of time and convenience [1]. These networks include Automated Teller Machines (ATMs), Point of Sale terminals (POS), Internet and online banking facilities, branch networks, and payment instruments.

Card Based Payment System is one part of payment instruments and has been widely used in developed countries for long time since a bankcard was introduced as a payment mechanism and is rapidly expanding in developing countries with alternate payment channels [1].

According to the Institute of Electrical and Electronics Engineers (IEEE) [2], interoperability is the ability of two or more systems or components to exchange information and to use the information that has been exchanged. The term is often used in technical systems, or alternatively in a broad sense, taking into account social, political, and organizational factors that impact system performance.

Interoperability can be classified in to four layers on implementations; these are [3]; Organizational, Semantic, Syntactic and Technical layer Interoperability. Since this research work emphasized on the Organizational and Semantic Interoperability

for the existing Payment Card System, the definition and meaning of these two interoperability layers is discussed in brief.

Organizational Interoperability: is related to whether the participating organizations use the shared information in similar contexts of the business functions, services or business process, i.e., the shared information that impact the business activities related with business services, legal, political, and process interoperability are to be considered.

Semantic Interoperability: is related to the meaning of the shared information. It is the ability to automatically interpret the information exchanged meaningfully and accurately in order to produce useful results as defined by the end users of both systems. To achieve semantic interoperability, both sides must refer to a common information exchange reference model. The content of the information exchange requests are unambiguously defined: what is sent is the same as what is understood.

2. Background

The current payment infrastructure is characterized by heavy use of cash for all types of payments. Although banks are extending their branch networks rapidly and the emergence of new banks is accelerated since the year 2000, access to banking service in the country is still limited. According to the National Bank of Ethiopia, in 2008, 4.3% of the population or around 20% of households have opened bank accounts. Based on this report, for example, in 2008 there was only one bank-branch for every 140,000 inhabitants, indicating that Ethiopia still remains one of the most under-banked countries in the world even by Sub-Saharan standards [4].

As a result of the rapid initiation towards the Growth and Transformation Plan and recent directives that mandated to have core-banking and modern payment systems by the National Bank of Ethiopia, few financial Institutions started to provide alternative channels to their customers in facilitating payments. Recently banks in Ethiopia started alternative channels on card based payment system and deployed systems to enable issuing and acquiring of card based transactions. However, several issues arise around interoperability and resource sharing. Moreover, electronic payment systems are at an embryonic stage. Presently Dashen, Zemen, Wogagen, and CBE have started introducing card based payment systems independently and other banks are planning to invest in such technology. However, since the cardholder can't use other banks' networks other than the issuing bank, the current implementations on payment systems are not interoperable.

This interoperability payment process problem is mainly related to Organizational Interoperability layer as a result of lack of procedures and legal interpretation to be adopted in the processing of multiple bank payment system. Consequently a card issued to one banks' customer can't be used on other banks' ATM or POS. There is no sharing of information on the existing terminals and payment systems in similar contexts of their business functionalities, or business processes. Therefore, banks and the nation in general are wasting a massive amount of money and resources purchasing separate technologies which will serve only for the limited customer base of particular financial institutions.

The reflection of this problem is also observed on Semantic Interoperability to interpret the meaning of the transactions request from the terminals that leads to decline the transactions while attempt by the cardholder other than their banks' channel. Thus a customer more than one account with various banks will have different cards issued and linked to its account based on each bank's interest which will create technological duplication as well as inefficient functionality to the consumers in using his/her cards for payment.

Therefore, this paper attempts to study the organizational and semantic interoperability problems on the payment process and infrastructures based on requirements from the owners in the Ethiopian banking sector and workout interoperability architectural model to fill the gap that will save cost in relation to implementation time and resources.

3. The Proposed Solution

In terms of the available and discussed models (business and technical), operation scope, pros and

cons explored with all possibilities and alternative solutions, the closed national switch, with the establishment of a common company to run the business of the central platform is suitable for Ethiopia. This is because of the limitations associated with both local and foreign outsourcing on open with outsourced Processing. Despite their advantages of easy implementation, the outsourcing options will not give the required level of control and flexibility on the operation of the business. Moreover, gradual learning and skill development is more attainable with the option of establishing a common company on the closed national switch, which is critical considering the sustainability of the scheme. The company can communicate with all banks' core-banking systems in the case of no EFTSwitch implementations and connect to banks with EFTSwitch that have already deployed payment card systems, and act as a third party issuer for card issuing and transactions routing with terminal deployer alternatives for the later.

The interoperability model discussed above shall be designed in such a way that it handles the interests of all banks that are on different levels in terms of card payment operations. Banks that started card payment with their own switch can easily interface to the common Switch. Banks without an EFTSwitch can use this common Switch as their switch to drive their ATMs/POSs. Banks that want to work only on issuing cards can again use the common Switch to personalize their card and allow their customers to use ATMs/POSs of other participant banks.

Based on the above rationale, limitations on the exiting country-specific interoperability models, output of the methodology and recommended a layered and functional architecture of Figure 1 as a contribution for the Ethiopian Banking sector on how they can interoperate and integrate their payment systems as per the proposed modalities. The basic concept of the Layered Architecture will give solution on how member banks implement interoperability with particular emphasis on Integration and Business layers.

- The authorization EFTSwitch and the card and business management interface component interact with international card schemes for not-on-us transactions.
- All the communication between the interface and Business Logic Layer network lies in the integration layer.
- The business Layer comprises the business logic of the different interactions between member banks for billing, ATM and POS management, transactions and reconciliations, card management, and loyalty components.
- The Data layer is the last layer for which all the transactions stored and retrieved for settlement, reconciliation as well as dispute management functionalities.

The high level architectural diagram of Figure 2 reveals the logical network connectivity between participating Bank's Switch or ATMs/POSs, international card scheme, TPP processor and ACH for clearing settlement purposes. This architectural diagram is different from others; as the result of the connectivity taken into consideration based on the existing Ethiopian banking payment system, organizational and operational scenarios.

4. Discussion

4.1 Response to initial problem

The initial problem of the statement of the research aimed to address the issue of the resource sharing and interoperability of terminals on ATMs as well as POS terminals for which the cardholder of one bank can be able to use the terminals of others. Since the proposed system uses as a common EFTSwitch, it can solve the issue of interoperability and resource sharing using the switching and routing of transactions initiated from one bank terminals to the base systems and/or terminals.

On the other hand, a customer who has more than one account maintained with various banks can use his/her accounts linked on card at any terminal for transaction mandated by the proposed common switch. This shall create efficiency in terms of functionality. Moreover, it can be able to work in two technical modalities; for those who have already EFTSwitch, and with Core-banking system that haven't EFTSwitch to eliminate the duplication of resource and maintain resource efficiency.

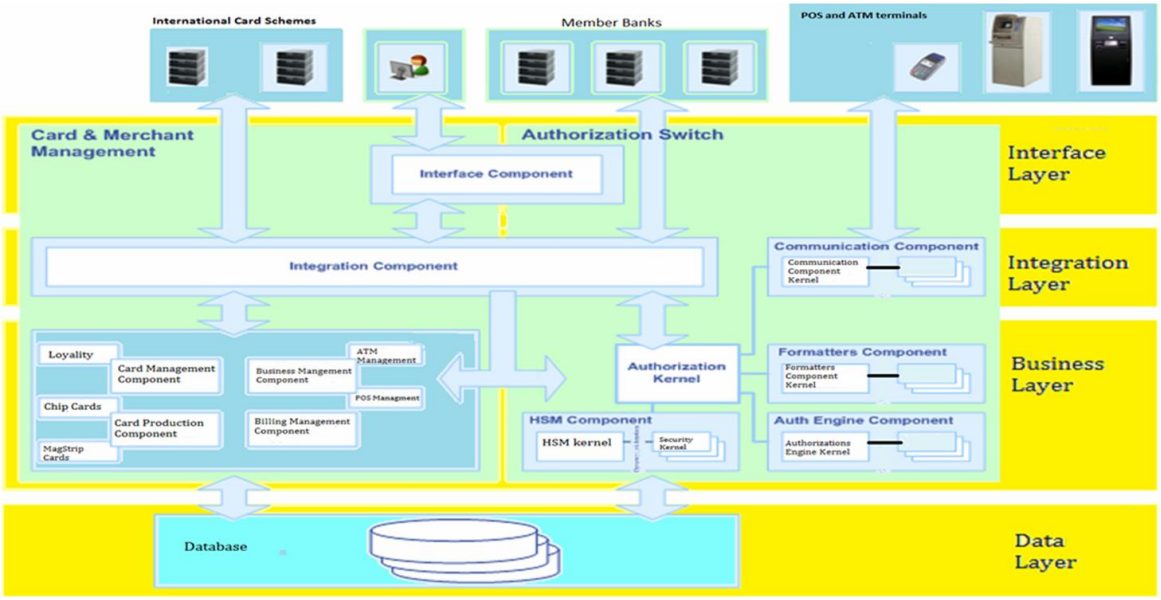


Figure 1: High Level Layered Architecture

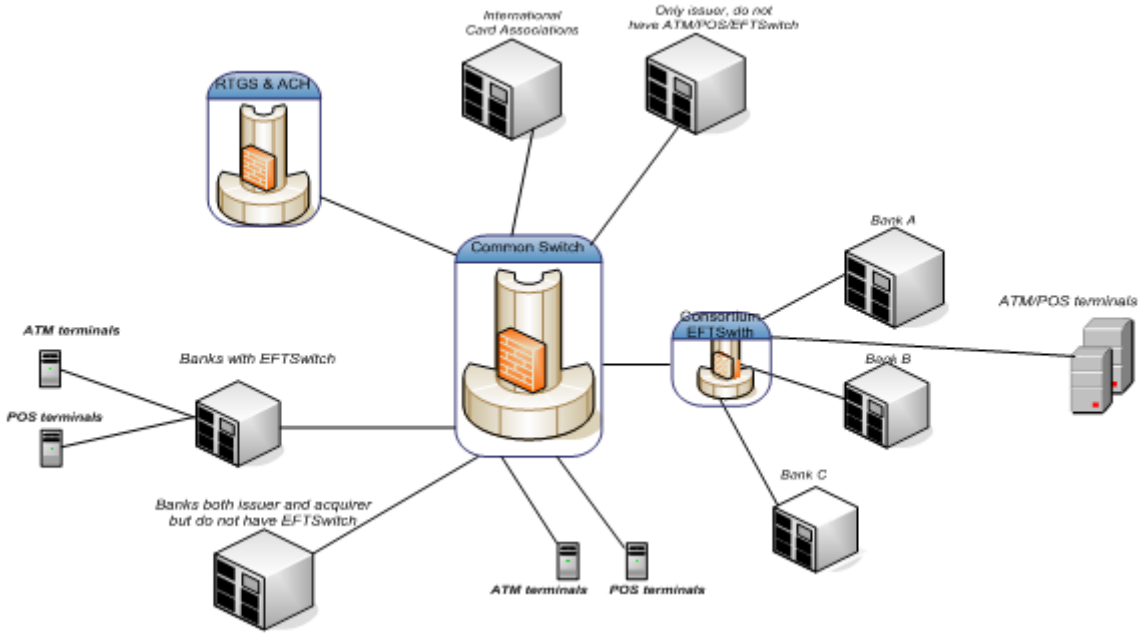


Figure 2: High level logical network Architecture

4.2 Per the technical, governance and business constrains

Technical: One of the technical constraints discussed was the issue of lack of knowledge and capital inadequacy to invest on the high availability and redundant payment systems. Based on this constraint, the recommended EFTSwitch will solve same by creating redundancy for those who have EFTSwitch with Stand in Transaction Processing (STIP). Additionally this common switch can create the technical personnel by producing knowledge base on payment systems, finally will create knowledge base for all members bank.

Governance: With the recent GTP plan, one of the government's initiatives is to enhance the financial institutions to reach unbanked societies and creating financial inclusions. Based on this, the proposed common switch will facilitate the payment system by inviting banks who haven't EFTSwitch for fast implementations and integrations.

Business: The business modality proposed is closed national switch, with establishment of a common company to run the business of the EFTSwitch platform. This will solve the business constraints of the limitations associated with both local and foreign outsourcing on open with outsourced Processing. Additionally there is also an

advantage of easy implementation, on top of the gradual learning and skill development is more attainable with the option of establishing a common company on the closed national switch business modality.

4.3 Response to different platforms

The proposed platform can work for interoperability in different platforms that will work to avoid the duplication of resource and knowledge for banks.

Banks with EFTSwitch: On the proposed modality banks with EFTSwitch implementations can use the proposal for resource sharing. They can connect with the common switch for interoperability and resource sharing. Also the common switch can be configured for STIP processing for those who have EFTSwitch when there is an issue on their EFTSwitch.

Banks without EFTSwitch: The proposed solutions are able to accommodate a direct connectivity for banks without EFTSwitch. In this platform they can use as Third Party Processors (TPP) model to deploy their ATMs and POS terminals if they want to continue in the acquiring models or to accept transactions for issuing model.

5. Related Work

5.1 The Berlin Group

With the vision of European Central Bank (ECB), the European Commission (EC) and the European Payment Council (EPC) on card payments in a single Euro Payment Area (SEPA), they created The Berlin Group with the participation of 23 major players in the card industry from 10 different Euro-zone countries and from the UK, Sweden, Denmark, Norway, Iceland, Latvia, Estonia, Lithuania, Turkey, Croatia, Bulgaria, Hungary and Serbia aimed on capitalizing and preserving the high levels of efficiency, brand awareness, security, convenience and ease of use in national card schemes [5].

The group has explored the feasibility of this concept with compatible card schemes within the Euro-zone and has defined a common set of standards for interoperability between the acquirer host and the issuer host, which is independent from a specific card scheme. The principal goal is to meet the aims of the EPC, the ECB and the European Commission with regard to a Single Payment Area, and in particular to be compliant with the SEPA Cards Framework which has been developed by the EPC.

The Berlin group concentrated on the technical interoperability than discussing on Organizational or Semantic Interoperability. It requires crafting a framework on Organizational and Semantic interoperability before using the technicality in the case of Ethiopian scenario, hence adapting the Berlin group approach to our own use may lack the wholeness of the issue raised on the problem analysis.

5.2 The Euro Alliance of Payment Schemes (EAPS)

The Euro Alliance of Payment Schemes (EAPS) is a joint initiative by European payment schemes, enabling European cardholders and retailers to make card payments and cash withdrawals with payment card across Europe's national boundaries. The objective of the EAPS is to enable participating schemes to offer their member banks with a new alternative for pan-European card payments [6].

The EAPS believes that the implementation of the open Berlin Group Standards enhances competition and strengthens European presence and influence in the card payment market. The objective of EAPS is to deliver facilities along the vision of European banks, card holders, business and regulators for secure, reliable and efficient card payments across Europe. The standardization work of the Berlin Group plays an important role in supporting the efficiency of European card transactions and guarantees independence from proprietary standards.

6. Conclusion

Currently Ethiopian banks are willing to invest in a modern payment system to expand the range of services, promote efficiency, and bring down cost. The restricted number of Automatic Teller Machine (ATM) and Point of Sales (POS) terminals in the country should expand using the interoperability mechanism in order to offer their cardholders more convenient and efficient payment services and to

widen their client-base. Within this context, the proposed requirement driven interoperability

model will help in using their clients in more efficient ways for the existing transactions and other projected banking services in meeting the demand of the banking sector client-friendly. It is convenient and cost-effective for communication and processing payment, creating large customer base for more efficient payment transactions, development of efficient and low cost channels for payment systems as well as reducing the risks in the infrastructure for payments and clearing and settlement transactions.

References

[1] S. Sumanjeet, Emergence of payment systems in the age of electronic commerce: the state of art, University of New Hampton, 2009.

[2] "IEEE Standard on interoperability", retrieved from http://www.ieee.org/ def?/definition.htm, Last Accessed on June 14, 2012.

[3] H.Kubicek et al., Organizational Interoperability in E-Government, DOI 10.1007/978-3-642-22502-4_7, Springer-Verlag Berlin Heidelberg, 2011.

[4] "National Bank of Ethiopia- Requirements for Core Banking Systems", Circular, 2011.

[5] P. Martin, Transaction EFT Switching: E-banking processing, Visa Inc, Abuja, 2010.

[6] "Bilateral and Multilateral Processing of International Transactions in Europe Clearing and Settlement Interface Specification" retrieved from http://www.berlin-group.org, last accessed on July 05, 2012.

Enabling Seamless Sharing of Data among Organizations Using the DaaS Model in a Cloud

Addis Mulugeta
Ethiopian Sugar Corporation, Addis Ababa, Ethiopia
addismul@gmail.com

Abrehet Mohammed Omer
Department of Computer Science and IT, Addis Ababa Science and Technology University, Ethiopia
abrehet@gmail.com

Abstract

Organizations are struggling to allow seamless data sharing and synchronization with the intention of data verification and avoid data redundancy. Currently, many organizations in Ethiopia share data manually using a formal letter. This is not economically as well as technically feasible. It is tiresome, inefficient, error prone and impractical as the systems and problems become larger and more complicated. This paper explores data sharing methods anticipated functionalities through inspecting real cases and different existing data sharing mechanisms by proposing an extended framework, including: (1) identifying the core issues that have to be addressed during data sharing, i.e., data owners autonomy, heterogeneity, and need of naming standard; (2) developing a customized conceptual framework, which allows providing data as a service and specifies different components of data as a service provider and service consumer, i.e., Acquaintance manager, Query manager and ECA rule manager; (3) developing a detailed implementation framework; (4) demonstrating the framework by implementing a prototype based on the implementation framework. The proposed approach takes advantage of the newly emerging computing paradigm, cloud computing, that enables providing Data as a Service.

Keywords: Cloud Computing; Data as a Service; Data Sharing

1. Introduction

The need for sharing data between different organizations has been increasing with the technology and economic growth of the country. Sharing data among organizations will help organizations to increase efficiency and performance by reducing manual work, in order to get accurate data, and to improve delivery of the service for the customer. In addition, with the current high mobility and advancement of technologies, few illegal persons take advantage of sharing data between the organizations for illegal purpose that increase the demand of data verification.

Moreover, many organizations build their own database to be utilized only within the organization even if the same data is stored in another organization. As a result, the same data might be stored in multiple locations resulting in data redundancy. This redundancy can be avoided by sharing existing data.

Advances in computing paradigms and technologies triggered a new multitude of data sharing mechanism, which is providing data as a service. This new computing paradigm is called cloud computing which promises the provision of providing anything as a service [2]. Cloud includes the different participants involved in the cloud along with the attributes and technology that are coupled to address their needs and different types of service "XaaS" [1] X is a software, hardware, platform infrastructure, data business, etc. Everything is a service like Software as a Service (SaaS), Hardware as a Service (HaaS), Data as a Service (DaaS), Infrastructure as a Service (IaaS), etc.

DaaS is based on the concept that the product, data in this case, can be provided on demand to the user regardless of geographic or organizational separation of provider and consumer. The concept of DaaS advocates the view that with the emergence of service-oriented architecture (SOA), which includes standardized processes for accessing data "where it lives", the actual platform on which the data resides doesn't matter. With data-as-a-service, any business process can access data wherever it resides [2].

Currently, many organizations in Ethiopia share data manually using a formal letter for the purpose of data verification. This is not economically as well as technically feasible. It is tiresome, inefficient, error prone, and impractical as the database, systems and problems become larger and more complicated. In addition, from the real case study analysis it is observed that: (1) when any organization builds a database, it only considers internal requirements which imply data is stored in heterogeneous manner; and (2) all organizations decide to share or not to share data autonomously for security purpose. Thus, any data sharing mechanism is expected to be secure, flexible, and personalized.

This paper discusses ideas concerning the data sharing mechanism that includes the expected functionality mentioned above. Our approach relies on the concept of providing data as a service based on demand. We define our approach by developing a conceptual and detailed implementation framework that includes all components to handle heterogeneity, keep autonomy, and allow data synchronization.

The remaining part of the paper is organized as follows: Section 2 provides one of the case studies used to identify the problem at hand. Section 3 gives an overview of the proposed framework, conceptual as well as implementation framework. Section 4 presents the prototype implemented to demonstrate the proposed framework. Section 5 presents related work and compares our approach with some related work and finally Section 6 presents conclusions.

2. Case Study

The main purpose of this study is to know how an organization handles data and the sharing mechanism, the policy of sharing data and a detailed study on the organization, which verifies data of many persons with different organizations.

The case study is done on the Federal Ethics and Anti Corruption Commission (FEACC). One of the tasks of this commission is to register and verify the wealth of an appointee of some government employees. The commission has a directorate that registers and verifies thousands of appointee employees and their family data with all banks to check their account, municipality data to check house and land they owned, investment office to check their investment activities, and with other organizations in which each organization may have hundreds of branches all over Ethiopia.

Even though all Banks, Ethiopian Revenue and Custom Authority, Ministry of Trade, sub cities and other organizations own databases, there is no automated way to share data. When data verification is needed, the organization verifies data manually using a letter. In addition, there are many organizations with similar situation in the country, which need data for sharing and verification purpose; for example, banks to verify the legality of collateral, courts to check the legality of different documents from different organizations.

According to the study in the organization

1. The organizations are Autonomous
2. The databases found in the organizations are heterogeneous
3. There is no attribute naming standard.

The core issues to be addresses are the heterogeneity of the databases and the autonomy of database on each organization.

3. The Proposed Solution

From the different data sharing and the case study, we designed a conceptual framework and a detailed implementation framework. We adopted and modified the approach from the Hyperion project to fit the requirement from the case study.

3.1 Conceptual Framework

The proposed conceptual framework follows a peer-to-peer data sharing mechanism based on the Hyperion project [3]. It has two kinds of peers: the service provider peer and the service receiver peer as shown in Figure 1.

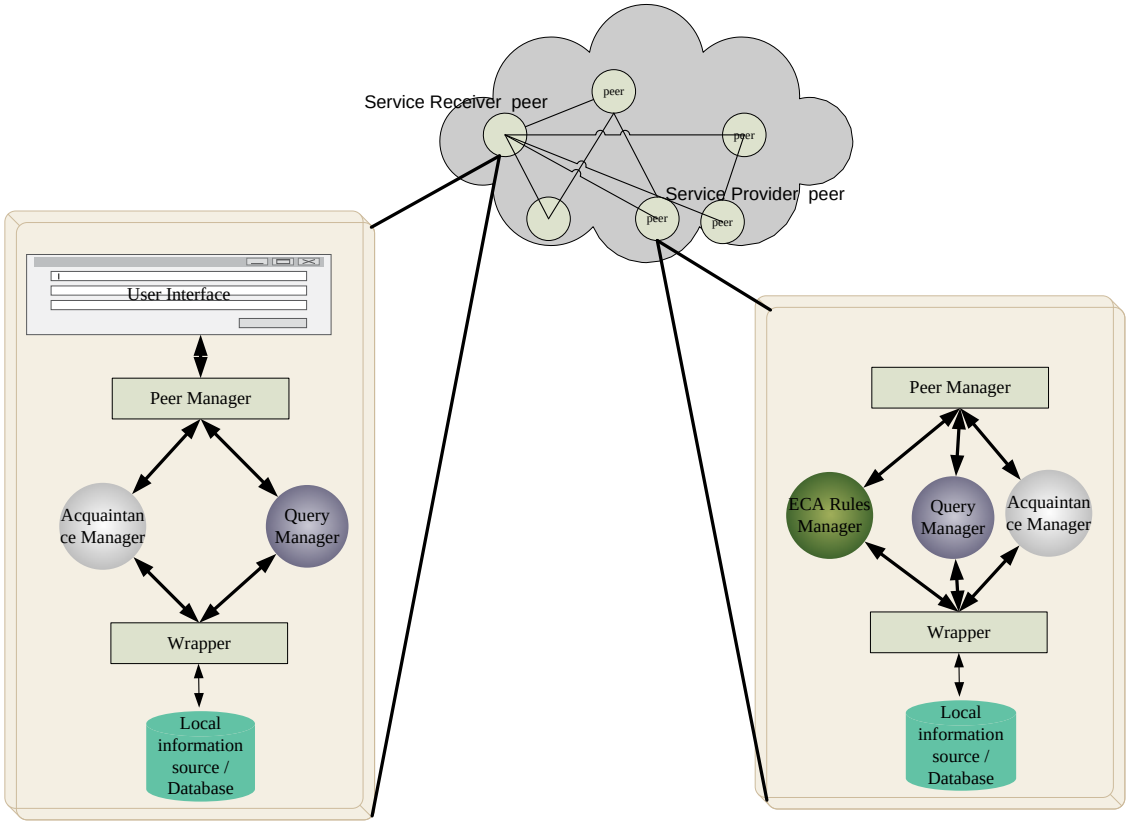


Figure 1: Proposed Conceptual Framework

3.1.1 Services Provide Peer

The major components are:

1. *Acquaintance manager:* manages the exchange of public schemas, mapping tables, and coordination rules between new acquaintances and the inference of new mappings.
2. *Query manager:* facilitates the ability to execute local queries to the format in which the query that the data model holds, and the result from the data model to the users can understand by rewriting the queries and mapping.
3. *ECA rule manager:* manages and executes distributed event condition rules in order to enforce consistency of the shared or provided data.
4. *Wrapper:* handles data conversion and table mapping from the organization's format to the data model format.

3.1.2 The Service Receiver Peers

Similarly, the service receiver includes the following components:

1. Acquaintance manager
2. Query manager
3. Wrapper: the wrapper in the service receiver changes from the data model format to the format which the user needs

3.2 Implementation Framework

For the actual implementation of the conceptual framework, a detailed framework is developed as shown in Figure 2. This framework gives details on how a data provider peer can provide services and how the service consumer peer can receive the service. The different components of service provider peers that enable providing data as a service are described below.

a. Database

The databases in different organizations are in different format. The files vary from simple flat files like Excel or text to commercial databases like Oracle, MS SQL, MySQL, etc.

b. Wrappers

The main purpose of the wrapper is to convert heterogeneous databases into a single format and to resolve conflicts caused by heterogeneous databases and put the data on the data model. XML is selected for the database. XML technology makes the

standardization description of all kinds of irregular information and rule information possible, and gradually becomes the standard to describe the data on the Internet, and sets up an enterprise information integration platform in the XML technology, and is an inevitable trend of information technology development [4]. In the wrapper, there is a schema and a table mapping. On the schema level, all the components of the database are changed to the XML database format. Each attribute on the database is mapped to the attribute of the standard data model. A standard data model is a table which is designed to accommodate all of an organization's databases with different data types and name to the standard name and data type which the user understands and queries easily.

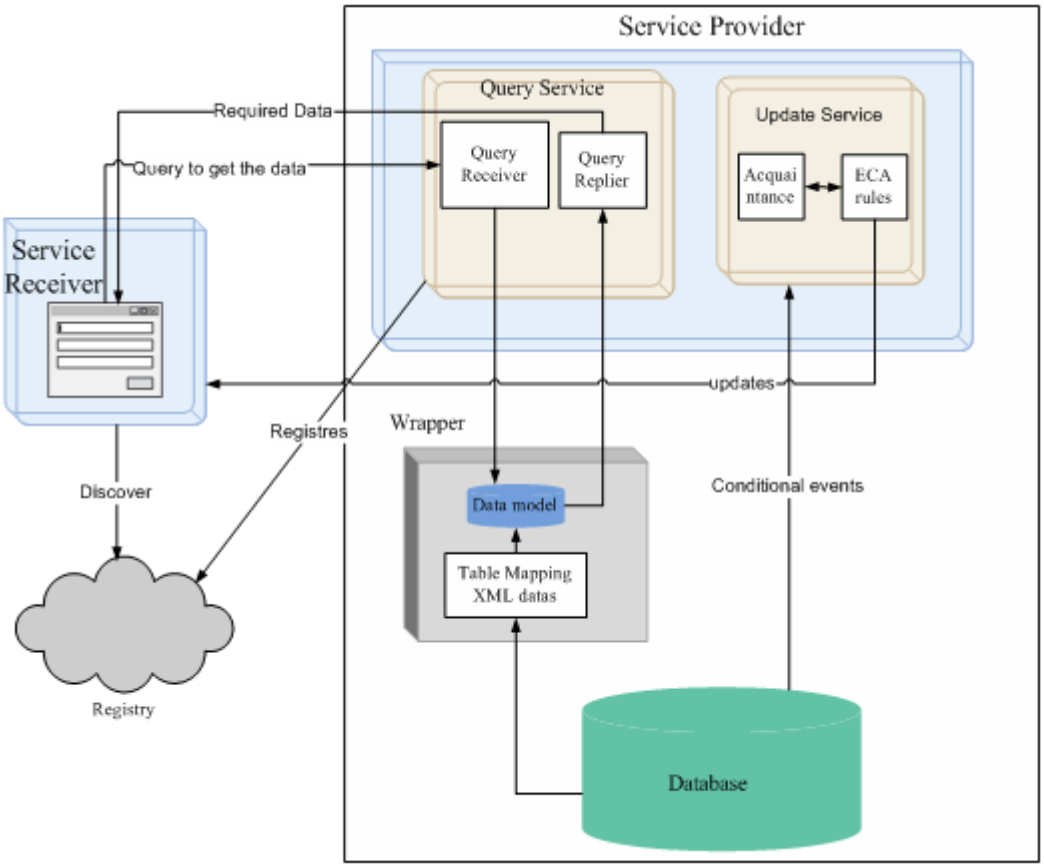


Figure 2: Proposed Implementation Framework

c. Update Service

This service has two components: acquaintance and ECA rules. Acquaintance is used to store and manage the peers. ECA is a coordination rule that describes when, how, and where a message should propagate to the concerned peers based on the agreements of peers (organizations). These components allow data synchronization among peers.

d. Query Service

The main purpose of the query service is to provide a unified interface for the client to query the required data from the heterogeneous databases. The query service accepts the standard query from the interface and changes to the query that is used to fetch the data from the data model that is found in XML format.

In addition to the above listed components, the framework includes the service receiver and registry components which are actually included to show the complete system of DaaS in a Cloud.

The service receiver can be any organization which needs the data from different organizations for verification and sharing. The registry holds constantly evolving information about the available services in the service provider which allows the service receiver efficiently discover and

communicate with service providers. When the service receiver needs data, it gets the available services from the registry and then communicates with the service providers and using a browser the receiver gets the necessary information.

4. Prototype

A prototype is developed to demonstrate and validate the proposed framework. The prototype shows all the implementation except the update service.

The prototype is developed with the Talend open software and Microsoft Visual Studio 2010. Talend open source software is used to implement the wrapper task.

The wrapper

- Copies the data from the organization,
- Changes the database from any format to the XML format or any other standard format,
- Maps the table of the organization into the data model table,
- Puts the data in the data model, and
- The data model is ready for the service.

The Service Receiver

- Searches the available service from the registry and gets the address of the registry,
- Within the address enters into service receiver page,
- In the page the user fills the form accordingly and posts the data,
- The service provider accepts the data query, process it and returns back the result,
- The data which is obtained by the query in XML format is changed to the standard format, and
- The data will be displayed in the page.

The prototype is designed to convert data in the organization that intends to share its data to the XML format and put in a file. Using a web interface from the XML file, the user queries data and the result is shown in Figure 3.

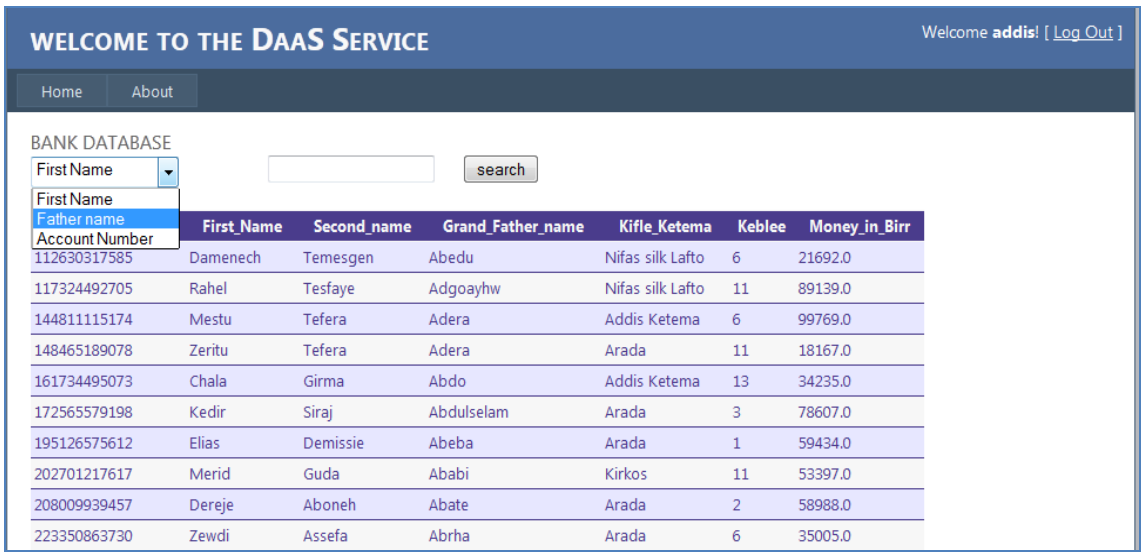


Figure 3: Interface for the user

5. Related Work

5.1 NETDB2

NetDB2 is a data sharing mechanism proposed in [5]. The basic NetDB2 system is implemented as three-tier architecture, namely; the presentation layer, the application layer, and the data management layer. The benefits of the layers are the insulation of software components of one layer from another and better interoperability and higher scalability. This data sharing mechanism is used by a number of universities to help teaching database courses at different locations.

5.2 Clinical Pharmacology and Data Management

Bayer Schering Pharma proposes a data sharing technique that enables sharing of data among

different departments (entities) of clinical pharmacology [6]. It proposes a centrally managed data sharing mechanism known as an exchange area. An exchange area is a static directory tree on a UNIX server with subdirectories. It is used to archive files that had been transferred or shared. All data for one transfer are compressed in a ZIP archive and named according to the direction of data transfer. UNIX drives are mapped within the environment of other operating systems. The transfer from the area to the departments is performed automatically by a cron job in predefined time interval. Cron is a UNIX utility that allows tasks automatically run in the background at regular intervals.

5.3 Hyperion Project

Hyperion project is a system that supports data sharing for a network of independent peer Relational Database Management Systems. In the Hyperion project, each peer includes a database with its own schema and data. Peers can join or leave the network at their own discretion. Moreover, a peer may form an acquaintance with another peer for data sharing purposes. These metadata take the form of mappings, both at the data level and schema level, and they help to bridge semantic and syntactic heterogeneities between peers. Metadata at the data level are expressed as mapping tables [3]. The framework fulfills all the necessary requirements as shown in Table 1 that are not fulfilled in any of the data sharing mechanisms mention above.

Table 1: Comparison of Data Sharing

	Data source Heterogeneity		*Data base type*	*Autonomy*	*Data as a service*
	Schema Matching	*Table mapping*			
NET DB2	Yes	No	Relational	No	Yes
Hyperion	No	No	Relational	YES	No
Clinetics Data Merge	No	No	Any Type	No	No
Proposed Architecture	Yes	Yes	Any Type	Yes	Yes

6. Conclusion

In this paper a framework is proposed which is used to share data seamlessly by autonomous organizations from heterogeneous databases. This approach uses a cloud computing DaaS model in which the data is available as a service. The users get the data wherever there is a browser and a connection regardless of the operating system or the need of other software. An XML database is used to facilitate the provision of data as a service. The proposed framework is demonstrated with a prototype for which any organization can easily share data by keeping their autonomy and with heterogeneous database systems.

References

[1] B. P. Rimal, E. Choi, and I. Lumb, "A Taxonomy and Survey of Cloud Computing Systems," *in 2009 Fifth International Joint Conference on INC, IMS and IDC*, 2009, pp. 44–51.

[2] J. Dyché, "Data as a service explained defined." http://searchdatamanagement.techtarget.com/answer/Data-as-a-service-explained-and-defined, accessed on 12-Nov-2011.

[3] A. Kementsietsidis, "Data Sharing in the Hyperion Peer Database System," on Very Large Data, pp. 1291–1294, 2005.

[4] W. Xiaoli and Y. Yuan, "XML-based Heterogeneous Database Integration System Design and Implementation," *2010 3rd International Conference on Computer Science and Information Technology*, pp. 547–550, Jul. 2010.

[5] H. Hacigumus, B. Iyer, and S. Mehrotra, "Providing Database as a Service," *in Proceedings 18th International Conference on Data Engineering*, 2002, pp. 29–38.

[6] B. V. Gmbh and A. Schwarzer, "Clinetics Data Merge - A Platform for Exchange of Pharmacokinetic Data," pp. 1-5, 2010.

Unit Selection Based Text-to-Speech Synthesizer for Tigrinya Language

Agazi Kiflu
agazi_s@yahoo.com

Tibebe Beshah
School of Information Science, Addis Ababa University, Ethiopia
tibebe.beshah@gmail.com

Abstract

This paper brings together the development of the first unit selection based Text-to-Speech (TTS) system for Tigrinya using the Festival framework and practical applications of it. Construction of a unit database and implementation of the natural language processing modules are described and a Unit selection-based approach generates speech by selecting proper units from a speech corpus and connecting them together. In this approach, a set of features are defined to describe the speech units in the corpus and the expected units in the synthesized utterance. In this paper the major tasks have been performed, via development of concatenative Unit selection voice using phone as basic unit. We have used a speech corpus having a size of 4 hour, 38 minutes and 29 seconds, labelled at phoneme level.

We describe the implementation and evaluation of a G2P conversion model for a Tigrinya TTS system. Letter to sound conversion for Tigrinya usually has simple one to one mapping between orthography and phonemic transcription for most Tigrinya letters and an automatic clustering technique to cluster units based on their phonetic and prosodic context. Having constructed the phonetic, prosodic, and acoustic features extraction inventory for each phone to synthesize the input text the Festival speech synthesis was then adopted in order for the synthesizer to use cluster unit selection algorithm. In order to minimize acoustically defined target and join costs, a selection is made from cluster at the time of unit's synthesis.

The test results indicate that almost all of the words and sentences are recognizable. The system is evaluated using MOS, one of the most popular testing techniques in speech synthesis. The system is tested for naturalness and intelligibility of speech. On average, 97.1% of the sentences are correctly recognized by the listeners. The naturalness of the synthesized speech demonstrates the appropriateness of the proposed approach.

Keywords: Text-to-Speech Synthesis; Concatenative Speech Synthesis; Unit selection based speech synthesis; Syllable based concatenation; Consonants and Vowels

1. Introduction

Text-to-Speech (TTS) synthesis can convert arbitrary input text to intelligible and natural sounding speech so as to transmit information from a machine to a person [1]. It is a process through which input text is analyzed, processed, and understood and then the text is rendered as digital audio and then spoken [2]. The basic types of synthesis system are Formant, Concatenated, and Articulatory [3].

The process of TTS conversion allows the transformation of a string of phonetic and prosodic symbols into a synthetic speech signal. The quality of the result produced by a TTS synthesizer is a function of the quality of the string, as well as of the quality of the generation process. The most important qualities of a speech synthesis system are naturalness and intelligibility [2]. Naturalness describes how closely the output sounds like human speech, while intelligibility is the ease with which the output is understood. In this research, we used the concatenative text-to-speech system and the issues relevant to the development of a Tigrinya speech synthesizer using different choice of units: a word, phrase, clause, sentence or phonemes as a database. Since there is great advancement regarding TTS in other languages globally, attempt will be made to design and implement a method for TTS system in one of the local language of Ethiopia.

This paper is organized as follows: Section 2 focuses on the nature of the Tigrinya script. Section 3 explains literature review, while Section 4 outlines the methodology of the proposed solution with the

voice building process. Section 5 shows the results of perceptual testing and finally conclusion and recommendation are given in Sections 6 and 7.

2. The Tigrinya Language

The script of Tigrinya is phonetic in nature. It uses different choice of units: a word, phrase, has 39 consonants and 7 vowels [5, 9]. The orthographic representation of the language is organized into orders. Each of the 39 consonants has seven orders (derivatives). Six of them are CV combinations while the 7th is the consonant itself. The way Tigrinya orthographic characters are written is very similar to the way they are spoken. It means Tigrinya is a phonetic language. The mapping of the written form and the spoken form is one to one except the epenthetic vowel. Characters representing the same consonant followed by different vowels are similar in shape. For example, here are the characters representing: /he/, /hu/, /hi/, /ha/, /hie/, /h/ and /ho/: **ሀ ሁ ሂ ሃ ሄ ህ ሆ**.

The total number of orthographic Symbols of the language exceed 273. Like other languages, Tigrinya also has its own typical phonological and morphological features that characterize it. Among these, we found gemination of consonants and the use of the automatic epenthetic vowel to be very critical for naturalness in Tigrinya speech synthesis. Tigrinya language has special property in its spoken form (CV or CVC sequence of the acoustic form of the orthographic representation).

2.1 Phonology of Tigrinya Word

Phonology is the study of the distribution and patterning of speech sounds in a language and of the tacit rules governing pronunciation [4]. In phonology, phoneme is the fundamental unit that describes how speech conveys linguistic meaning. The phoneme represents a class of sounds that convey the same meaning. The meaning of a word is dependent on the phoneme that it contains [4].

Tigrinya has a fairly typical set of phonemes for an Ethiopian Semitic language. That is a set of ejective consonants and the usual seven-vowel system. Unlike many of the modern Ethiopian Semitic languages, Tigrinya has preserved the two pharyngeal consonants which were apparently part of the ancient Ge'ez language and which, along with [x'/ቐ]a velar or uvular ejective fricative, make it easy to distinguish spoken Tigrinya from related languages such as Amharic [9]. These are exception characteristics from Amharic beside their accent, manner and place of articulation

qa	k'	ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ
qwa	kw'	ቈ		ቍ	ቋ	ቌ	ቍ	
kxa	x	ኸ	ኹ	ኺ	ኻ	ኼ	ኽ	ኾ
kxwa	xw	ኾ		ኽ	ኻ	ኼ	ኽ	
qha	ʁ'	ቐ	ቑ	ቒ	ቓ	ቔ	ቕ	ቖ
qhwa	ʁw	ቘ		ቝ	ቛ	ቚ	ቝ	

Figure 1: Tigrinya Syllabic Structure

A syllable in Tigrinya is made up of only /**cv**/ and /**cvc**/ (a consonant + a vowel) or (a consonant + a vowel + a consonant). The vowel is a syllabic nucleus, while the first and the last consonants of the syllable are an onset and a coda respectively [9]. The Tigrinya native speaker, for example, can divide the words (sabara) and (biili) into three and two syllables respectively. For example, /qaatala/ (Figure 2) is one word which has three syllables in it. Some syllables of Tigrinya have a nucleus or peak and an onset, while other syllables have a coda in addition to an onset and a peak. Observe the following:

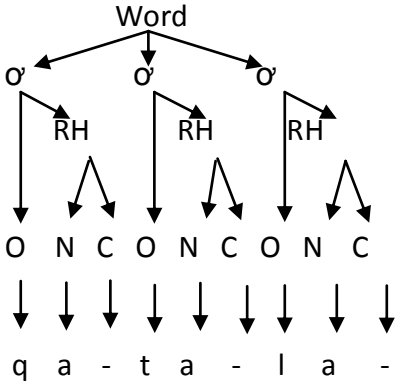


Figure 2

Moreover, each onset and coda position is occupied by a consonant, where as a nucleus position is occupied by a vowel.

2.2 Gemination

Longer duration of identical segments, adjacent consonants or vowels that are the same can form

germination. In Tigrinya sequence of vowels is not permissible. Whenever sequences of vowels occur, either one of the vowels must be deleted or epenthetic segments are inserted between the vowels. However, we do find geminated Tigrinya segments, with the exception of laryngeals and pharyngeals that may be geminated in only very limited environments as indicated in [9].

Consonant germination may bring meaning differences in words. If we compare /zawara/ 'he got roaming' and /zawwara/ 'he drove', /halifu/ 'he passed' and /hallifu/ 'he excelled'. There is a difference of meaning in each pair. In each pair, we observe a geminated or ungeminated medial consonant that brings a meaning difference in each of them.

2.3 Insertions

Insertion is one way for arriving at a well formed or acceptable assignment of syllable structure. The syllable structure of Tigrinya is either /cv/ or /cvc/. Insertion, unlike deletion, is the appearance of new elements in a formerly unoccupied position. The epenthetic (inserted) segments may appear word initially, word medially or word finally. There are several Tigrinya epenthetic segments, vowels and consonants, in different positions. Observe the following:

- asraha 'he made others to work'
- Awassaxom 'he made them to add'

The morpheme /a/ is added to the root consonants/srh/

- zii + asriih-a - zasriiha
- zii + awassaxa - zawassaxa

3. Literature Review

In [7] Indian Natural Language Processing Lab Centre for Development of Advanced Computing (CDAC) uses multiform speech unit to develop the speech synthesis. It primarily uses syllable and phonemes. The speech corpus contains most frequent words and initials. They have been segmented and labelled into different speech units as required for development of a Hindi speech synthesis system. This research doesn't indicate as to whether any kind of implementation has been made or not. Additionally, it is not known to have been tested or proven in any related manner.

Alam *et al.* in [8] proposed a TTS system that creates the voice data for festival, and additionally extends the use of festival to its embedded scheme scripting interface to incorporate Bangla language support. The researchers' TTS implementation used two different kinds of concatenative methods supported in Festival: unit selection and multi-syn unit selection [8].

The researchers on their future work indicated that a number of future plans need to be made to develop the complete TTS system for Bangla language including the following: document analysis, text analysis, phonetic analysis, developing large number pronunciation lexicon, automatic lexicon entries instead of adding manually, find out LTS or Grapheme-to-Phoneme (G2P) rule so that it can handle unknown words), prosody analysis, and waveform synthesis by diphone technique. In conclusion, the researchers observed that unit selection and multisyn unit selection has a drawback because of the requirement of large set of speech corpus.

Eker in [6] has found a research which exploits the Turkish language structure and tried to implement the system that takes a text as its input. It assumes that the text consists of words and it processes word by word [6]. When a word is obtained from the text, it is passed to a unit that can process word as text and produces the corresponding speech. This part separates the word into diphones; using diphone database, it gets a speech file corresponding to diphone and its pitch value.

Finally it concatenates the previously recorded speech segments using PSOLA algorithm and manages to produce sound. As a future work, the researcher recommended that the first thing should be done is to complete the diphone database and apply more experiments on words. The produced output is acceptable for small sentences, but it requires much time for long sentences. Therefore, in order to have a real-time reading system, the system should be faster. In conclusion, this means that the method used in this paper is an applicable one which with some effort on completing and preparing a

better diphone database will result in a system that will produce more understandable output for all Turkish words.

We have observed that few research attempts were made on local languages. One of the few attempts made was by Sebsibe H/Mariam *et al.* in [13]. Their focus was issues that need to be considered in developing a concatenative speech synthesizer. They have tried to describe the issues to be considered in developing a concatenative speech synthesizer for Amharic language. The complexity of the syllable structure of the language, the phonetic nature of the language, and the result of the perceptual test of the synthesizer has been discussed. The researchers tried to explore the nature of Amharic script representation of the phone set, Amharic syllable structure and syllabification rules, and showed the voice building process. Having noted that the quality of speech synthesiser for Amharic was not high, they recommended on the need in the future to work on improvement of the quality desired. They suggested that this can be done by:

1) Proper selection of unit. Since the language is phonetic, syllable as a basic unit may outperform the phone as a basic unit.
2) Optimal selection of corpus, which proportionally covers all basic units and variations, will give better quality.

Based on the reviewed made so far and knowledge of the researchers, none of the works so far have tried to design grapheme to phoneme converter or letter to sound speech synthesizer for Tigrinya. None of them show prototype for natural sound for Tigrinya, which synthesize by accepting normalized Tigrinya texts and generate prosodic features (i.e., intonation, stress) using syllable based approach. The main focus in this work is to find the proper quality speech corpus, which matches the quality of synthetic speech from synthesizers including linguistic tasks and develop naturally sounding text to speech for Tigrinya language.

4. Methodology

After an extensive literature review regarding concatenative speech synthesis method, unit selection concatenative synthesis is found to be the most popular method of performing speech synthesis recently and is found to differ from older types of synthesis by generally sounding more natural and spontaneous than formant synthesis or diphone based concatenative synthesis. Unit selection synthesis is proven to score higher than other methods in listener ratings of quality but it involves a tedious recording many hours of speech by a single speaker.

In this research we tried to explore the nature of Tigrinya script representation of the phone set, rules of letter to sound, Tigrinya syllable structure and syllabification rules that would show the voice building process. To do these researches we used the following.

- Transliteration scheme based on orthographic ordering of the script and acoustic similarity of the letters were defined using ASCII characters.
- In Festvox, the phone set of the language is described with the corresponding features like voicing, tongue position, tongue height, place of articulation, and manner.
- Experiments on Phonology of Tigrinya word, phone set and we try to cover all phonemes defined a transliteration scheme using ASCII characters.

5. Design and integration of Tigrinya unit selection into festival frame work

The speech inventory is divided into clusters, where each cluster holds units of the same phone class based on their phonetic and prosodic context. An outline of the steps to build a unit selection synthesizer are given below. A more detailed description of same is available in [10, 13].

- Design speech and text corpus
- Creating LTS rules and phone set
- Building utterance structures
- Generating speech unit clusters
- Building the unit synthesizer

What have been done in each step to build unit selection voice for Tigrinya is explained below. Tigrinya proverbs, articles, newspapers, magazine and bible sentence’s are collected from different sources and are primary data. We selected a native

speaker of the language and tried to record in quite environment by a male speaker using PRRAT. We used wave surfer for manual labelling of the recorded voices. In this paper, we built a corpus of around 13171 words. The script of this speech corpus is selected from a large text corpus (around 84000 characters). The corpus is designed to cover the frequently used syllable and context as much as possible.

The input to the TTS system is the transliteration of a text in Tigrinya. The pronunciation generation module generates the sequence of basic units using a lexicon of units and letter-to-sound rules. The lexicon is a list of all speech units - monosyllables, bi-syllables and tri-syllables, present in the waveform repository. The letter-to-sound rules are framed in such a way that each word is split into its largest constituent syllable units As the pronunciation of most of the words in Tigrinya can be predicted from their orthography, these rules suffice to generate correct pronunciations. The unit selection algorithm generates a target specification for the speech units that have been identified and picks the best sequence of speech units that minimize both the target cost and the join cost. The waveforms of these speech units are then concatenated to produce synthetic speech.

5.1 System Architecture

As the system architecture shown in Figure 3, the synthesizer has text analysis and speech synthesis parts. The text analysis part uses grapheme to phoneme converter to match the word to its pronunciation whereas the synthesis part selects the best sequence of units for target specification produced at the end of text analysis, and finally generates the speech from of the speech parameters.

- Defining the phone-set of the language
- Tokenization and text normalization
- Incorporation of letter-to-sound rules
- Incorporation of syllabification rules
- Assignment of stress patterns to the syllables in the word
- Assignment of duration to phones
- Generation of f0 contour

Once a speech repository is in place, the repository is integrated with the Festival framework. An outline of the steps to build a unit selection synthesizer are given below. A more detailed description of same is available in [1, 2, 6, 10].

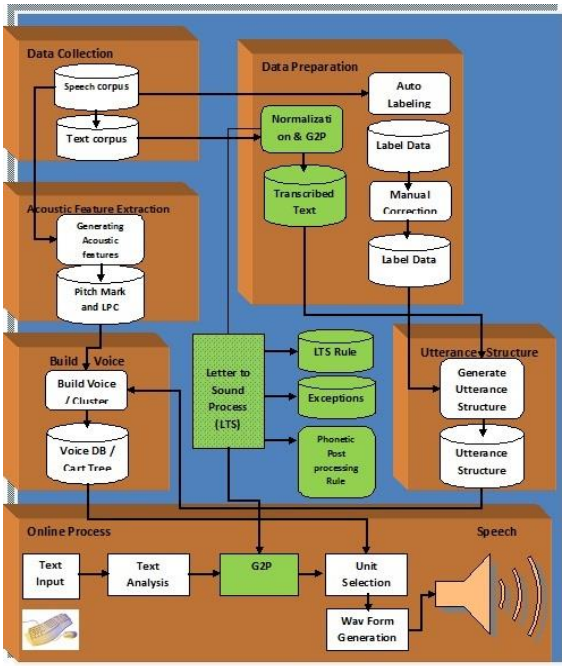


Figure 3: The system architecture of the Tigrinya speech synthesizer using cluster unit selection

5.2 Description of the Implementation Design

After we collected speech and text corpus, the next step was to check recorded utterances against the transcription text in order to design the prompt list in Festival format and correct the label manually. Appropriate modifications have been made to get them ready to be used in voice building process. By doing so the speech and text corpora has been built.

5.2.1 Labelling the Utterance

The process that generates the labelled utterance is labelling. Labelling is the process of giving a label for each speech signal in the utterance. Unit selection synthesizers are highly sensitive to the accuracy of labelling. Bad labels will adversely affect the quality of synthesis in a number of ways [2, 13].

The phone label itself can be incorrect, potentially causing the wrong word to be said, or said with an undesired accent. However, it is time taking and laborious, as part of our efforts to improve speech synthesis, we have labelled the speech database thoroughly using a tool called Wave Surfer.

5.2.2 Creating letter-to-sound rules and phone-set

A comprehensive set of letter-to-sound rules was created to syllabify the input text into the syllable-like units. These rules are framed in such a way that each word is split into its largest constituent syllable unit. The phone set, which is a list of basic sound units for Tigrinya that the synthesizer supports, was created by enumerating all the speech units identified in the syllabification process.

5.2.3 Incorporation of Tigrinya Phone set and Grapheme to Phoneme Converter

The phone-set definition is the first text analysis module in which every phoneme of the alphabet is classified according to phone features like consonant voicing and vowel height. The second text analysis module is the lexicon module.

The Tigrinya phone set is incorporated in Festival corresponding to their characterizing features. Each phone has eight features that describe how the vocal organs behave when the sound is uttered. These features are vowel/consonant identification, consonant voicing, place of articulation, consonant type, vowel length, vowel height, vowel front ness, and lip rounding.

The grapheme to phoneme converter is used to convert an orthographic text into its corresponding phonetic representation. After incorporation of Tigrinya phone set and the Tigrinya grapheme to phoneme converter into Festival, it provides the label files for each sentence in the prompt list. We have made manual label correction using the label automatically generated along with the corresponding wave file.

The grapheme to phoneme converter is used to convert an orthographic text into its corresponding phonetic representation. We implemented the grapheme to phoneme conversion architecture by making modification of syllabification algorithm proposed in [11]. A C# based syllabification program is implemented which is graphical based system and modified into C++ command line based G2P system is done as per the requirement of Festival tools. After incorporation of Tigrinya phone set and the Tigrinya grapheme to phoneme converter into Festival, it provides the label files for each sentence in the prompt list.

5.2.4 Building utterance structures for the database

The ‘utterance’ structure holds all the relevant phonetic and prosodic information related to a speech unit within this data structure. The phonetic information in an ‘utterance’ structure describes the position of the speech unit in the word it appears and the information of units adjacent to it. Prosodic information holds information about the duration and pitch of the unit. Festival provides relevant scripts for building ‘utterance’ structures for each speech unit.

5.2.5 Generating speech unit clusters

The process includes building coefficients for acoustic distances (MFCC, F0 and energy coefficients), creating distance tables for each class of units based on acoustic distances and generation of features for building CART trees.

5.2.6 Building the unit synthesizer

Using the letter-to-sound rules, phone set and clusters of each speech unit built in the previous steps, Festival generates the necessary files that need to be used along with the core Festival speech synthesizer to build a unit selection synthesizer for Tigrinya using appropriate scripts.

The rules of the language in relation to epithetic vowel insertion, gemination and syllabification effect on speech synthesis to determine the pronunciation of given Tigrinya words based on its spelling, in the process of grapheme to phoneme converter.

Epenthetic vowel insertion and germination rule for Tigrinya are adopted from [11] and modified with:

1. Accept input words and scan from left to right.
2. If consonant cluster occur at word initially position, insert epenthetic vowel between them.
   - Exception: If the first phoneme is consonant and the next consonant is glide/w/ pharyngeal/h/ plain/x/ (rule #1).
3. If three consonants are appeared in sequence word medially or word finally, position insert

epenthetic vowel before the third consonant (rule #2).

- Exception: if the middle consonant sonority is greater than the rest insert epenthetic vowel after next the first consonant.

4. If a cluster of consonant contains the germination and singleton in sequence, insert epenthetic vowel after the geminated consonant (rule #3).
5. If a cluster of consonant contains the singleton and geminate in sequence insert epithetic vowel after the singleton consonants (rule #4).
6. If a cluster of consonant contains two different germinations in sequence, insert epenthetic vowel between the two geminate consonants (rule #5).
7. If the sonority of the final consonant is greater than that of the proceeding consonants, the epenthetic vowel is inserted between the final consonant clusters (rule #6).
8. If a consonant cluster occurs at word final position, insert epenthetic vowel /i/.
9. Repeat steps 2 to 7 until the entire phoneme are parsed in the phonemes list.

6. Perceptual Evaluation and Experimental Results

Perceptual evaluation is essential to determine the quality of synthesized speech [13, 14]. The perceptual evaluation in this paper investigates the naturalness and intelligibility of Tigrinya TTS. In this research work mean opinion score (MOS) is used to test the output of the synthesized speech. MOS is an evaluation technique where evaluators indicate their assessments on a scale ranging from bad (1) to excellent (5). Then the average score of the opinion given will be taken as the performance of the system [6, 12, 14].

As we stated in the above section, the impact in perception is assessed by comparing the evaluation average result to be obtained from the score ranks given by native speakers at the end of their perceptual judgment for the synthetic speech produced by the synthesizer. Subsequently, the perceptual tests were carried out to evaluate the extent of naturalness and intelligibility of synthetic speech generated by the speech synthesizer.

6.1 Data Preparation and Prototype Testing

We conducted perceptua1 tests on 6 people who are native speakers of Tigrinya: 2 females and 4 males. All subjects are between 30 to 60 years old. Each subject listens to all of the 6 sentences with various lengths selected from the data set used in the voice construction and gives his/her ranking value for the naturalness and intelligibility of the speech. They evaluated based on the quality of the speech output by giving a measure of quality.

Based on the result found, we can conclude that proper selection of units done by the TTS has great role for perceived naturalness and intelligibility of synthetic speech sounds.

The results show that regarding the question as to whether the voice is good to listen to or not, 38.8% considered the voice is very good, 58.3% of them thought that the voice was good and 2.7 % considered the voice unnatural. From the result it is clear that more than 97.1% of the listeners found it to be ok and none of the listeners found it to be excellent, fair or very poor. In general, the output was acceptable by most of the listeners. When compared to a previous work done on unit selection in [8] for the Bangla language in Festival framework at sentence level, the average score is 90.1%, thus when compared to this thesis it has improved by 7.1% [8].

6.2 Summary

Even if the experiment is conducted on a small scale, the results obtained are promising. From this result, it appears to indicate that with an ever increasing size of speech database, the unit synthesizer would be able to produce natural speech with high flexibility and intelligibility. However, every feature of the Tigrigna language was not considered. This paper has achieved promising result by defining

- Transliteration scheme to work with Tigrigna scripts
- Incorporated phone set, Syllabification rules, and Letter to sound rules

- Stress assignment into Festvox.

7. Conclusion and Recommendation

In this work, a first attempt is made to develop a speech synthesizer for Tigrinya language using unit selection method. However, every feature of the Tigrinya language was not considered because it needs a lot of time and detailed and deep linguistic knowledge. Hence, only the characteristics and way of creation of Tigrinya phonemes are considered.

From the result and analysis, it can be concluded that this paper has achieved its main objectives. This project has produced a Tigrinya TTS system that has the ability to process the input of Tigrinya raw text to an output of Tigrinya speech sound. From the sentences level test also, it has proved that the user can understand almost every simple sentences spoken by the system.

The major contributions of this paper are:

- Identification of a new syllable based speech unit and suitable phone set for concatenative speech synthesis for Tigrigna,
- Development of LTS rule for Tigrigna.
- Development of natural sounding TTS systems for Tigrinya.
- Demonstration of the prototype.
- Investigation of techniques to develop speech synthesis systems.
- To assist raise further research question.

There are quite a lot of methods that can be used to improve this system. This improvement may range from its database method to its NLP processing method and the following points are recommended for future work either to extend the work or to increase the quality of the synthesized speech. As a future work we would like to suggest the following points:

- Syllabification of words: will greatly improve prosodic modeling with the segmental prosody of Tigrinya should be appropriately studied and modeled.
- The proper identification of Tigrinya stress point to help determine where should fall within a word.
- Automatic gemination and epenthesis handling algorithm.
- Deep studies on syllabification and final consonant cluster will improve the performance of the syllabifier.
- Stress assignment algorithm.
- Tigrinya morphological analyzer
- Duration modeling of consonants and vowels

References

[1] Thierry Dutoit, “A Short Introduction to Text-to-Speech Synthesis”, TTS research team, TCTS Lab.1997.

[2] A.W. Black, P. Taylor, and R. Caley, “The Festival speech synthesis system”, 1998.

[3] Juergen Schroeter, “Text to-Speech (TTS) Synthesis”, AT&T Laboratories, 2002.

[4] Nadew Tademe Mergia, “Formant based speech synthesis for Amharic vowels,” MSc Thesis, Faculty of Informatics, Addis Ababa University, Ethiopia, 2008.

[5] N. Sridhar Krishna, “Text-to-speech synthesis system for Indian languages within the Festival Framework:, M.S. Dissertation, Department of Computer Science and Engineering, Indian Institute of Technology, Madras, 2004.

[6] Barış Eker, “Turkish text to speech system”, MSc Thesis, The Institute of Engineering and Science of Bilkent University, 2002.

[7] Natural Language Processing Lab Centre for Development of Advanced Computing, “Building speech corpora for unit selection based concatenative text to speech system for Indian Languages”, India.

[8] Firoj Alam, Promila Kanti Nath, and Mumit Khan, “Text To Speech for Bangla Language using Festival”, BRAC University, Bangladesh.

[9] Tesfaye Tewolde Yohannes, “A modern Grammar of Tigrinya,” Rome, Italy, 2002.

[10] Alan W Black and Kevin A Lenzo, “Building Synthetic Voices, For FestVox 2.0 Edition”, Language Technologies Institute, Carnegie Mellon University and Cepstral, LLC, 2003.

[11] Nirayo Hailu Gebregziabher “Modeling Improved Amharic Sylliblification Algorithm”,

MSc Thesis, Faculty of Informatics, Addis Ababa University, Ethiopia, 2011.

[12] Sebsibe H/Mariam, S P Kishore, Alan W Black, Rohit Kumar, and Rajeev Sangal, "Unit Selection Voice for Amharic using Festivox", 5th ISCA Speech Synthesis Workshop, Pittsburgh, pp. 103-107, 2005.

[13] Yonas Demeke, "Duration modeling of phonemes for Amharic text to speech system", M Sc Thesis, Faculty of Informatics, Addis Ababa University, Ethiopia, 2011.

[14] Hyunsong Chung, "Duration Models and the Perceptual Evaluation of Spoken Korean", Proceedings of ISCA Archive, France, 2002.

[15] AlanW Black and Kevin A. Lenzo, "Building Synthetic Voices", For FestVox 2.1 Edition. 2007.

ERP Implementation Framework: The Case of Ethiopia

Alemu Samuel
United System Integrators, Addis Ababa, Ethiopia
alexsmulatu@gmail.com

Mesfin Belachew
Ministry of Communication and IT, Addis Ababa, Ethiopia
mesfinbelachew@gmail.com

Mesfin Kifle
Department of Computer Science, Addis Ababa University, Ethiopia
kiflemestir95@gmail.com

Abstract

Over the past decade, Enterprise Resource Planning (ERP) systems have become one of the most important developments in the corporate use of information technology. ERP implementations are usually large, complex projects, involving large groups of people and other resources, working together under considerable time pressure and facing many unforeseen developments. In order for an organization to compete in this rapidly expanding and integrated marketplace, ERP systems must be employed to ensure access to an efficient, effective, and highly reliable information infrastructure. They have almost become the de-facto option for replacing legacy systems. Despite the benefits that can be achieved from a successful ERP system implementation, there is evidence of high failure in ERP implementation projects. There is evidence that the overwhelming majority of ERP implementations exceed their budget and their time allocations. Researchers have begun to analyze ERP implementation in case studies in order to provide an implementation framework which maximizes efficiencies.

Identifying the Critical Success Factors (CSF) as early as possible can provide valuable clues to help project managers improve their chances of success. This research focuses on seeking the most important CSF that influence the implementation process of an ERP system in Ethiopia.

An implementation framework is designed so that the research questions can be studied systematically. Based on a literature review and a survey study on five organizations, a list of 17 CSFs are identified as important in ERP system implementation in Ethiopia. Based on the survey result an interview guide is prepared for three case organizations. The interview result shows that all the 17 CSFs have significant influence on ERP implementation. However, a significant variation is seen on the degree of importance in state organizations and the private sector on some of the factors.

Keywords: Critical Success Factors; ERP Implementation Framework

1. Introduction

Nowadays Information Technology has become an important part of our daily activity. IT and Information Systems changed the business atmosphere. Production and services grew, quality is improved, and at the same time competition between companies is increased. In this competitive situation, organizations can survive only if they improve quality, decrease costs in their whole supply chain, reduce inventories, diversify their products and services, and provide more reliable delivery service in better ways in comparison to their competitors.

ERP systems have the ability to automate and integrate business processes, share data and practices across the organization, and produce and access real-time information. A successful ERP can be the backbone of business intelligence for an organization, giving management a unified view of its processes [1].

With the objectives of improved business productivity, streamlined business operations, and increased cost savings, organizations worldwide have launched initiatives to integrate ERP systems into their existing business environments. There has been a growing increase in using ERP systems as a business information system platform for large organizations and government corporations [2].

Due to the capabilities of large ERP system and the essential solutions expected from this system to support the enterprise, its implementation process is complex and risky. It engages a considerable amount of enterprise resources, which are put at risk during

implementation. The managers do not have clear and useful method to direct, effectively and efficiently, the process of implementing an ERP system. They have no guarantee that the system will likely provide the expected benefits [3]. Therefore it is very essential to identify the critical success factors in implementing ERP systems in an enterprise.

2. The Proposed Solution

In order to research into the critical success factors for an ERP implementation and to have a guidance framework to conduct an examination and capture research data, a theoretical framework has been developed to aid the research process. A theoretical framework offers the possibility to group complex issues of investigation together in a more manageable research overview for the researcher [14].

2.1 The Proposed Framework

The proposed framework is based on the existing literature on information systems success, implementation of information systems and ERP systems and previous studies conducted on ERP critical success factors.

The proposed ERP Implementation Success Framework shown in Figure 1 is represented by five broad categories: National context, Organizational context, ERP system context, External Expertise context, and ERP success.

2.2 National Context

National culture influences the way IT is perceived or used. Several authors found proof of this in their studies. All of these studies show a certain impact of national culture in the perception and use of IT [14]. Huang and Palvia [11] identified national factors as: current economic status and economic growth, infrastructure, and government regulations fundamentally impact on IT adoption and ERP penetration. In infrastructures such as transportation, telecommunications, Internet and intranet, mobile telecommunications, and public database systems, developing countries obviously have a poor record and suffer from the consequences [11].

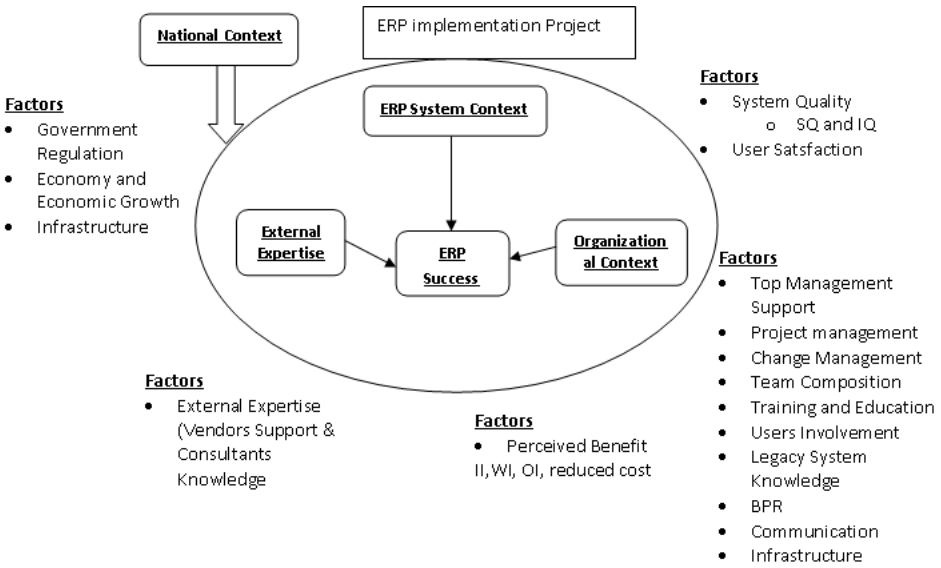


Figure 1: The Proposed ERP Implementation Success Framework

In another study, Abdelghaffar *et al.* [12] like Huang and Palvia [11] identified the national factors which consist of the factors that affect implementing ERP in the country from the macro level. Huang and Palvia [11] argue that the implementation of ERP systems in China - as a developing country - is affected by both national and organizational factors.

In the proposed framework under the national context, the following three factors are included: the level of economy and economic growth, the status of infrastructure like telecommunication, Internet, mobile telephone, and government regulation towards promoting ICT in the country.

2.3 Organizational Context

ERP implementation influences an organization in a number of ways and these changes will be questioned in order to derive CSFs for ERP implementation projects.

From an organizational and internal perspective, Huang and Palvia [11] identified low IT maturity, small firm size, and lack of process management and BPR experience as hampering factors for ERP adoption and they also argue that firms also lack process management orientation and BPR experience. Unlike past computer systems, ERP systems are off-the-shelf and impose their own logic on the company, often forcing companies to change the way they do business. While promising, the actual experience of using IT to redesign business processes is limited in developing countries.

Esteves and Pastor [13] also identified sustained management support, change management, project management, BPR, Project team, and user involvement as strategic factors and consultants' quality, communication, training, and trouble shooting as tactical factors that will affect the implementation of ERP at organization level.

In the proposed framework, the following factors will be included from the organizational context: Sustained Management Support, Change Management, Project Management, Business Process Reengineering (BPR), User Involvement, Communication, Legacy systems, Training and Education, Team Composition, and Infrastructure.

2.4 ERP System Context

An ERP, by nature, is a one-system-only information system that models all the business processes in one, the management of this is seen to be crucial for the success of the organization.

The first factor in the system context is the ERP System Quality (SQ). This factor was identified as critical by many researchers [6, 15, 16, 17]. The questions that this factor will address concerning system quality will be: the functionality of the system, completeness of information, ease of understanding, relevance of the information and how the users' perceive it, and security issue of the ERP system.

The second factor in the system context will be ERP Information Quality (IQ), DeLone and McLean [5] found that researchers had focused on the system's ability to produce the desired information. The ERP information quality concerns the actual information produced by the ERP system.

An ERP system's main selling point is its ability to streamline the information flow in the organization, thus this should be investigated.

The questions that this factor will address concerning information quality will be: the adaptability of the system, its availability to the users, the reliability of the output of the system, and its usability.

2.5 External Expertise Context

When the ERP vendor/consultant provides adequate technical support, has a good relationship with the adopting firm, and so forth, the transfer of relevant information and knowledge to the client is enhanced [8, 15]. It can be argued that the client will be in a better position to use the acquired software efficiently and effectively in achieving organizational goals when such an arrangement exists [15].

Therefore Consultants' Quality in the implementation and in transferring knowledge to employees of the organization and the unreserved Vendor Support provided for the organization will be the two important factors in this context.

2.6 ERP Success Context

Perceived benefits are the most important success measures as they capture the balance of positive and negative impacts of the system in the organization and its customers.

According to Chien and Tsaur [16], the primary benefits expected to result from ERP are closely related to the level of integration that is promoted across functions in an enterprise. Expectations for improved business performance after adoption may result from both operational and strategic benefits [16].

Therefore, Perceived Benefit is the major factor in this context and it will be measured by Individual Impact (II), Work Group Impact (WI), and Organizational Impact (OI).

3. Data Presentation and Analysis

The survey was conducted on managers, consultants, supervisors, IT professionals, key users, and end users from five different organizations which have implemented ERP and or are in the final process of implementing one. A total of 40 questionnaires are distributed. Two of them are sent and returned by e-mail and another two discarded due to dishonesty. As shown in Table 1, survey participants were from construction, communication, government, technology, cement, and manufacturing firms. From the total of 40 questionnaires, 30 were returned and 28 were valid. That means about 70% of the questionnaires sent were received and used for the analysis.

Table 1: Demographic data of Respondents

Company	*Type*	*Industry*	*Questionnaire Delivered*	*Questionnaire Collected*	*Valid*	*%*
Sunshine	Private	Construction	8	6	6	75
ethio telecom	Public	Communication	10	8	8	80
MoFED	Government	Government	10	8	7	70
TCT	Private	Technology	8	6	5	62.5
Mesobo	Private	Cement	2	2	2	100
MIE	Private	Manufacturing	2	0	-	
Total			*40*	*30*	*28*	*70*

3.1 Critical Success Factors Identified

The survey result in Table 2 summarizes the 14 CSF identified for implementing ERP in Ethiopia. These factors were ranked according to their importance valued by their mean value. All the factors have a mean value greater than 3.5 indicating that all the 14 factors are significantly important for implementing ERP in Ethiopia.

Table 2: CSF by their degree of importance

Rank	*CSF*	*N*	*Mean*	*Std. Deviation*
1	Perceived Benefit	28	4.45	0.52
2	Training and Education	28	4.15	0.77
3	User Involvement	28	4.14	0.80
4	BPR	28	4.05	0.54
5	User Satisfaction	28	4.04	0.82
6	Team Composition	28	4.02	0.76
7	System Quality	28	4.02	0.70
8	Communication	28	3.94	0.81
9	Infrastructure	28	3.93	0.77
10	Change Management	28	3.90	0.71
11	Legacy System	28	3.85	0.48
12	Top Management	28	3.82	0.95
13	External expertise	28	3.79	0.96
14	Project Management	28	3.70	0.64

3.2 Reliability of the Survey Instrument

The reliability is measured in cronbach's alpha coefficient method. The croanbatch alpha result for the whole survey (14 items) result shows 0.924.

3.3 Within Case Analysis and Interpretation of Findings

a. Organizational Context

Factors in this context include: Team Composition, Infrastructure, Training and Education, User Involvement, BPR, Top Management Commitment, Change Management, Communication, Legacy System and Project Management. Except Legacy System, Top Management Support and Project management factors in this context are ranked in the top ten CSF for ERP system implementation in Ethiopia. Training and Education, User involvement, and BPR ranked from 2nd to 4th with mean values of 4.15, 4.15, and 4.05 respectively and the other CSF, Team Composition, ranked 6th with a mean value of 4.02. However Change Management, Top Management, and Project Management were ranked 10th, 12th and 14th with mean values of 3.90, 3.82, and 3.70 respectively.

b. ERP System Context

System Quality and User Satisfaction are the two main CSF identified for ERP system implementation in Ethiopia. System Quality ranked 7th with a mean value of 4.02 and User Satisfaction ranked 5th with a mean value of 4.04.

c. External Expertise Context

Factors in this context were vendor support and consultant quality. These two are external expertise that are identified as CSF for ERP implementation. This two factors are merged into one and ranked 13th with a mean value of 3.79.

d. ERP Success Context

Perceived Benefit in terms of performance improvement, cost reduction, and time saving is the most CSF in Ethiopia. With a mean value of 4.45, it ranked the 1st CSF in ERP system implementation in Ethiopia.

3.4 Cross Case Analysis and Interpretation of Findings

This Section draws comparison from the case study organizations with the aim of identifying similarities and differences in CSF identified in ERP system implementation. The analysis was based on the framework designed in section 3. Figure 2 shows the relative importance of CSFs in the study case organizations. Perceived Benefit in terms of performance improvement operating cost reduction, and time saving is the most significant factor in all the three case organizations and the other CSFs that rate relatively similarly (equal rank) are user involvement, change management, communication, and project management. However, high variance is seen in top management and team compositions.

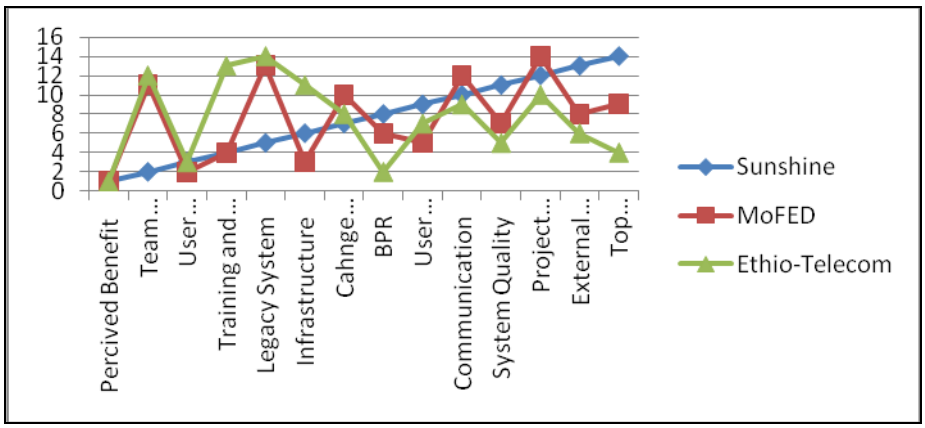


Figure 2: Rank of CSF by case organization

When comparing the similarity and variation of CSF in the two case organizations, MoFED and ethio telecom, most of the CSFs are not as dispersed as when comparing Sunshine with one of the other two.

One thing this companies share is the ownership of the companies, i.e., MoFED is a government organ and ethio telecom is a government owned company. The two companies ranked government commitment and support higher than Sunshine Construction Company which is a private owned company. Another similarity between state owned companies is seen in team composition. These two companies established a separate project office for ERP implementation while Sunshine has not established a separate project office for its ERP project.

Yet another interesting result was seen in legacy system. Both ethio telecom and MoFED have many legacy systems they have been using and expected to be replaced by the new system and other systems that need interfacing to the new system, ranked legacy system knowledge below 10th while Sunshine, without any centralized legacy system that will be abandoned and no interfacing to any other system, ranked legacy system knowledge in the 5th position.

The other finding is Training and Education. Sunshine and MoFED ranked training and education on 4th position while ethio telecom ranked it 13th. Here again ethio telecom is highly experienced in implementing technology related systems that need intensive trainings, and this experience differentiates it from the other two companies. When we see the degree of importance of BPR, ethio telecom ranked it on 2nd most important and MoFED ranked it on 6th and Sunshine on 8th place. The case organizations have made different levels of structural changes in organizations before the implementation and or during the implementation. ethio telecom made total organizational restructuring, MoFED made BPR as part of performance improvement plan of the government while Sunshine made a small process changes during the implementation.

4. Related Work

ERP software packages provide enterprise-wide generic solutions to many organizations. An enterprise system imposes its own logic on a company's strategy, culture, and organization [4].

Rajapakse and Seddon [7] showed that fewer than 7% of ERP implementations worldwide have been in developing countries (in Asia where they have done their studies). Organizations in the developing-country context pursue ERP systems for similar reasons to organizations in the developed countries: mainly to support their growth beyond what their previous in-house developed systems all owed, and to stay competitive (and link strategically) with other organizations globally [11, 12].

In their study, Abdelghaffar *et al.* [12] claim adopting ERP in developing countries is faced by several obstacles compared with developed countries. The challenges faced by developing countries vary on organizational level and cultural factors and lack of knowledge of ERP systems

Abdelghaffar *et al.* [12] studied significant factors influencing ERP implementation in large organizations in Egypt. They have adopted Huang and Palvia's framework to assess which factors have more significant impact on the success of ERP implementation in Egypt and they come up with their own modified framework as shown in Figure 3 [12].

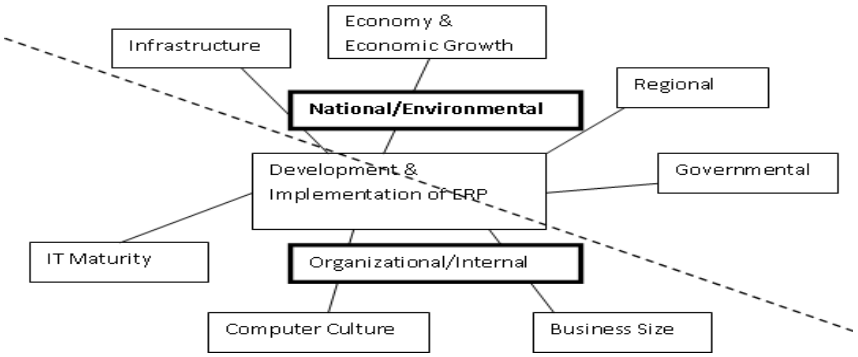


Figure 3: ERP Implementation Framework

They have identified 7 CSF for implementing ERP in large organizations in Egypt. Under the National/Environmental factors: Infrastructure, Economy and Economic Growth, and Regional and

Governmental factors are more significant and under the organizational or internal factors they have identified IT maturity, Business size, and Computer Culture.

Due to low success rate of ERP system implementation in China, Zhang *et al.* [17] developed an ERP implementation success framework by adapting the DeLone and McLean's [5] IS success model to identify both CSFs and success measures (Figure 4). From the research framework, they have identified 12 CSFs: Top Management Support, Company-wide Support, Business Process Reengineering, Effective Project Management, Organizational Culture, Education and Training, User Involvement, User Characteristics, ERP Software Quality, Information Quality, System Quality and the success measures were User Satisfaction, Individual Impact, Organizational Impact, and Intended Business Performance Improvement.

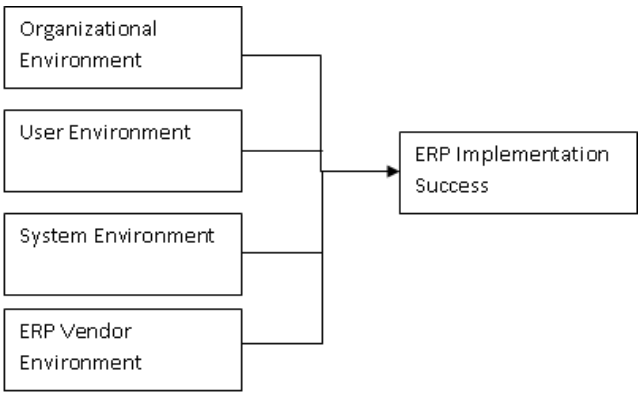


Figure 4: ERP implementation success framework

Yihe Zi [18] explored the differences in CSFs for ERP system implementation stages in China and Finland as a representative of the western countries. Previous research on the subject matter was thoroughly explored to form a solid basis for the empirical study. Yihe Zi used Shanks *et al.*'s [10] synthesized ERP project lifecycle as an ERP implementation process model to show which factor is more important at each of the implementation stages.

The major CSFs identified as important are: Top management support, ERP teamwork and composition, Project management, Clear goals and objectives, Change management, User training and education, Business Process Reengineering and minimal customization, Presence of a champion, Use of experts and consultants, Interdepartmental communication and cooperation, Master data quality, and Performance measure.

In their Comparative Study of Critical Success Factors (CSFs) in Implementation of ERP in developed and developing countries, Moohebat *et al.* [9] studied 52 researches identified as a representative of developed countries and 33 as developing countries. Among these researches, 14 developed and 16 different developing countries are identified. USA with 25 and China with 11 researches had most resources in this study, respectively between developed and developing nations.

According to Moohebat *et al.* [9], differences exist between ERP critical success factors in developed and developing countries as shown in Table 3 [9]. Few factors like "ERP vendor" and "National culture" seem more important for developing countries and some factors like "Business process reengineering" and "Fit between ERP and business/process" differs among developed and developing countries.

Table 3: CSF in Developed and Developing Countries

Critical Success Factors	*Developed Countries*	*Developing Countries*
Appropriate business and IT legacy systems	6	6
Business plan/vision/goals/justification	13	14
Business Process Reengineering	31	19
Change management	35	30
Communication	20	19
Country-related functional requirements	3	3
Data accuracy	9	8
ERP project team	31	24
ERP strategy and Implementation	24	17
ERP Vendor	8	13
Fit between ERP and business/process	9	1
Monitoring and evaluation of performance	9	7
National culture	4	10
Organizational characteristics	7	6
Project champion	13	6
Project management	31	26
Software Development, testing, and troubleshooting	13	6
Top management support	28	29

The framework in [12] has not recognized the impact of external expertise on ERP implementation in countries where ERP implementation is new and all the expertise on the field will be from outside the organization and the ability to transfer knowledge is essential to the success of ERP in the companies. Whereas Zhang *et al.*'s framework doesn't consider the national context in which the implementation is done. It is evident from other researches such as in [12] implementation in developing countries is highly affected by the national factors. Moohebat *et al*. studied CSF for ERP implementation in developed and developing countries by studying from thorough literature review and Yihe Zi studied the CSF in china against Finland. These studies will be used as a basis for this study.

5. Conclusion

The implementation of ERP systems in organizations is an enormously complex undertaking. It is a high-risk project that needs to be managed and planned properly because it can affect nearly every aspect of organizational performance and functioning.

Many organizations do not achieve success in their ERP implementation projects. The aim of this research was to explore CSF for ERP implementation in Ethiopia and how they affect the organizations. This study is valuable to researchers and practitioners interested in implementing ERP systems in Ethiopia. It provides interesting results by identifying the factors that actually have an impact on the successful implementation of ERP and how it affects organizations. Implementing managers can use the result of this study as input for planning ERP system projects.

To ensure successful implementation, organizations must learn how to identify the critical issues that affect the implementation process and know how to address them effectively to ensure that the promised benefits can be realized and potential failures can be avoided.

References

[1] Janice Yick, "Implementing Vanilla ERP Systems: Factors to Consider in Strategy, Business Alignment, and Customization", University of Oregon Applied Information Management Program, 2011.

[2] Jim Odhiambo Otieno, "Enterprise Resource Planning Systems Implementation and Upgrade: A Kenyan Study", School of Engineering and Information Sciences, Middlesex University, 2010.

[3] Noe Garcia Sanchez, Luis E, and Perez-Bernal, "Determination of Critical Success Factors in Implementing an ERP System: A Field Studying Mexican Enterprise", Journal of Information Technology for Development, 2007.

[4] Davenport, T. H., "Putting the Enterprise into the Enterprise System", Harvard Business Review, July-August, 1998.

[5] Delone, W. H. and Mclean, E. R, "Information Systems Success: The Quest for the Dependent Variable", Information Systems Research, Vol. 3, No. 1, 1992.

[6] Rockart, J. F, "Chief Executives Define their own Data Needs", Harvard Business Review, 1979.

[7] Rajapakse, J. & Seddon, P., "ERP Adoption in Developing Countries in Asia: A Cultural Misfit", 28th Information Systems Seminar in Scandinavia, 6-9 August, Kirstiansand, Norway, 2005.

[8] Markus, M. L. and Tanis, C., "The Enterprise Systems Experience - From Adoption to Success, Framing the Domains of IT Research: Glimpsing the Future Through the Past", Ed. Zmud, R. W., Pinnaflex Educational Resources, Inc, Cincinnati, 2000.

[9] Mohammad Reza Moohebat, Asefeh Asemi, and Mohammad Davarpanah Jazi, "A Comparative Study of Critical Success Factors (CSFs) in Implementation of ERP in Developed and Developing Countries", International Journal of Advancements in Computing Technology Vol. 2, No. 5, December 2010.

[10] Shanks, G., Parr, A., Hu, B., Corbitt, B., Thanasankit, T., and Seddon, P., "Differences in Critical Success Factors in ERP Systems Implementation in Australia and China: A Cultural Analysis, Proceeding of European Conference on Information Systems", Vienna, July 2000.

[11] Huang, Z., & Palvia, P., "ERP Implementation Issues in Advanced and Developing Countries", Business Process Management Journal, Vol. 7, No.3, 2001.

[12] Hany Abdelghaffar, Reem Hamdy, and Abdel Azim, "Significant Factors Influencing ERP Implementation in Large Organizations: Evidence From Egypt, European, Mediterranean, and Middle Eastern Conference on Information Systems, Abu Dhabi, UAE, April, 2010.

[13] Esteves, J. and Pastor, J., "Enterprise Resource Planning Systems Research: An Annotated Bibliography", Communications of the Association for Information Systems, Vol. 7, No. 8, August, 2001.

[14] J. Gilbert Silvius, "The Impact of National Cultures on Business & IT Alignment", Communications of the IIMA, Vol. 8, Issue 2, 2008.

[15] Princely Ifinedo, "Enterprise Systems Success Measurement Model: A Preliminary Study", Journal of Information Technology Management, Vol. XVII, No. 1, 2006.

[16] Shih-Wen Chien and Shu-Ming Tsaur, "Investigating the Success of ERP Systems: Case Studies in Three Taiwanese High-Tech Industries", National Pingtung Institute of Commerce, Department of Commerce Automatic & Management, Department of Management Information, Ching Yun University, Jhongli, Taiwan, 23 March 2007.

[17] Zhang, Z., Lee, M.K.O., Zhang, L., and Huang, X, "A framework of ERP systems implementation success in China: An Empirical Study", International Journal Production Economics, Vol. 98. Issue 1, 2005.

[18] Yizi He, "A Comparative Study of Critical Success Factors for ERP System Implementation in China and Finland", Swedish School of Economics and Business Administration, 2007.

Domain Knowledge Based Requirements Change Management Framework: Case of Airlines Systems

Almaz Tsegaye
Ethiopian Airlines, Addis Ababa, Ethiopia
almazt96@gmail.com

Mesfin Kifle
Department of Computer Science, Addis Ababa University, Ethiopia
kiflemestir95@gmail.com

Abstract

Requirements change management is becoming a challenging task since changes evolve frequently due to the dynamic situations in business and governance. Changes in business and governance affect the business context and the way systems manage. Traditionally, business and software system experts work together to analyze the impact of change on the existing systems. In other words, the change enforces experts to maintain their existing domain knowledge.

In this paper, a domain knowledge based framework is intended to capture the basic knowledge as well as the accumulated experiences of an enterprise to facilitate requirements change management. During the research, a survey was conducted and analysis was done to get a clue to solve the research questions such as "What kind of framework is appropriate to represent existing Domain Knowledge and for Requirement Change Management?" and "How do we design and validate the framework?"

A framework for requirement change management is proposed for the case of Airline Systems. Based on knowledge types, their properties and relationships, the impact analysis algorithm is formulated. In order to validate and verify the framework, a case study was conducted on Airline Revenue Accounting and a prototype was developed for framework verification.

Keywords: Domain Knowledge; Domain Knowledge Representation; Framework; Requirements Change Management; Requirement Management

1. Introduction

The primary measure of success of a software system is the level to which it meets the purpose for which it was projected, and RE (Requirements Engineering) is the process of discovering that purpose. The RE process commonly includes five phases: Requirements Elicitation, Requirements Analysis and Negotiation, Requirements Documentation, Requirements Verification and Validation, and Requirements Management [1]. As presented in [1], the exposition of a set of assumptions, concepts, values, and practices that constitutes a way of understanding the research within a body of knowledge is called a framework.

Domain Knowledge change is inevitable in Airlines industry due to various factors. The research objective was to study the existing Requirements Change Management process, reviewing concepts about Domain Knowledge base Requirement Change Management, identifying Domain Knowledge that can be applied for Requirement Change management, analyzing, modeling and development of technique and tools to validate the framework.

In order to study the existing Requirements Change Management process; interview was conducted with Domain and IT Experts. To complement the findings obtained during interview, work flow observation and secondary sources of information, such as documents, policies, and manuals have been assessed. Finally, framework was proposed, prototype for framework verification was developed and case study on Airline Revenue Accounting System to verify the framework was done.

The remaining part of the paper is organized as follows: Section two briefly explains information related to the research. Section three deals with the proposed framework design and modeling. In Section four, experimentation and results are presented.

Section five deals with related work to put conceptual foundation, and finally Section six presents the conclusion.

2. Background

Requirement Management involves: controlling changes to the requirements baseline, keeping project plans current with the requirements, controlling versions of both individual requirements and requirements documents, tracking the status of the requirements in the baseline, and managing the logical links between individual requirements and other project work products [2].

Once the initial requirements for a body of work are in hand, we must cope with the inevitable changes that customers, managers, the development team, and others request. Effective change management demands a process for proposing changes and evaluating their potential cost and impact on the project. Requirements change is a reality for all software projects, but disciplined change-management practices can reduce the disruption that changes can cause [2].

As explained in [2], impact analysis is a key aspect of responsible requirement management. The analysis examines the proposed change to identify components that might have to be created, modified, or discarded and to estimate the effort associated with implementing the change. Before a developer says, "Sure, no problem" in response to a change request, s/he should spend a little time on impact analysis. It provides better understanding of the implications of a proposed change, which helps to make informed business decisions.

Therefore, in order to implement change impact analysis solution, domain knowledge based change management helps to easily trace change dependency and to make informed decision to accept or reject the change, based on change impact analysis. Change impact analysis is identifying the potential consequences of a change, or estimating what needs to be modified to accomplish a change [3].

Domain knowledge is a form of knowledge which is used to refer to the knowledge possessed by the people about certain business processes in an organization [4]. The authors in [5] stated that domain knowledge is acknowledged to be important for the acquisition and validation of requirements specification. Unfortunately, requirements validation and acquisition is one of the most error prone and costly activities in system development which has led a number of researchers to investigate how the process of requirements capture and validation may be improved. Previous approaches have varied from development of powerful specification language, to investigation of declarative, rule based paradigms and embedding domain knowledge in CASE (Computer Aided System Engineering) tools to allow intelligent requirements validation. A possible solution to the Domain Knowledge acquisition dilemma is to use more generic Domain Knowledge.

3. Framework Design and Modeling

According to [6], framework is a hypothetical description of a complex entity or process. Framework is one of the possible ways for finding solutions to problems. Frameworks are commonly used to combine the techniques on a specific problem, topic or area [1].

In order to investigate the necessity of Domain Knowledge based Requirement Change Management Framework, a survey was conducted which intends to assess the current Requirement Change management process for Airlines systems. The survey result revealed the existence of requirement change management problem. This section presents the proposed Domain Knowledge based Requirement Change Management Framework and Domain Knowledge Model. The proposed framework consists of domain knowledge maintenance and change impact analysis. The elements of the Domain Knowledge are rule, processes or tasks and system. The proposed framework is depicted in Figure 1.

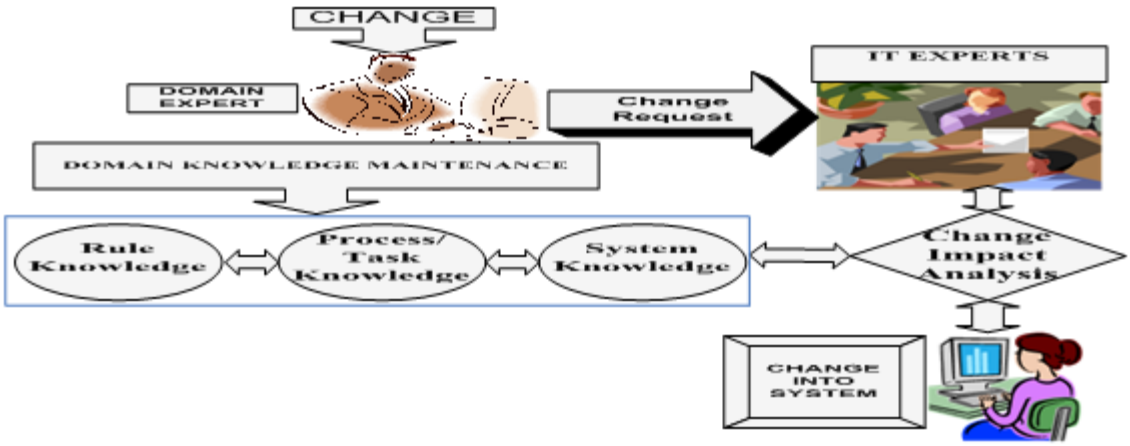


Figure 1: Domain Knowledge based Requirement Management Framework

When the change happens due to industry regulation change or any business area requirement, first the Domain Expert will do analysis work based on change source information. The analysis could be like studying the new change implication from the source of the requirement, checking the existence of new knowledge due to the change in order to add new knowledge into Domain Knowledge. The change may happen to already existing Domain Knowledge which may require removal or update of existing knowledge. The proposed framework contains change request submission, knowledge maintenance, and change impact analysis. Domain Knowledge or an update to existing knowledge will be decided by the domain expert and later the IT expert will involve on software change impact analysis and will prepare the impact analysis report for Management decision. If required, s/he can query the existing knowledge to check dependency of the new change on existing knowledge or previous requirement. The domain expert can do at high level the impact analysis of the change on the existing system. Then the domain expert can forward the requirement to the IT section.

The IT Expert further studies the impact and dependence of the new requirement in detail, but this time most of the task is already done by the Domain Expert. Hence, the IT expert can manage the change timely and with better quality. The IT Expert identifies Airline systems which will be impacted due to regulation change or any change request. The new framework will also reduce rework on change request, because the domain expert will have a role to play in the new change request rather than just forwarding the request to the IT section. The Domain Experts will participate on impact analysis which will help them to understand the volume of work required by the IT section. It will also help for knowledge reuse and for other new software development.

The knowledge representation model of the system, tasks, processes, and rules are shown in Figure 2. The model is done using frame based knowledge representation. The model shows the dependency among four knowledge types. The tasks which exist in the Airline domain are highly dependent on rules and regulations of the organization itself and other regulatory bodies. For instance, all flight operations and ticket sales depend on each country's and regulatory bodies' rules. Accordingly, processes are also guided by rules indirectly, since the task(s) under each process are done based on rule. On the other hand, systems, subsystems and modules are highly interrelated; one is the subset of the other. In addition, a process depends on the system, tasks and modules have relations as they are subsets of a system.

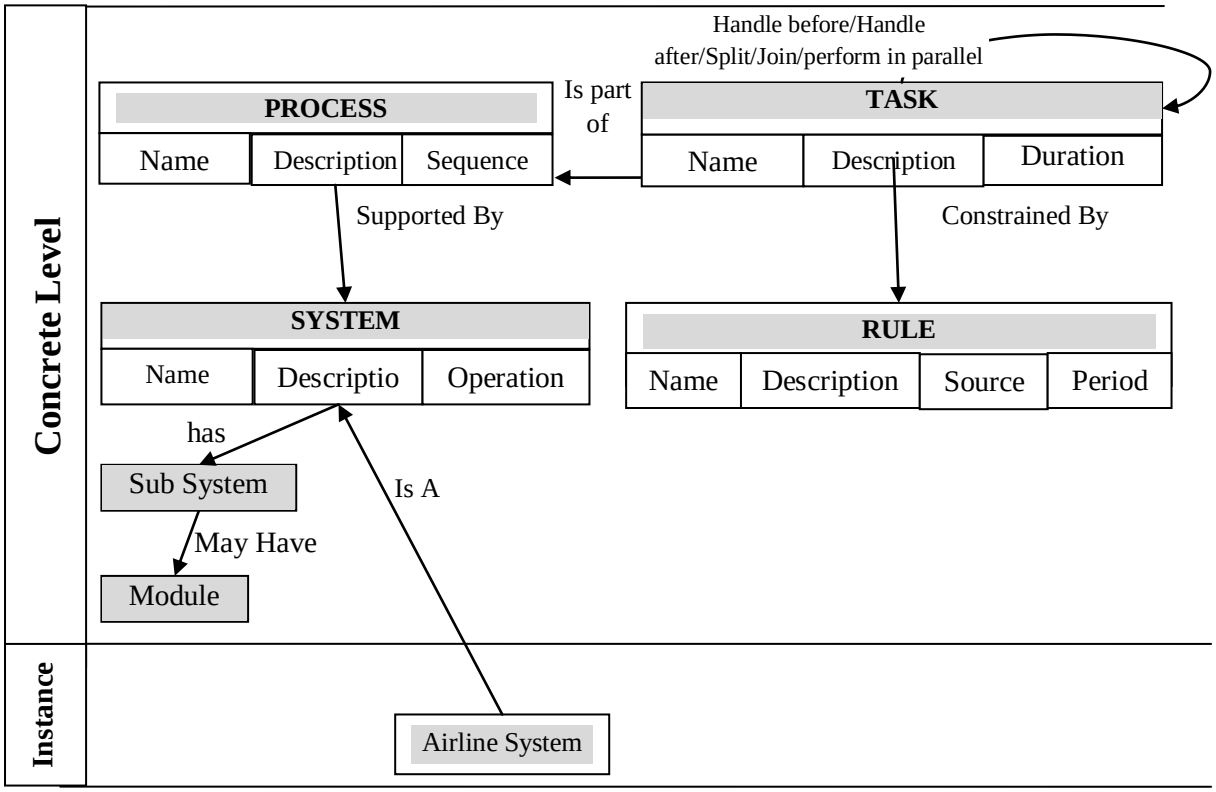


Figure 2: Domain Knowledge Representation Model

4. Experimentation and Results

The aim of the experimentation is to demonstrate the output of the research and to verify and validate the proposed framework. In order to verify our framework with practical example, Airlines Revenue Accounting was chosen as a case study. The reason we choose Revenue Accounting was because it encompasses all the knowledge types (process, task, system and rule/regulatory bodies and company rules) which are in the framework. As per the researchers' knowledge, there are a lot of new change request and requirement change due to various reasons.

To conduct the case study, it is necessary to have a system to test the framework and the model. A Domain Knowledge and change request repository RDBMS (Relational Database Management System) was chosen, which is Microsoft Access. The programming language used for implementation was C#. The architecture of the prototype has a client side and at backend database for data storage.

In the framework, the Domain Knowledge is classified into three main areas. They are rule, process or task, and system knowledge. The technique used to utilize the Domain Knowledge was by storing the knowledge in a database to share and apply for change impact analysis.

The prototype designed had a capability of change request and domain knowledge maintenance and the change impact analysis report extraction. There are user interfaces which have a capability of record maintenance for change request, rule, task, process, system, and module. In addition to the data maintenance operations, it has the capability of searching the related or impacted knowledge from process, task, rule and system tables. For the case study sample, previous change request data has been taken. When the change occurs, the domain expert searches for change request record to check if the change already exists by the search request command using as a parameter change request keyword. The keyword is more related to domain knowledge classification or terms in the domain like, for example, the request if it is related ticket sales data entry system or module. The domain expert may use as a keyword like "Ticket" and search any request which exists previously by this same keyword. If it exists, the domain expert will ignore the data entry for change request. But s/he can check the status of the request whether it is active or not. If it is active,

s/he will follow up the status with his/her section or in the IT section. The actual status of the request may be waiting to change request approval, or it is being developed in the IT section. If the request is new, s/he will add the request into the system, search for change impact and the request is next handled by an IT expert after proper change request submission to the IT section in the form of a letter or e-mail. Then the IT experts will do additional change impact analysis using impact search command which exists on the user interface.

The user interface will handle change request and knowledge insertion, deletion, update and search. The screen could be used both by Domain Experts and IT experts to extract impact analysis. For the IT expert, the same user interface could be used for change impact analysis.

5. Related Work

The authors in [7] designed a framework and architecture supporting the utilization of Domain Knowledge embedded in software. The framework is based on identifying Domain Knowledge acquired during the development process (especially in requirements analysis) and formalizing it. They have identified seven fundamental concepts commonly used in modeling application domains: Actors, Roles, Resources, Services, Goals, Constraints, and States (Consisting of Attributes). They verified their framework by implementing a system and by classifying the system architecture into two parts: one represents the Domain Knowledge and the other is responsible for the actual processing (using this knowledge). Their work focused on explicitly extracting Domain Knowledge which is embedded within the system. It does not include the additional Domain Knowledge which exists external to the system like within Domain Expert's mind and in the various documents. The research work could be used to utilize part of the Domain Knowledge and the knowledge which exists within the system. Addressing the entire domain knowledge representation is advisable if a new requirement is requested which we can use the existing knowledge for requirement management. If we identify and represent the Domain Knowledge right from the origin, it will be easy to utilize during requirements management. The use of Domain Knowledge which exists in software itself may not be enough for Requirements change management. The entire Domain Knowledge could not be found in existing software. Our research recommendation is to represent Domain knowledge which exists within the mind of Domain Experts, in policies and procedures, and systems by storing in separation to the system itself so that knowledge maintenance can be easy.

The research work in [8] contributed a method and framework for Domain Knowledge assisted requirements evolution. Their platform facilitates requirements elicitation for a team which are geographically in different areas but are working as a team for the same project. This work can be referred to as a baseline that can be applied in requirements change management process perspective. What the researchers learned considering domain knowledge as "seeds" of requirements can potentially be a new viewpoint in Requirements Engineering. Their framework design and the way they apply the domain knowledge to fulfill requirement elicitation parameters like completeness, consistency, ambiguity, and correctness is well taken. But the objective of our research work was to utilize domain knowledge as a means to extract change impact analysis.

The framework in [9] helps to integrate requirements management and design knowledge reuse. The framework is then demonstrated using a case study example: vacuum pump design. Requirements are presented as a component of the integrated design knowledge framework. The proposed framework enables the application of requirements management as a dynamic process, including capture, analysis and recording of requirements. It takes account of the evolving requirements and the dynamic nature of the interaction between requirements and product structure through the various stages of product development.

6. Conclusion

Recently, the business environment is dynamically changing which in turn affects existing software systems. As a result, the business

knowledge and system requirements will change. Requirements change management is focused to conduct change impact analysis. In order to study this problem, a survey was conducted which intends to assess the current Requirement Change management process for Airlines systems. The survey result revealed the existence of requirement change management problem. Therefore, in order to solve the existing Requirement Change Management problem, this paper proposed a Domain Knowledge based Requirement Change Management Framework. The proposed framework consists of domain knowledge maintenance and change impact analysis. Experimentation is done to demonstrate the output of the research, to verify and validate the proposed framework. In order to verify the framework with practical example, Airlines Revenue Accounting Domain was chosen as a case study.

References

[1] M. Husnain, M. Waseem, and S. A. K. Ghayyur, "An Interrogative Review of Requirement Engineering Frameworks," *International Journal of Reviews in Computing*, 2009.

[2] Karl E. Wiegers, *Software Requirements, Second Edition*. 2003, pp. 516.

[3] "Change Impact Analysis," 2012. http://en.wikipedia.org/wiki/Change_impact_analysis. [Accessed: 10-Mar-2012].

[4] "Domain Knowledge." http://www.freetechexams.com/computers-tips/computer-tips/domain-knowledge.html. [Accessed: Aug-2011].

[5] A. G. Sutcliffe and N. A. M. Maiden, "Use of Domain Knowledge for Requirements Validation," in *Proceedings of IFIP WG8.1 Conference on information System Development Process, Como Italy*, 1993, no. September, pp. 1-3.

[6] "Freedictionary." http://www.thefreedictionary.com/framework. [Accessed: 26-Apr-2012].

[7] E. Rubin and Y. Wand, "A Framework Supporting the Utilization of Domain Knowledge Embedded in Software," *Reproduction*, vol. 83, 2007.

[8] P. Rose, M. Kumar, N. Ajmeri, M. Agrawal, V. Sivakumar, and S. Ghaisas, "A Method and Framework for Domain Knowledge Assisted Requirements Evolution (K-RE)," in *Proceedings of CONSEG-09: International Conference on Software Engineering Dec 17-19, Chennai, India*, pp. 87-97.

[9] D. Baxter and et al., "A framework to integrate design knowledge reuse and requirements management in engineering design," *Robotics and Computer Integrated Manufacturing Systems*, pp. 1-9, 2007.

Achieving Organizational Interoperability for Agricultural Research with Cross Organizational Business Process Modeling

Amanuel Tekalegn
Africa Union Commission, Addis Ababa, Ethiopia
eng.amantk@gmail.com

Fantahun Abera
PhD Candidate, Addis Ababa University, Ethiopia
fanetsa8@yahoo.com

Abstract

A layered approach to interoperability between multiple e-systems (e-government, e-health, etc.) has been considered by both the industry and academia; long been exercised, with organizational, semantic, syntactic and technical interoperability identified as the possible layers. While practical solutions have been discovered and well tested for the lower three layers, the development and utilization of interoperable information systems is still held back by different issues of organizational interoperability layer, and remained to be a challenge in the software engineering discipline. As such, issues pertaining to organizational layer in the interoperability spectrum have been actively investigated and endeavors to come up with a holistic solution are underway. Recent works in this area promote the use of business process standardization and ensuring legal and strategic interoperability across organizations for organizational layer problems.

In this paper, business process modeling across multiple organizations is investigated within different governmental institutions taking part in agricultural research activities in Ethiopia.

The research takes a form of investigative approach and constructing a business process network model between participating institutions, followed by establishing organizational matrix setup and formulate governance frameworks to ensure organizational interoperability. The result of the research showed the possibility of business process modeling across multiple collaborating institutions as long as they share common goals and strategies. Business Process Modeling Notation (BPMN) is used for depicting the business process choreography between institutions. Lastly, the modeled business process, connecting multiple institutions, is used to model a SOA based solution using Web services.

Keywords: Organizational Interoperability; Cross organizational Business Process; BPMN; SOA

1. Introduction

Different layers of interoperability have been identified at different times. The European Interoperability Framework (EIF) classifies interoperability into three layers: Technical, Semantic, and Organizational with reference to three dimensions of organizational interoperability insights from recent studies for improving interoperability frame-works in [1]. The European Standards Institute added a syntactic layer between the technical and semantic layers of EIF [1]. Other researchers and standards bodies considered the mentioned four layers of interoperability by adding legal, strategic, governance, and crosscutting issues into the frameworks [1].

The organizational interoperability layer, which is a crucial component of the interoperability requirements, is the one with less common understanding. The gap emerged from the specific concerns of each of the groups and lack of holistic views to the problem. In an effort to improve the existing interoperability frameworks, the authors in [1] suggested the redefinition of the presently called organizational interoperability to be limited to business process interoperability standards and concepts concerned with the linkage of business processes between collaborating institutions.

In order to align activities and support decision making at different levels, considering enterprise architectures is important for supporting interoperability on cross organizational business process. The enterprise architecture will help organizing mechanisms to structure concepts. There are different enterprise architecture frameworks in order to support modeling business process [2]. In

this paper it is tired to consider different enterprise architectures as solution for organizational interoperability with different issues raised while implementing.

2. Background

Governments must avail services for citizens and to other stakeholders. As government institutions, they must share and use information about their common customers in order to work together. To achieve the requirements of service delivery, institutions need to exchange information about service provision, the common customers and the processes. This last requirement was termed interoperability by practitioners and researchers in the field. Interoperability between two or more institutions can be described as the ability of the participants to share relevant information in a timely manner and use the shared information to effectively and efficiently work together [3, 4]. Interoperability between institutions can be achieved with common understanding among different institutions working together. That means institutions must have rules, regulations, and governance in order to work together. Since working together means sharing information based on the specified rules and regulations stated in the governance, those institutions must act accordingly. Information sharing alone will not result in an effective and efficient performance. In order to successfully implement interoperability, the institutions participating in the provision of services crossing organizational boundaries must agree with each other on the level of their collaboration. For the institutions to use the shared information, it must be clearly understandable and support the service provision [5].

3. Proposed Solution

In order to achieve organizational interoperability, different institutions that are working together for shared services must have common understanding on the service shared and also must sign service level agreement on which each institution has responsibility for the continuity of cross boundary services. Interoperability is central to any form of collaboration between organizations, as it enables information and knowledge sharing by cooperating entities within and across organizational boundaries [6]. Interoperability is important for different stakeholders that use the services from public sectors for effectively working together on cross boundary services. Different enterprise architectures are found that comprise models and tools that can support and develop IT applications within an enterprise [6]. Adapting Enterprise Architecture must undertake rigorous study on the business and IT governance of the business process in order to achieve interoperability. As a solution and before adapting Enterprise Architecture, considering cross organizational business process model must be undertaken.

In order to reach to a solution, this paper proposes modeling cross boundary business processes and assign mandates to different institutions that are working with cross boundary institutions. Accordingly, there must be formulation of an organizational matrix setup on different institution work performers to give responsibility for the established organizational setup in order to handle the modeled and implemented cross boundary services within different agricultural institutions. In this regard, setting up new organizational structure to handle the modeled cross organization business processes is very critical. The Ethiopian Government has developed a growth and transformation plan on the agriculture and rural development sector for formulation of need based policies and strategies and capacitating the human resource skill so as to use the human resource in different agriculture sectors. Using land resources in efficient and effective way, developing different agriculture ecological products within different zones for developing the regional capacity, entrenching specialization and diversification, coordinating agricultural and rural development proceedings, and strengthening the agriculture strategy and development in depth are also central to the growth and transformation plan.

In order to effectively and efficiently deliver agricultural research services, the role of the designed business process model is important in addressing cross boundary services within different livestock agricultural research institutions for shared services. Therefore, the establishment of

organizational matrix setup is considered below in order to achieve the following advantages:

- Enhance collaboration and coordination of work performance,
- Strictly follow the performance of each cross boundary service for agricultural research institutions performance on shared services,
- Create accountability and transparency for cross boundary services and clearly show inactive work performers with different institution, and
- Appropriate and on time decision making for cross boundary business process for shared service.

4. Experimentation/Prototype

The business process network model diagram shown in Figure 1 illustrates the entire business process for agriculture research on livestock service delivery. The diagram can be elaborated as follows: Research problems will be identified by different researchers working in research institutions. In order to undertake data collection, researchers from EIAR and MOA will communicate the regional, zonal and woreda bureaus. The collected data will be analyzed and recommendation will be generated in order to undertake research. Different researchers will draft their proposals based on the gathered data from zonal and woreda levels. At regional level there will be meetings on prioritizing the proposals that can cover the regional current issues with best solution. Once the regional, zonal and woreda levels have decided to approve proposals, the regional bureau will send the prioritized proposals to federal research institutions. In federal research institutions, there will also be a meeting on selecting the current issues in different regions with scientific justification and cost analysis breakdown. In the meeting, researchers from international livestock researcher institutions and federal and regional institutions will meet and make a decision on selection of the different regions problem solving. After a presentation of each regional researchers and the committee (research directors) review on proposal paper acceptance and rejection of research papers will be made. The approved proposal will be appended by a financial budget to undertake the research. Each region will present the approved proposals with the financial implications for MoFED. MoFED will approve the research budget in each region. Once the research budget has been approved, the research director will communicate each research institution that is implementing the research proposal. Research will be conducted and for verification of technologies different stakeholders will be communicated. Researchers from MOA, EIAR, and RARI will be invited and approve the technology. Once the technology is tested in both the laboratory and the field and the technology is approved, a request letter will be provided for MOA. MOA is responsible for information sharing and technology dissemination.

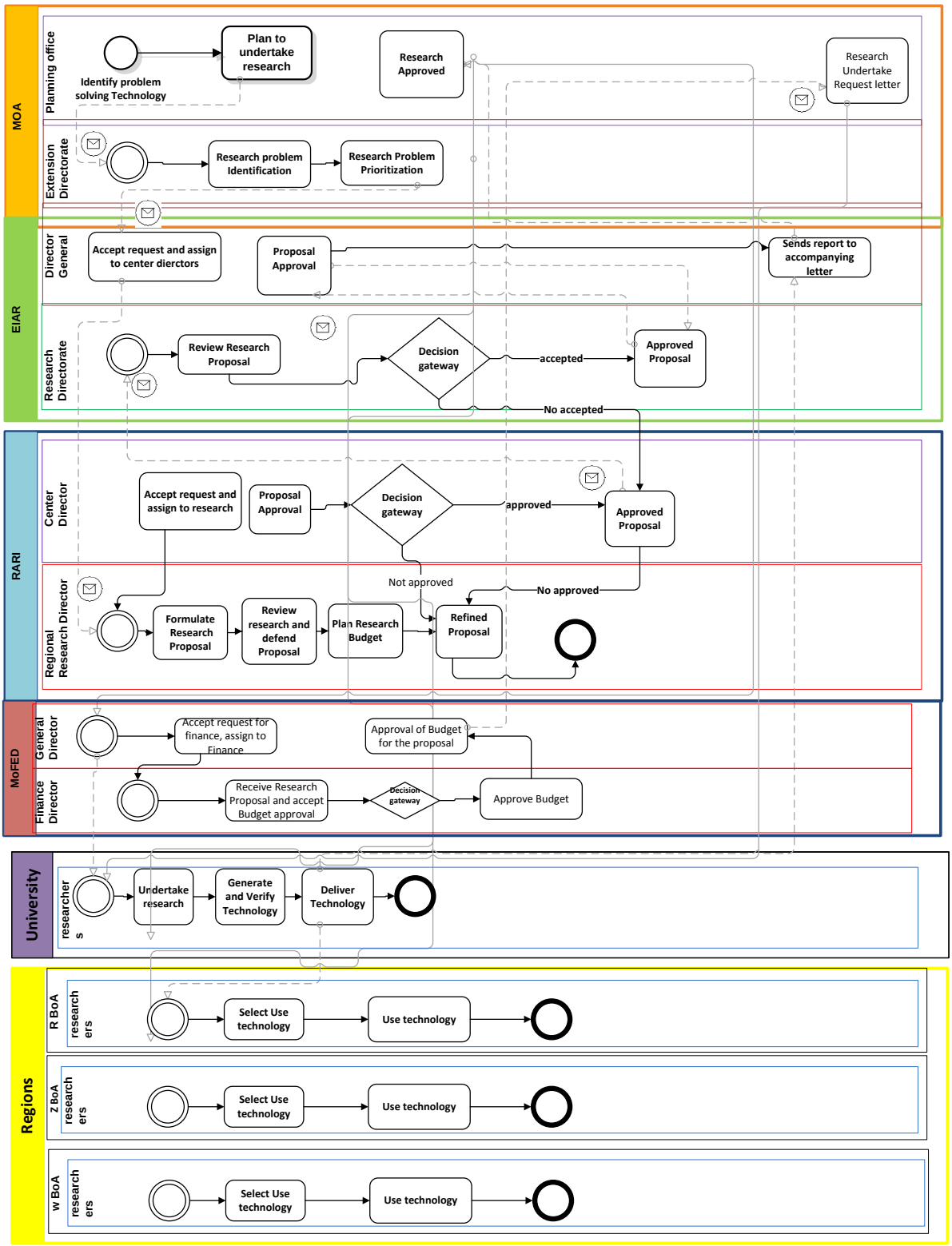


Figure 1: Overall livestock service delivery business process

5. Related Work

Government institutions need to collaborate in order to improve citizen service delivery. Institutions need to work across portfolio boundaries to find collaborative, networked and multi-communication channel approaches to deliver information and services from business to citizens [2]. Development of an interoperable framework provides an integrated service by articulating a set of agreed policies, principles, and standards to facilitate information and transactions from government to all stakeholders. Improving the capability of cross institutions means

communication is confidently managed, transferred and an exchange of information and data [7].

The World Bank defined e-government as the use by government agencies of information technologies that have the ability to transform relations with citizen business and other arm of the government. E-government simply means the use of ICT to improve the activities of public or private sector institutions [8].

The technologies enable governments in various ways including effective and efficient service delivery of government services to citizens, improving interactions between institutions, and sufficient information access that will result in more efficient and effective government management [8]. E-government encompasses all ICT technologies but the key motive is to create strong connection between government sectors. The connectivity and collaboration project will strengthen accountability, connections between government and business/citizens transforming service delivery [1]. Generally, e-government promises to make government more efficient, responsive, transparent, and legitimate and is also creating a rapidly growing market of goods and services, with a variety of new business opportunities [9].

It has been a while since the implementation of e-government started in Africa. There are growing numbers of e-government projects, some of which are contributing to public sector reform and delivering gains of efficiency and or effectiveness across a broad agenda. However, this picture must be set alongside significant challenges.

In order to achieve interoperability, institutions must communicate effectively. For service consumers to communicate with the service provider, a communication channel must be established first and there must be a communication media on which institutions can exchange data or information using shared services. In order to communicate within the institutions, the shared service must be clearly understood and use same understanding level for information exchange [10]. Only the presence of a communication channel and data or information exchange between institutions that have shared service will not be enough; the institutions must be able to use the shared services [11]. For effective use of shared services, the institutions must have service level agreement [12].

Cross organizations boundary requires the study of organizational governance, existing business processes, IT services and technologies to effectively and efficiently work together in order to support the service delivered. There is an interest to go towards implementing e-government services in government institutions in Ethiopia. It is mandatory to consider how cross organizational business processes are modeled to identify cross boundary services. Even though there are different Enterprise Architectures developed, cross organizational business process and service interoperability is not described in detail. Different countries across the world have exercised implementing e-government services such as the USA, UK, Belgium, Australia, and others. They have built their own government interoperability framework.

From African countries, South Africa and Egypt have adopted one of the Enterprise Architectures [13]. The Ethiopian government is working on implementing e-government applications for 22 Ministries and Agencies, out of which Agriculture is one of the most focused sectors on which the government gives much attention since agriculture is the main backbone for Ethiopian Economy [14].

The Agricultural Economy holds 80% of the country's rural development program. The Agricultural Economy holds 45% of the country's annual income. 90% of the country's export income is from agricultural products. 85% of employment opportunity relies on agriculture. Therefore, the Ethiopian government will raise 15% of the country's annual budget. This clearly shows that the government of Ethiopia has wide attention for the agricultural sector [15].

Tesfaye Debela in [9] discussed how business process shall be guided with the theory and practices of institutions. BPR has very strong points where the business process of the institutions will be improved through supporting governance of an institution with implementation of Business Process Reengineering [15].

A few years back, the Ethiopian government has not formulated ICT policies, but now MCIT is working on producing governmental ICT policies that can be adapted by different institutions [14]. In the past, there is lack of standard documents such as ICT standards, polices, Data Center standards, enterprise level security framework, document on information sharing policies, alignment of IT with business requirement, Interoperability Framework, Information Security Standard, cross boundary service level agreement for shared services, adoption of appropriate policies or standards for different technology, application process and architecture are very critical for implementing e-government services.

6. Enterprise Architecture Frameworks

Consideration of Enterprise Architecture from different perspectives has been exerted in this research to identify the limitations that Enterprise Architectures have in handling cross organizational business process for shared services for implementing e-government applications.

Selection of an Enterprise Architecture is based on goals and outcomes or based on approaches in general [16]. A comparison of five Enterprise Architecture are considered even though different Enterprise Architectures have their own advantages. In order to select an appropriate Enterprise Architecture, there must be a feasibility study on the business process and understanding the application software use and characteristics [17]. From different Literature review, Zachman Enterprise Architecture framework is useful for classifying the various architectural artifacts [18, 19] institutions practice guidance is more detail described by Gartner Enterprise Architecture. TOGAF Enterprise Architecture describes process completeness and information about institutions. FEA Enterprise Architecture describes more on organization Maturity Model descriptions and Reference Model Guidance, Partitioning Guidance and Perspective catalog [20]. Both FEA and Gartner Enterprise Architectures describe Governance Guidance, Business focus and Time to value. Five enterprise architecture frameworks have been examined in this paper in order to understand cross organizational business process model for each architectural design. Different information resources such as “Overview of an existing Enterprise Architectures Frameworks” [19] and “A Comparison of the Top Four Enterprise-Architecture Methodologies” have overview on architectural frameworks [16]. Zachman Enterprise Architecture which is the most comprehensive from all Enterprise Architectures is Timeline framework abstraction. Zachman establish Vocabulary set of perspective for describing Enterprise systems using six by six matrixes [17, 18, 22]. DODAF is developed by US defense, the Enterprise Architecture view with three perspectives and linkage between the views defined in dictionary to define terms. FEAF designed by the US government to follow industry trend design Architecture Framework is a practical guide for Enterprise Architecture Framework. FEAF contains guidance and is oriented towards Enterprise Architecture to IT Architecture. TEAF information view is same description like FEAF Data Architecture. TEFA is facilitating integration, information sharing, and exploitation of requirement across departments. It describes work product for documenting and modeling Enterprise Architecture, align with FEAF model and DODAF product. TEAF functional view is same description as FEAF application Architecture. TEAF information view is same description as FEAF technology Architecture. Only TEAF has organizational view. The Enterprise Architecture is TOGAF which focuses on mission critical business application that uses open system building blocks using AMD. TOGAF has strong business architecture and technology architecture view. TOGAF is comprehensive on process involved. TOGAF is more to provide guidance [20, 21].

7. Conclusion and Future Work

This paper has identified existing problem within different governmental institutions while working together for delivering services to citizens. The considered business process study focused on cross organizational business process boundary for shared services. In order to address the existing problems within agricultural research institutions cross organizational business process was considered. The Ethiopian agriculture sector has been selected in

order to consider the existing problems. The agriculture sector has different research institutions that are working on problem identifying, prioritizing, planning, undertaking research, verifying technology and disseminating technology produced for different stakeholders such as farmers, pastoralists, investors, and citizens.

In this research it was tried to show alignment of business and IT can be easy when, the institutions business process has been clearly studied within the different institutions participating for shared services. And as long as the participating institutions share similar government goals and strategies, use of cross organizational boundary business process modeling can lead to organizational interoperability through the use of standard policy, rules and regulation effectively implementation.

Implementing e-government service is not an easy task since different considerations on business governance, Political Governance, IT Governance and other Governance must be studied. Therefore; the paper tried to show clearly how much considering cross organizational business process is very important in order to reach interoperability within different governmental institutions that are working for shared services.

In this research it was tried to show there is a problem in organizational interoperability within different governmental institutions that are working for common goal and strategies by using a case study on agriculture research system. Organizational interoperability is still a research area in different institutions. Therefore, considering different case studies and identify framework that can solved the particular problem with a careful consideration of the business and IT governance.

Even though this research has only considered livestock research system; the existing problems on cross organizational business process are shown within the different governmental institutions in Ethiopia. In order to achieve interoperability among different governmental institutions; modeling cross boundary services is an ideal solution but modeling the business process model is not the only solution but there must be organizational setup in order to control, monitor and evaluate cross boundary services within different institutions.

By modeling cross organizational business process and studying the governance (both business governance and IT governance) an alignment of business process with IT solutions will be achieved.

Since the government is implementing different e-services applications for different ministries and agencies, considering cross organizational business process for shared service is very critical in order to achieve interoperability within the different institutions to successfully implement e-government services that is addressable to different stakeholders.

References

[1] Herbert Kubicek and Ralf Cimander, “Three Dimensions of Organizational Interoperability Insights from Recent Studies for Improving Interoperability Frameworks”, Institute for Information management Bremen (IFIB).

[2] Alejandro Sanchez, Rilwan Basanya, Tomasz Janowski, and Adegboyega Ojo, “Enterprise Architectures - Enabling Interoperability between Organizations”.

[3] “Frameworks, A review of Government Interoperability Frameworks in Selected Countries”, E-government Interoperability, 2007, UNDP, Bangkok.

[4] M. D. Headayetullah and G. K. Pradhan, “Interoperability, Trust Based Information Sharing Protocol and Security: Digital Government Key Issues”.

[5] “E-government Interoperability: Overview”, 2007, UNDP, Bangkok.

[6] The Australian Government Business Process Interoperability Framework, Australian Government information Management Office (AGIMO), 2007.

[7] Abbas Rajabifar, “Critical Issues in Global Geographic Information Management with a Detailed Focused on Data Integration and Interoperability of Systems and Data”.

[8] Behara, G. K., Varre, V. V., and Rao, M., “Service Oriented Architecture for e-Governance Methodology”, pp. 1-13, 2009.

[9] Tesfaye Debela, "Business Process Reengineering in Ethiopian Public Organizations: The Relationship Between Theory and Practice", Ethiopian Civil Service University.

[10] Chakravarti, B., Ieee, S. M., and Varma, V., "An Enterprise Architecture Framework for Building Service Oriented e-Governance Portal Architecture".

[11] Sixth, T, Conference, I, & Milano, M. (2008). Business Process Management. Information Systems Research.

[12] Kambhampaty, S., Chandra, S., Towers, T. S. R., and Road, R. B., "Service Oriented Architecture for Enterprise Applications Architecture", 2006, pp. 48-54.

[13] Position, A., Architecture, E., & Group, W., Egyptian Government Enterprise Architecture Framework (EGEAF), 2006.

[14] National Enterprise Architecture Framework for Ethiopia (NEAFE).

[15] Strohmaier, M. B. (2005). PhD Thesis B-KIDE : A Framework and a Tool for Business Process Oriented Knowledge Infrastructure Development Knowledge Management.

[16] Framework, Z., Framework, Z., & Framework, Z. TOGAF and the Framework - TOGAF and Zachman the Zachman Framework What is the Zachman Framework Framework, 1-8.

[17] TOGAF (The Open Group Architecture Framework) Version 8.1 "Enterprise Edition", 2003.

[18] A Comparison of the Top Four Enterprise-Architecture Methodologies.

[19] The Open Group Architecture Framework - Wikipedia, the free encyclopedia.

[20] A Comparison of the Top Four Enterprise-Architecture Methodologies, Roger Sessions Object Watch, Inc.May 2007

[21] Methodologies, Enterprise Architecture Zachman.

[22] Framework, Z., Framework, Z., & Framework, Z. TOGAF and the Framework - TOGAF and Zachman the Zachman Framework What is the Zachman Framework.

Mobile Banking Framework in the Ethiopian Context

Ayenew Wudu
Abay Bank S.C, Addis Ababa, Ethiopia
ayenewww@gmail.com

Fekade Getahun
Department of Computer Science, Addis Ababa University, Ethiopia
fekadegetahun@gmail.com

Abstract

This research aimed at providing a comprehensive mobile banking framework tailored to the Ethiopian context. To do so, assessment on the existing banking service coverage in Ethiopia has been conducted. Based on the assessment on banking reports, related literature and own experience, Ethiopia remains as one of the under-banked economies. The current practice of opening of new branches is not profitable and the penetration rate is slow to reach the mass market.

In this paper, assessment on the various initiatives taken by the National Bank of Ethiopia (NBE) regarding electronic fund transfer proclamation formulation, the requirement set for local banks to implement core banking system and to do seamless interbank transactions is discussed. The operating environment – legal framework and favorable conditions, mobile infrastructure provided by ethio telecom and the future expansion plan, and the experience of other countries' mobile banking business models are explored to adopt for the local context.

Survey was conducted to identify the most needed mobile banking services to take as an input to propose the mobile banking framework. Based on the data collected from local banks, balance inquiry and fund transfer services stood out as the basic services to be availed.

The proposed mobile banking framework consists of mainly three layers. The presentation layer uses the various mobile technology options as a means to access the mobile application. The business layer shows the mobile banking services. The data layer deals with the underlying data to be maintained. In addition, there is an integration service component for interbank transaction. Finally, an algorithm is designed and a prototype is developed for selected mobile banking services.

Keywords: Banking Framework; Banking Services; Mobile Banking Requirement

1. Introduction

Ethiopia is an under banked country and a lot needs to be done in the banking sector to mobilize resources and to assure sustainable development. There is a big gap between the banking service coverage and the banking service demand by the society. In today's world, there are many banking channels – bank branch, ATM, Internet, and mobile to avail banking services. However, all banking channels are not cost effective for developing countries due to high initial investment requirement. 67% of the world population has mobile handsets which make mobile the largest consumed electronic device in the world [1]. According to Juniper research, the amount of mobile payment transaction is expected to reach 630 Billion USD by the year 2014, with half a billion mobile money users globally.

In this paper, an effort is made to look into mobile banking challenges and requirements with a mobile banking framework tailored to the Ethiopian context. The rest of the paper is organized as follows: Section 2 presents the background information regarding banking services and related efforts. Section 3 assesses related works in mobile banking to learn from prior works. The proposed solution is detailed in Section 4. Section 5 deals with the prototype developed as a proof of concept. Finally, concluding remarks are given in Section 6.

2. Background

Ethiopia is a country with an area of 1.1 million square km and about 78 million population [2].

According to the 2008/2009 annual report of the National Bank of Ethiopia, the total number of bank branches across the country is 636 for all the thirteen Banks [3]. This number may increase as four additional banks are being established and some of the existing banks are opening new branches after the report was released. As of writing this paper, there are seventeen banks, among which three are governmental and the rest are private. Most of bank branches are concentrated in the capital city, Addis Ababa and some regional cities. However, comparing the population and the geographic area coverage, the people-to-bank ratio shows that Ethiopia remains as one of the under banked economies [3]. Actually, this is a critical problem in African countries in which only 20% of African families have bank accounts [2].

Currently, branch expansion is very limited in number and yet branches in different areas are not interconnected. As a result, a customer is obliged to appear in person in a specific branch where s/he has opened account(s) to get banking service. Cash is still the most dominant medium of exchange and electronic payment systems are at an embryonic stage [2]. The conventional banking service, which requires appearing in person at a branch, cannot fulfill the growing customer demand.

In the future, the competition between banks will be towards reaching the unbanked through various electronic channels mainly through mobile. Mobile banking service is promising to reach the unbanked. Mobile banking is a subset of e-banking in which customers access a range of banking products like saving accounts and credit instruments via electronic channels [4]. Recent reports show that mobile phone has better penetration than bank expansion. It is estimated that 2.6 billion people in the world do not have access to formal financial services and yet one billion of them have mobile phones [5].Besides, prices of both handsets and air-time are decreasing from time to time.

In addition, currently, there are other encouraging initiatives that support the introduction of mobile banking in Ethiopia. The "Payment and Settlement proclamation- proclamation No. 718/2011" includes provision for transaction originated from mobile devices, the establishment of a company called EthSwitch to provide interbank transaction services, and NBE's enforcement for implementing standard core banking software are the major ones.

Therefore, to provide such innovative services, we need a mobile banking framework that is suitable for our country. However, associated with mobile banking, there are a number of challenges such as security and lack of comprehensive legal framework that governs the transaction, use of local language, and related issues.

3. Related Work

The literature review in this section briefly shows the various mobile banking business models, security concerns, usability, and related issues. There are two well-know mobile banking business models – additive model and transformational model [5]. In additive model, the mobile phone is merely another channel to an existing bank account. Transformational models focus on providing banking services for the unbanked, who are largely low income people. In transformational model, a bank extends its branches through retail outlets so that mobile banking customers can deposit and withdraw cash at retail outlets (also called agents) [5, 6].

Depending on the nature of the organization providing the scheme, Dermish *et al*. distinguished between bank-based and non-bank-based mobile banking models [5]. For instance, MPESA of Kenya and smart money of Philippines are mobile banking services operated by telecom companies. Wizzit mobile banking solution of South Africa is operated as partnership between the Standard Bank and the MTN- mobile operator.

In some researches carried out on mobile banking adoption, emphasis is given for the impact of social and cultural factors such as perceived credibility, facilitating conditions, perceived elicitation and demographic factors to adopt mobile banking by consumers [7, 8]. In [9], Zarifopoulos and Economides exerted an effort to come up with MoBEF (Mobile Banking Evaluation Framework) which is focused mainly on Internet websites that provide mobile banking services.

In addition, consumers may hesitate to use mobile banking service for security reasons. The security

concern emanates from the nature of mobility and the inherent weakness of wireless infrastructure. Takada and Koike in [10] proposed to use image as password instead of text passwords. Emmanuel in [11] came up with secure framework for SMS-banking services which is focused on security aspect of SMS mobile banking.

Although mobile phone penetration is better than the conventional banking service to reach the unbanked in developing countries, there is a challenge of using mobile banking applications easily by the poor [12]. According to Medhi [12], non-text designs like spoken dialog and rich multimedia are strongly preferred over text-based for such users.

Mobile banking is becoming as one component of disaster recovery solutions [13]. Mobile banking is not only a choice which is convenient for day-to-day financial transactions but also relatively easy to restore a basic banking service from natural or manmade disasters [14]. All these literatures do not show a comprehensive mobile banking framework except discussing a certain part of it in brief.

4. The Proposed Solution

This paper came out with a comprehensive mobile banking framework tailored to Ethiopia. The framework is designed taking into account the extensibility of services, scalability of the application, and usability and security in mind. In order to come up with the architectural framework, it is crucial to determine the architectural significant requirements. Therefore, both the functional mobile banking services requirements and non-functional requirements are identified using survey method. The proposed mobile banking framework is depicted in Figure 1.

The functional requirements are obtained from the survey questionnaire distributed to six commercial banks. The non-functional requirements are partly obtained from the survey and the rest are derived from the application's nature. For instance, it goes without saying that a mobile banking application shall be secure, scalable, available and responsive. Interoperability is also found to be important as there will be interbank transactions that deal with heterogeneous environments.

The framework is composed of three layers. The presentation layer facilitates the interaction between customers and the system. This layer empowers a customer to use the various mobile technology options and local languages while interacting with the system. To bring substantial value to the unbanked, the use of multimedia and local languages on mobile devices is important.

The middle layer represents the various Mobile Banking services offered by the system. Researchers categorized services into two forms [15]. The first classification is based on the nature of the interaction and operations performed as Transactional and Inquiry services. The second classification is based on who initiated the service (the service owner or the consumer), and how information flows (i.e., pull and push). A pull transaction is one in which a mobile phone user requests a service. A push transaction, on the other hand, is one in which the bank sends information based on a set of rules. There is also Bank Entity locator service to locate a nearby bank branch, agent or ATM.

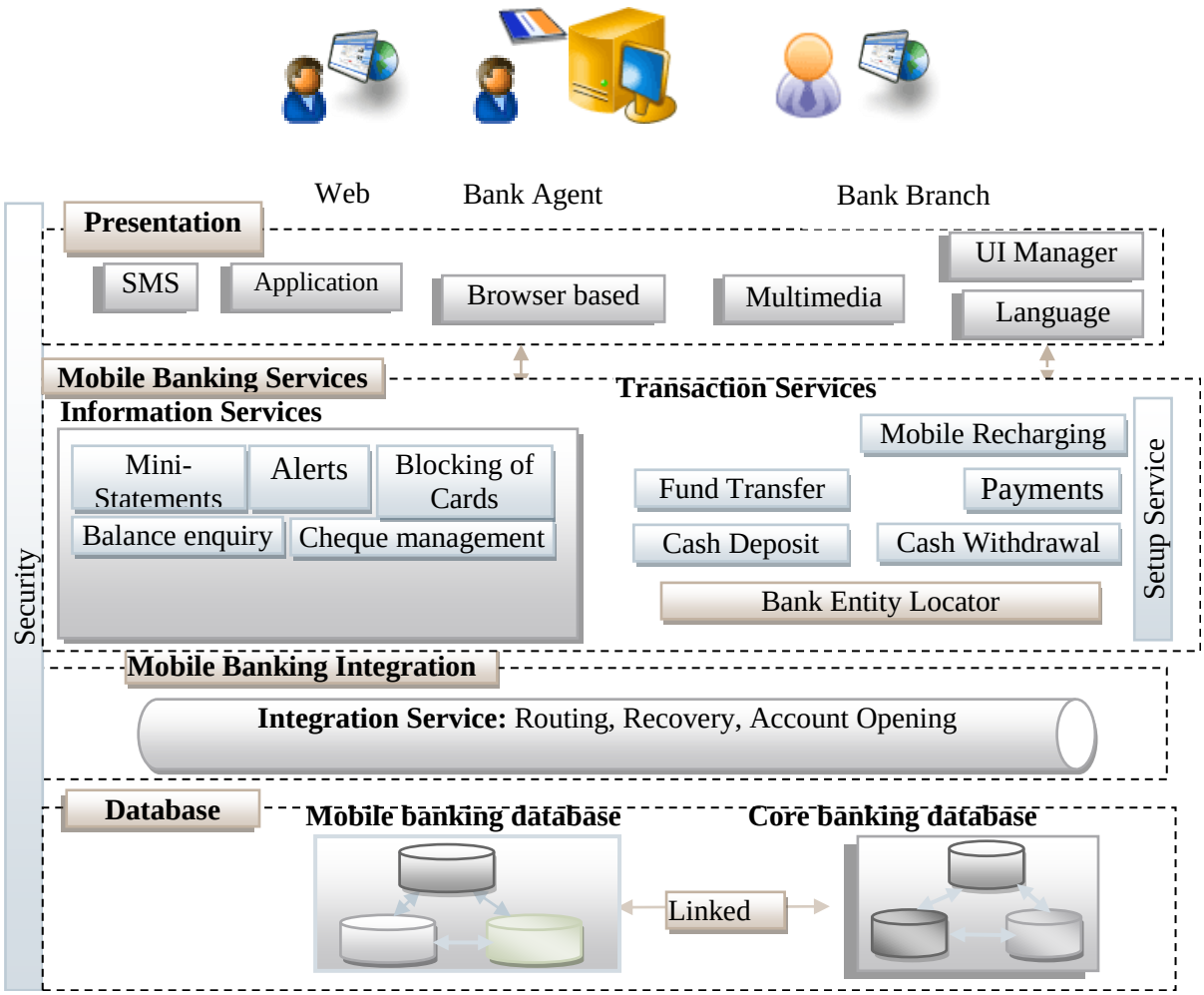


Figure 1: Mobile Banking Framework

The integration service is a platform for building enterprise-level service operation execution. In the mobile banking context, this service is responsible to let customers to transact through without being restricted to a specific bank. That means the service allows interfacing with a standard third party system in order to allow interbank mobile transactions. In this context, banks form a federation to allow customers of a member bank to transact seemingly. The database layer is responsible to maintain the data need of the framework. In our framework, there are mainly two types of databases, i.e., mobile database and core banking database. These databases are distributed and heterogeneous in model, structure, format, etc. Depending on a particular business scenario, a database can be distributed for scalability, availability, and performance reasons.

5. Prototype

After defining the mobile banking requirements and come up with the architectural framework, two databases – mobile baking database and core banking database - are created. A prototype is developed using Microsoft Visual Studio 2008, C# as a development environment. The designed framework is comprehensive and challenging to implement all functionalities within this short period. Therefore, only a small SMS application as shown in Figure 2 is implemented as a proof of concept. In addition to the Microsoft Visual Studio, a trial version of Ozeking SMS gateway was used. The SMS gateway serves as a medium of exchange of messages between the application and the mobile network. The application mainly shows the balance inquiry and fund transfer services. The transfer service is designed and implemented in such a way that when there is insufficient fund in a given mobile account, the transfer is carried out from the usual banking account linked to that mobile.

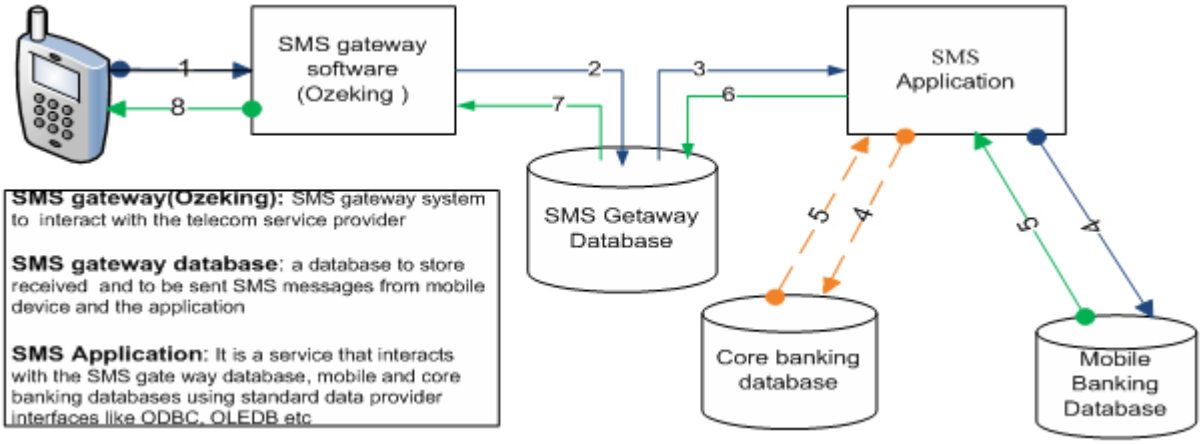


Figure 2: SMS messaging design with databases

The SMS Gateway database uses two tables namely, inbox and outbox to store incoming and outgoing messages. When a message is sent from a mobile as depicted in the mobile banking framework designed in Section 4, the media of communication with the mobile banking system could be through SMS, or a small application to be deployed on each mobile device (like J2ME) or through Internet (WAP). The prototype is focused on SMS because all mobile devices are capable of processing SMS although there are security concerns since the SMS messages are transported as plain text.

When SMS is received from a customer, the SMS Gateway software inserts the message into the SMS Gateway database - the inbox table. The mobile banking application monitors continuously the inbox table and picks the message and based on the request type, it will connect to the mobile database. After processing the request (be it balance inquiry or fund transfer service), the message to be sent is placed in the outbox table in the SMS Gateway database to be sent by the SMS Gateway software to the customer. If the mobile banking application does not get the information corresponding to the incoming message from the mobile database, a request will be sent to the core banking database in case the customer has bank account and is registered for mobile banking service. Note that since the application runs as a service on a computer and handles incoming and outgoing SMS messages, it has no graphical user interface to be shown.

6. Conclusion

As per our assessment, there is no mobile banking service provided so far in Ethiopia although there are some efforts here and there to introduce the technology. A research in the area is almost nonexistent. Therefore, we found designing a mobile framework to reach the unbanked essential. In order to come up with the appropriate framework, the local banking practice and operating environment are assessed. The mobile banking services survey was conducted from six local banks. In addition, related work review has been made. Finally, a comprehensive mobile banking framework is designed. As a proof of concept, a prototype is developed for balance enquiry and fund transfer services on SMS platform.

References

[1] "Mobile Banking – Market Dynamics for Pakistan", www.dryaseen.pk/m_banking_article.pdf, accessed: 20-Mar-2012.

[2] G. Worku, "Journal of Internet Banking and Commerce," *Journal of Internet Banking and Commerce*, Vol. 15, No. 2, 2010.

[3] "National Bank of Ethiopia-Annual report 2008/2009." http://www.nbe.gov.et/pdf/annualbulletin/Annual Report 2008-09/NBE Anual Report.pdf, accessed on 17-Aug-2011.

[4] U. Knowledge, "Mobile Phones and Mobile Banking in Ethiopia," *Intelligence*, 2007.

[5] A. Dermish, C. Kneiding, P. Leishman, and I. Mas, "Branchless and Mobile Banking

Solutions for the Poor: A Survey," *Money*, pp. 1-33, 2011.

[6] L. B. and B. Söderberg, *Mobile banking – financial services for the unbanked?* Uppsala: Reklam & Katalogtryck, 2008.

[7] "An adoption model for mobile banking in Ghana Margaret Crabbe, Craig Standing and Susan Standing Heikki Karjaluoto," *Management*, Vol. 7, No. 5, 2009.

[8] A. Aud and B. Agora, *Mari* Suoranta "Adoption of Mobile Banking in Finland", 2003.

[9] M. Zarifopoulos and A. A. Economides, "Evaluating Mobile Banking Portals," *Publish*, Vol. 7, No. 1, pp. 66-90, 2009.

[10] T. Takada and H. Koike, "Image-based Authentication for Mobile Phones using User's Favorite Images," *Framework*.

[11] Abunyang Emmanuel, "Mobile Banking in Developing Countries: Secure Framework for Delivery of SMS-banking Services" *Challenge*, August, 2007.

[12] I. Medhi, "A Comparison of Mobile Money-Transfer UIs for Non-Literate and Semi-Literate Users," *Design*, 2009.

[13] "Recovery Solutions Provides Mobile Banks after Natural Disaster," 2008.

[14] M. G. Foundation, "Gates Foundation and USAID Announce Innovative Fund to Incentivize Mobile Money Services in Haiti," 2010.

[15] "How Stuff Works 'Mobile Banking Background'." http://money.howstuffworks.com/personal-finance/online-banking/mobile-banking1.htm, accessed on 15-Feb-2012.

Sensor Based Context-Aware System Architecture for Improving Yields in Agriculture

Bereket Assefa Hailemariam
Telpay Incorporated, Winnipeg, Manitoba, Canada
bekihm@gmail.com

Dejene Ejigu
Department of Computer Science, Addis Ababa University, Ethiopia
ejigud@yahoo.com

Abstract

For countries like Ethiopia, agriculture is a backbone. The country's most economic activities depend on agriculture. Agricultural commodities account for half of the Gross Domestic Product (GDP). However, factors like poor infrastructure, traditional techniques used for farming, periodic drought, and soil degradation have been the main obstacles for not projecting the export income, and also for not supporting and sustaining people needing food assistance. On the other hand, globally Information Communication Technology (ICT) is growing in an alarming rate and developing countries like Ethiopia are doing their level best to mainstream ICT solutions in all sectors of the economy. The agricultural sector is one of the areas in which ICT can be of paramount importance.

In this work, we have designed and developed sensor based Context-Aware System Architecture (CASA) for improving yield in agriculture. The work in CASA has focused on modelling and creating system architecture that merges two well known technological domains namely context-aware computing and Wireless Sensor Network (WSN) for the benefit of improving agricultural products. The architecture would present generic solution to solve many problems like crop irrigation, fertilization, pesticide spraying, and early detection of various crop specific issues. The work has mainly focused on one of the problems, crop irrigation.

One of the technologies used is Context-Aware Computing, which involves sensing the context via sensors. Our work involves sensors of different kinds deployed throughout the desired area which are sensing the environmental parameters like temperature, humidity, soil moisture, etc. Then the data would pass through a middleware, which has some components like application interface, sensor network interface, and database controller. The aggregated data would be processed and analyzed to create context model, then an action command would be sent to the respective application.

Implementation of sensor based context-aware solutions in agriculture can be considered as an extension of Precision Agriculture. The proposed work would transform the traditional agricultural practices, would be low cost, would ensure proper monitoring of the fields, would require less involvement of human, would be instantaneous, and would bring about accurate decision making.

Keywords: Precision Agriculture; Context Aware; WSN; CASA

1. Introduction

In many countries, agriculture is one of the most important areas on which human life is very much dependent. Ethiopia's economy is based on agriculture. This domain currently uses traditional methods for producing crops. In the country, this domain would have improved yield production if it had used a different technology which has become an inseparable part for different disciplines. The research will focus on creating system architecture merging two well known technological domains namely context-aware computing and Wireless Sensor Network (WSN).

Context-aware applications adapt according to their location of use, the collection of nearby people and objects, as well as the changes to those objects over time [1]. Context-aware computing involves sensing the context via sensors. Wireless sensor network technology is one of the important technologies for implementing a ubiquitous society, and it is applied into various fields such as

distribution, logistics, construction, transportation, agriculture, defence, medicine, etc. [2]. One area where context-aware and wireless sensor network technologies would be highly beneficial is agriculture.

Agriculture should benefit from real-time data transfer and high resolution of sensor data that shows variations. Sensor technology has the potential to provide added value for agriculture e.g., for improving yield quality or for decreasing costs or risks in production. Currently sensor technology has been most commonly applied in real-time weather monitoring for support of management practices and in precision agriculture [3].

2. Background

In Ethiopia, agriculture is heavily reliant on rainfall; productivity and production are strongly influenced by climatic and hydrological variability that are reflected as dry spells and droughts [4]. The agriculture domain currently uses traditional methods for producing crops. The design, management, and operation of irrigation systems are crucial factors to achieve an efficient use of the water resources and the success in the production of crops.

Globally ICT is growing in alarming rate and developing countries like Ethiopia are doing their level best to mainstream ICT solutions in all sectors of the economy. The agricultural sector is one of the areas in which ICT can be of paramount importance. To accomplish this, the use of pervasive computing technology would be the best strategy. It is widely known that pervasive computing introduces a radically new set of design challenges. In particular, pervasive computing demands applications that are capable of operating in highly dynamic environments and adapting to changing contexts. Context-aware applications adapt according to their location of use, the collection of nearby people and objects, as well as the changes to those objects over time [1]. Context-aware computing involves sensing the context via sensors.

WSN presented itself as an essential part of all systems that require physical and environmental attribute measurement [5]. Wireless Sensor and Actuator Networks (WSAN) emerged as a new generation of WSN that offer both sensing and control. One area where context-aware and wireless sensor network technologies would be highly beneficial is agriculture.

As presented in [6], both irrigated and rainfed agriculture are important in the Ethiopian economy. Most food crops in Ethiopia come from rainfed agriculture with the irrigation sub-sector accounting for only about three percent of the food crops. Better operation and management of irrigation water leads to significant savings and to a more sustainable use of water resources, as well as enhanced soil productivity. To achieve this, systematic monitoring is required for the soil water content in the root zone, as well as for other relevant parameters (soil temperature, salt content of the groundwater, climatic parameters, etc.). As the sensor devices are not biased, they would give us the exact values of the environmental parameters on the basis of which it would be decided that which of the areas is having a deficiency of water, then as per the need of the crop, the water would be supplied. This is the motivation to design a sensor based context-aware system. The system helps agricultural organizations and also local farmers to produce a better crop yield.

3. The Proposed Solution

The proposed system architecture is shown in Figure 1. It is named Context Aware System Architecture (CASA). CASA has three sections, each composed of different components: Sensor Network Manager, Middleware, and CASA Application.

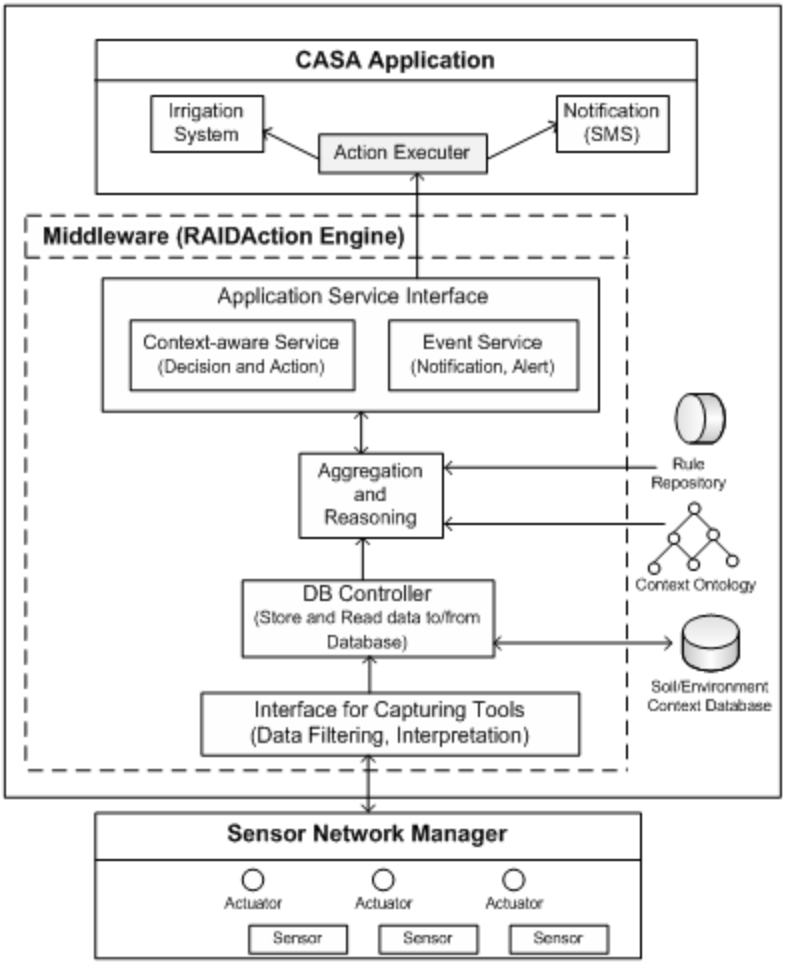


Figure 1: Architecture of the system

3.1 Sensor Network Manager

This is the physical layer where the sensor devices collect soil and environmental data in the field. In the proposed system, we have used a number of sensor devices for data collection. These sensors measure soil and environmental parameters like temperature, humidity, pH, climatic conditions, etc. Sensors convert temperature, moisture, humidity, and pH values in the soil into digital electrical signals.

3.2 Middleware (RAID Action Engine)

In this work this component performs reasoning, aggregation, interpretation, and decision. It reasons over the context information, ontology and user defined rules. After all the reasoning and interpretation is completed, the next step is to decide on which action to execute. It is also responsible to interpret the captured low level context to high level meaningful context. When new contexts are received from the soil/environment, this component loads the ontology and user defined rules for processing. Then SPARQL query would be applied to extract the necessary information. For example, we can query the value where soil moisture is below the optimum value. Another example is SMS message to be sent by the system for critical situations in the farm land.

3.3 CASA Application

This component is where the new system will reside in. The application will be put on the server and it will be accessed remotely via Intranet/Internet. The Administrator would control the system by entering the necessary range values for each parameter. The User would be able to monitor the data collected from the sensor devices periodically. The farmer who owns the farm land sometimes may get SMS message sent from the application on critical parameter values from the soil and the environment.

3.4 Context Ontology

In this work, we have developed an ontology indicating concepts specific to irrigation system for agriculture. These are Person, Soil, Environment, Parameter, Device, Location, and SoilTreatment. These ontology classes have object and data type properties.

3.5 Rule Repository

One of the key benefits of building an ontology-based application is using a reasoner to derive additional truths about the concepts. During reasoning over context ontologies, reasoners use different rules to derive additional knowledge assertion from existing RDF/OWL statements. There are ontology rules which are implicitly defined and user defined rules that are explicitly defined and associated with the reasoner. In the proposed work we have used different user defined rules associated with the Jena reasoner, in addition to the ontology driven rules. Below is a user defined rule which is used by the system.

If a Soil s has a water content value less than some thresh hold value, then apply water to s.

Rule representation using generic rule syntax:

```
[ SoilMoisture_Rule:(?sensor CASA:locatedIn
      CASA:Location)

(?sensor CASA:detectParameter ?param)

(?soil CASA:hasSoilParameter
      CASA:soilMoisture)

(?param CASA:hasMinValue ?pmin)

(?param CASA:hasMaxValue ?pmax)

(?param CASA:hasValue ?smv)

LessThan(?smv,?pmin)

-> (?actuator CASA:applySoilTreatment
      CASA:WaterTreatment) ]
```

4. Prototype

In this section, we have developed simulation system to evaluate the proposed work. The simulation system has generator method which generates similar soil and environmental parameter values as the real sensor devices. These low level parameter values are processed to get high level context data. Then decisions are made to perform an action whether to initialize the Irrigation system or not. The processing (reasoning) and the decision are implemented using Java class.

The system has also an interface which displays generated soil and environment parameter values based on time interval. The average results of each parameter values are also displayed. There are other interfaces to set optimum range values for each parameter and also to add mobile phone number of the farmer. The system uses the mobile phone number to send SMS message to the farmer on critical situations which affect the normal growth of the crops.

The sensor devices send real-time data to the server through the gateway using wireless communication. The gateway can be connected to the server using wireless communication or cable. The data collected from the sensor devices is stored in the database for processing.

The CASA application, for this case simulation of Irrigation Control System, is deployed in the server so that it could be accessed in the local network or remotely. It is implemented using Eclipse thread which periodically generates valid values of soil and environmental parameters. This depends on range of values pre-configured in the application. For example, pH value should be between 1 and 14. The application has a user interface which the user monitors the data collected from the sensor devices periodically. The Context Reasoner component gives final a reasoning based on predefined rules. It is implemented as Java class which intensively uses the Jena API.

In this section, we will demonstrate the implementation of the prototype. The demonstration is prepared based on a scenario taken from Gensis Farm plc.

For this scenario we will take only Soil moisture as a parameter and examine how the reasoner component derives decision based on the user defined rules. The Soil moisture parameter reading shows DRY after calculating the average collected data. This parameter reading is not good for vegetable. Therefore, the system will take an action

by initializing the Irrigation system to make the soil wet enough for the vegetable. Table 1 shows the final context output. Figure 2 shows the data collected from the sensor devices and calculated average values.

Table 1: Soil moisture final context

Parameter	*Status context*	*Final decision (after reasoning)*	*Action*
Soil moisture	Soil moisture is DRY	The soil moisture is DRY and is not good for the vegetable.	Initialize Irrigation System

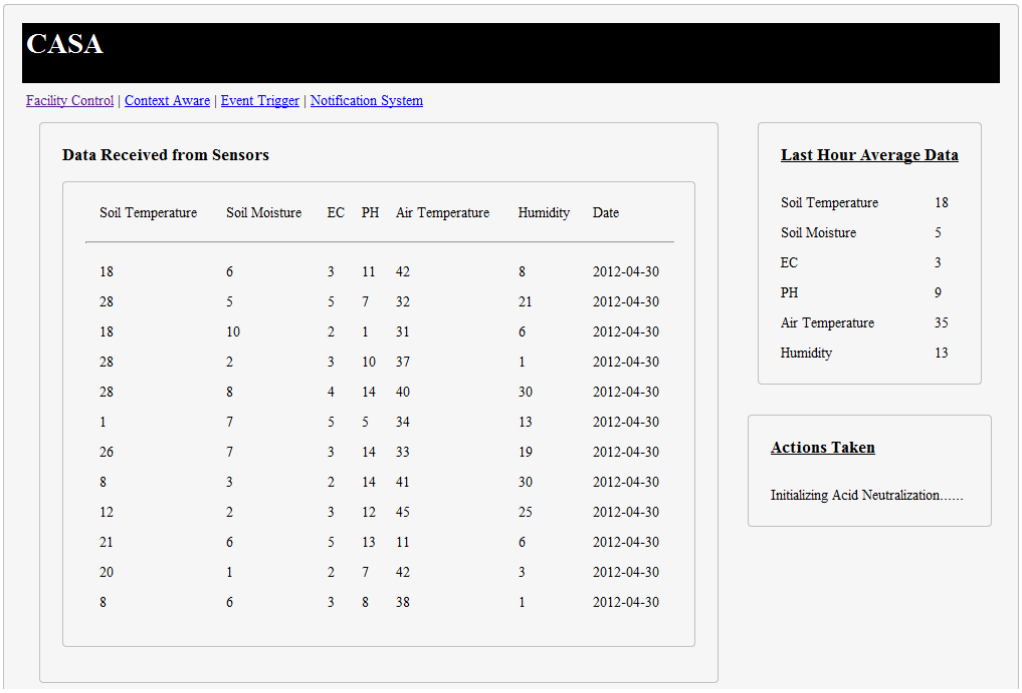


Figure 2: Data collected from sensor devices

5. Related Work

This section discusses some related works in the area of context-aware computing and wireless sensor networks f agriculture. Agriculture is one of the best suited for dynamic situation having highly variable environment. This concept is utilized by different researchers toward this domain.

The authors in [7] reported the use of sensor networks for integrated management of a vineyard. Their work was motivated by the primary importance of temperature in the development of grapes to ensure wine quality. Field work was conducted, 48 nodes were deployed over a period of more than 6 months in an Oregon vineyard, reporting temperature every five minutes. The results were logged in a centralized way and could be displayed on a map and retrieved on a per-sensor basis. Moreover, alarms were sent when the temperature decreased below 0, indicating a risk of frost. The history of temperature variations throughout a cropping season is especially critical.

The work in [8] explores the potential of wireless sensor networks for nationwide cattle monitoring. Each wireless sensor acts as an extended RFID ribbon storing the identity and health status of the animal, which can be tracked at different locations, such as pasture or farm buildings. Each location is equipped with a base station opportunistically recording the information from the ribbons as the animals come into its range. The system was evaluated through extensive rounds of simulations.

Tim Wark *et al.* [9] deployed soil moisture nodes which use commercially available ECH_2O capacitance based sensors that measure the surrounding soil's volumetric water content. These sensors generally don't require calibration and have an error of +/– 2 percent. The network automatically takes readings, typically at one-minute intervals, from each node and sends them back. Data is

aggregated at the base to give an up-to-date moisture profile for the whole field.

Jadoon *et al.* [10] proposed a machine based (IT Based) irrigation control system that could deal with the proper monitoring and distribution of water in Pakistan. The country's economy is based on agriculture. Due to mismanagement, the country faces immense problem of water shortage. The researchers proposed solution which uses WSN for Irrigation Control.

6. Conclusion

Implementation of sensor based context-aware solutions in agriculture can be considered as an extension of Precision Agriculture. By monitoring soil, crop, and climate in a field and providing a decision-support system, it is possible to deliver treatments, such as irrigation, fertilizer, and pesticide application for specific parts of a field in real time. We have presented in this paper an ontology-driven architecture for developing precision agriculture application. The proposed work would transform the traditional agricultural practices, would be low cost, would ensure proper monitoring of the fields, would require less involvement of human, would be instantaneous and would bring about accurate decision making. Thus, it is believed to contribute towards improving yields in agriculture.

References

[1] Aqeel-ur-Rehaman, and Zubair A. Shaikh. "Towards Design of Context-Aware Sensor Grid Framework for Agriculture", World Academy of Science, Engineering and Technology, 2008.

[2] Jeonghwang Hwang, and Hyun Yoe. "Study on the Context-Aware Middleware for Ubiquitous Greenhouses Using Wireless Sensor Networks". School of Information and Communication Engineering, Sunchon National University, April 2011.

[3] Sirpa Thessler, Lammert Kooistra, Fredrick Teye, Hanna Huitu, and Arnold K. Bregt. "Geosensors to Support Crop Production: Current Applications and User Requirements", June 2011.

[4] Seleshi Bekele and Douglas J. Merry. "Assessment of Small Scale Irrigation and Water Harvesting in Ethiopian Agricultural Development", International Water Management Institute (IWMI).

[5] Aqeel-ur-Rehman, Zubair A. Shaikh, and Humaria Yousuf. "Crop Irrigation Control Using Wireless Sensor and Actuator Network (WSAN)", National University of Computer and Emerging Sciences, Karachi, Pakistan, 2010.

[6] AQUASTAT - Survey, "Irrigation in Africa in Figures," Water Report No. 29, 2005, http://www.fao.org/nr/water/ aquastat/countries_regions/ethiopia/index.stm, accessed on October 13, 2011.

[7] Jenna Burrell, Tim Brooke, and Richard Bechwith. "Vineyard Computing: Sensor Networks in Agricultural Production". IEEE CS, 2004.

[8] Milena Radenkovic and Bartosz Wietrzjk. "Mobile Ad Hoc Networking Approach to Detecting and Querying Events Related to Farm Animals". Silicon Valley, USA, 2006, pp. 109-115.

[9] Tim Wark, Peter Corke, and Pavan Sikka. "Transforming Agriculture through Pervasive Wireless Sensor Networks", pp 50-57, 2007.

[10] Sultanullah Jadoon, Salman Faiz Solehria, and Mubashir Qayum. "A Proposed Least Cost Framework of Irrigation Control System Based on Sensor Network for Efficient Water Management in Pakistan". Hazara University, Pakistan, 2010.

Web Based Cultural Heritage Inventory System

Dereje Tadesse
National Archives and Library Agency, Addis Ababa, Ethiopia
derejetad@gmail.com

Fikreyohannes Lemma
PhD Candidate, Addis Ababa University, Ethiopia
fikreyohannes@yahoo.com

Abstract

Ethiopia is endowed with magnificent Cultural Heritages. Cultural Heritages are the result of creativity of mankind and have scientific, historical, cultural, and artistic values. Cultural Heritage Inventory allows to safeguard and manage priceless Cultural Heritages. However, Cultural Heritage Inventory in Ethiopia is in its rudimentary stage. As a result, the vast majority of the country heritages are inaccessible, undeveloped, and endangered for illegal trafficking.

The Web Based Cultural Heritage Inventory System Project is initiated to study the problems related to a Nation Wide Cultural Heritage Inventory system and come up with a software prototype that can tackle the cultural heritage inventory problems.

Attempt is made to study in depth the cultural heritage inventory work flows with particular interest on the field work. Literature review is made on filed work scenarios. The current system analysis is made by conducting qualitative study. Data collation is made through semi structured interview with purposefully selected Heritage experts at ARCCH. Secondary source data are consulted in similar systems surveys and GIS technology review is conducted. Through comparative analysis, existing system analysis is made. A Rational Unified Software Engineering process is used to produce architectural goals specification and the architectural documents. Object Oriented Programming is used to develop a prototype.

We identified the modality of Heritage Inventory field work as sedentary. We specified the architectural goals of the system and based on driving quality attributes, we proposed multi layer server client architecture for the implementation of the system. For interoperability, web services are proposed. The architecture incorporated web service for interoperability and Geographic Information System (GIS) for distributed heritage management. We developed a prototype that tires to avoid difficulties related with cultural heritage inventory.

Keywords: Heritage Inventory; GIS; Field Mobility; Client Server Architecture

1. Introduction

One cannot think of the modern world without software. E-government has become the order of the day. We cannot think of accountable and efficient public services without the application of software. The application of computing in various fields has brought increase in productivity, improvement in quality of service, and accuracy in collection and processing of data. This is not an exception in the cultural field and in particular in Cultural Heritage Inventory. Cultural Heritage Inventory is characterized by a data intensive activity that is carried out on the field. The application of computing in field work is increasingly narrowing the gap between a field work and an office and allowed in acquisition of sound field data, its effective management, and enrichment of the field work experiences.

This paper attempts to address the problem of Cultural Heritage Inventory as a field work from the perspective of software engineering. The software engineering methods followed to examine the problem area and come up with a working prototype are discussed. The functionalities of the prototype are presented.

2. Background

Ethiopia has recognized the benefits of knowledge economy and has embraced e-government. The government has introduced e-government policy. A functional model is developed to implement the policy. The functional model is

expected to tackle the problems in public sector services and make accessible government services and information to the public. One of the components of the model identified is Public Interface Applications (PISAs). Community and Public Interest Services are group of applications identified under this component and one of which is Cultural Heritage Inventory System [1].

Ethiopia is known for the vast richness of her Cultural Heritages. The Research and Conservation of Cultural Heritage (ARCCH) is a government body responsible among others to carry out a scientific registration and supervision of Cultural Heritages. However, the existing manual registration system is inefficient, expensive, and unmanageable to carry out the Cultural Heritage Inventory [2]. Lack of adequate information of Cultural Heritages has made the protection and preservation task inadequate and incapacitated the Cultural Heritages Development effort and hampered tourism development.

In Cultural Heritage Inventory process, the huge task of registration and data collection is done in the field. There are studies that have been carried out on the application of mobile computing in a field study. The studies identified three modalities in a field study based on user mobility and the associated use context. According to Kristoffersen and Ljungberg [3], the three modalities of mobility are wondering, travelling, and visiting. The field work mobility study is important to determine the type of technologies to be used and the software to be developed.

3. Objectives and Research Methodology

The general objective of the research is to come up with a working prototype that facilitates the field work in Cultural Heritage Inventory. To obtain in-depth understanding of the problem, qualitative research design has been adopted. Data collation is carried out by preparing semi-structured interview. Purposeful sampling technique is used to select ARCCH‘s experts that are engaged in Cultural Heritage Inventory and six of the ARCCH Heritage Management experts, managers and database administrators are approached for interview. Secondary sources such as reports and manuals are consulted. Review of related theoretical and empirical research literature is conducted with special focus on Geographic Information System, Web technology engineering methodology, architecture, and computing in field work context. Comparative analysis is used to understand the problem area and test driven development is used to develop and test the prototype. The research is mainly concerned on Cultural Heritage Inventory in the field work. However, it attempts to address software engineering issues related to usability, multimedia data collection, spatial data capturing, data visualization, presentation, and diverse data management. The system is believed to be beneficial in caring out efficiently and accurately the cultural heritage inventory. It will be applicable for day-to-day activities of experts that are engaged in cultural heritage inventory at ARCCH and regional cultural bureaus.

4. Software Requirement and Architecture

Based on the current system analysis conducted at ARCCH, the major activities of the Cultural Heritage Inventory field work are related to observation, measuring of artefacts, and recording of their observation. The cultural heritage field work modality is categorized as sedentary where experts spend some time in a given place to do inventory. Experts on the field collect heterogeneous data. Back office standalone inventory system has been identified whereby the field work data is entered.

A survey of similar systems reveals that inventory systems are web based. Given the heterogeneity of data to be collected, the systems have functionality to manage video, image, text, and spatial data. Most of them integrate Geographic Information System (GIS) to manage the spatial data. The various functionalities provided by the system allow users to update, view, edit, and search inventory data based on assigned privileges. XML and web service is the chosen functionality they implement to interoperate with other systems. Some systems are developed with Open Source and others with proprietary software. GIS allows not only the day to day management and maintenance of heritage site but also natural and cultural resource spatial distribution and inventory assessment. It also enables

identification of assessment of potential threats to cultural resources [4, 5, 6].

GIS technology is now matured enough to be applied in mobile environment. Mobil GIS as it is known allows for capturing of spatial data on the move using wireless application protocols. Web or Internet-GIS is a distributed system that works in networked computers in order to integrate, disseminate, and communicate geographic information across the internet [7, 8]. Web based GIS is identified as the technology to be implemented in the system.

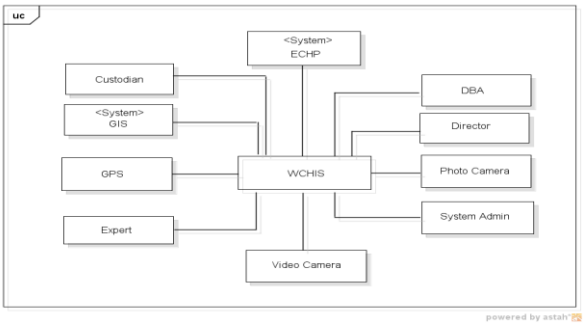


Figure 1: WCHIS Context Diagram

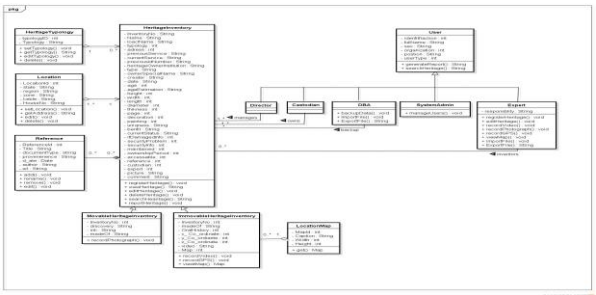


Figure 3: Class Diagram of the System

The data diagram that is mapped to the class diagram and which depicts the association between the relational tables is shown in Figure 4.

Fifteen sequence diagrams are used to describe graphically the sequence of interactions among architectural elements. The hardware and communication interface and the quality attributes of the system are specified. High level quality attributes are identified which are related to performance, interoperability, robustness, usability, maintainability, and security. Based on this requirement, the system architecture is designed. To develop the system architecture, the architecture of

Through in-depth analysis of the current system and survey of similar systems, the requirements of the systems are identified. The system context is defined as shown in Figure 1 and the functional and non functional requirements are specified.

Thirteen use cases are identified that are related to Heritage Inventory, Spatial Data, video, pictures capturing, user management and interoperability as shown in Figure 2. They are prioritized and their use case scenarios are prepared.

Structure of the system in terms of objects, classes, attributes, operations, and their associations are identified as shown in Figure 3.

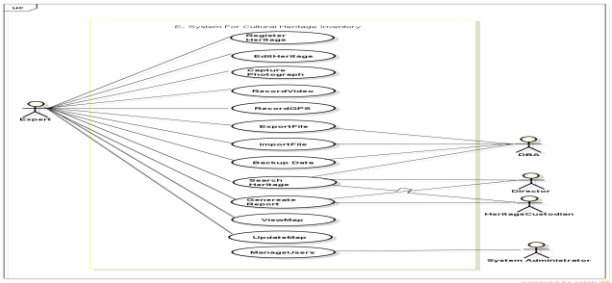


Figure 2: WCHIS Context Diagram

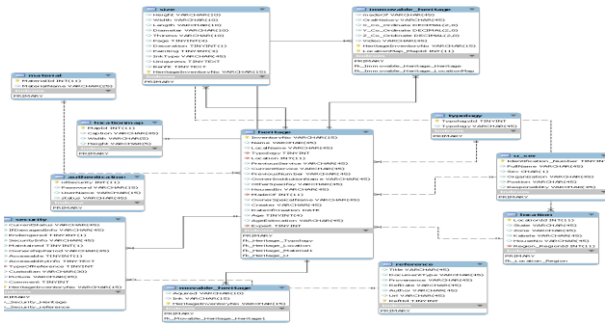


Figure 4: Association between relational

two similar systems is examined and well proven architecture styles and techniques are consulted.

Based on examination of the architecture of two similar systems, i.e., Ethiopian Cultural Heritage Project and Cultural Heritage Inventory Management System (CHIMS), the client server architecture is identified as the proper architecture pattern to be used. Taking into consideration the special requirements of WCHIS, a multi-tier client server architecture is chosen to be the architecture of WCHIS.

WCHIS's client server architecture pattern is separated into different layers with clear interface of communication between the layers. The data and

presentation services are loosely coupled which favours modifiability. The processing task can be shared between the application logic and the database server which improves the quality of service and enhances performance. Interoperability is supported using web service. It is scalable and allows integration of spatial and non spatial data. The multi-tier client server architecture of the system is depicted in Figure 5.

The system is designed based on open source technology as shown in Figure 6.

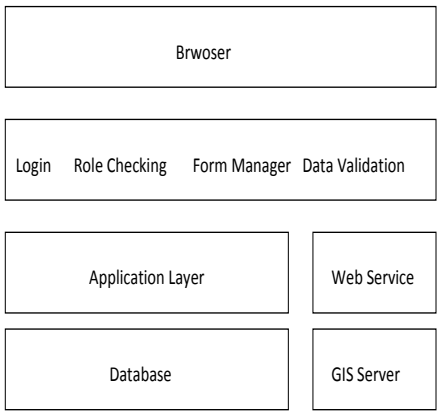


Figure 5: WCHIS System Architecture layers

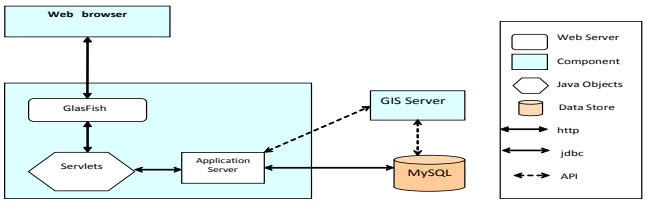


Figure 6: WCHIS System Design

5. Prototype

The development of the WCHSI system has followed the Object Oriented (OO) software development method. The method is chosen because it allows reusability and maintainability. The OO design and development is realized using the de facto standard Object Oriented Unified Modelling Language. Java programming language is used for the implementation of the system. The advantage of writing code in Java is that it is Object Oriented Programming language that allows maintainability, reusability, and portability.

The prototype has login page (Figure 7), movable and immovable heritage page, user admin, and report page. Users of the system need to log on to the system. The system allows users to gain access to pages based on their privileges. The user admin page allows the management of new and existing users. The system admin has access to the admin page. The system admin may grant, change or revoke user's access to the system.

The main work bench of the system is the movable and immovable Heritage Inventory Page. It is organized in such a way that a series of pages are prepared for registration of the inventory data. The user navigates through the pages to register the inventory data. Users are provided with the appropriate links that allow them to view previously registered data and make the appropriate updates. Figure 8 shows one of the series of pages opened by users to do registration.

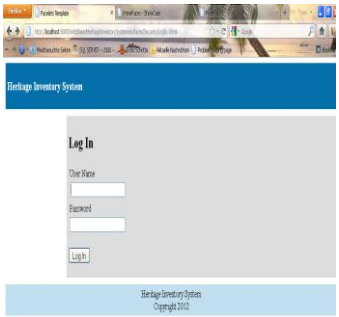


Figure 7: Login Page

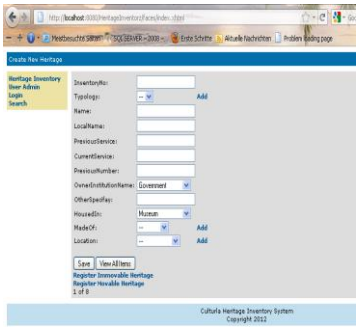


Figure 8: Inventory Page

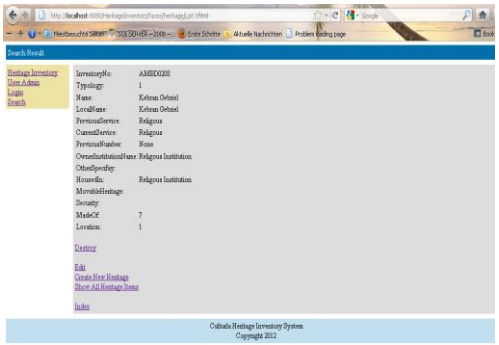
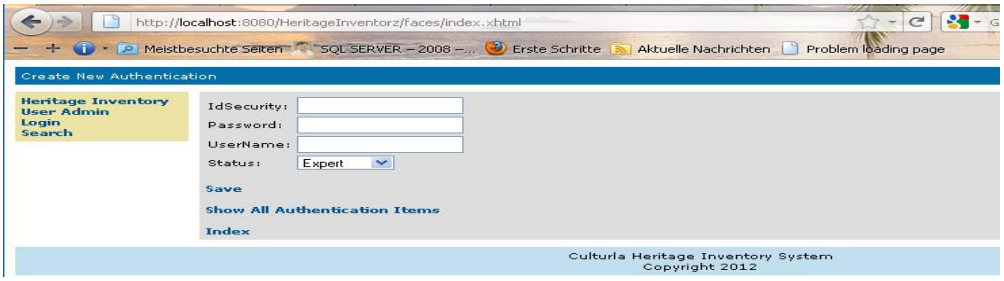


Figure 9: Search Result Page

The prototype allows users to register heterogonous data, view the data when necessary, and make the appropriate updates. The prototype runs on Glassfish server on a web browser and at the back end supported by MySQL. The prototype search page as shown in Figure 9 and the user admin page shown in Figure 10 are some of the pages that demonstrate the look and feel of the application.

Figure 10: Search Result Page



6. Conclusion and Recommendation

Cultural Heritage Inventory in Ethiopia is in its rudimentary stage. In order to appreciate the problems and come up with alternative solution, a literature review is made, qualitative study is employed, and system analyses document produced. Based on the current system analysis, related systems survey, and GIS technology review, the architectural goals for the new system are specified. Analysis of candidate architectural models is undertaken and multi layered client server architecture is proposed. The architectural views of the system are designed and object oriented programming is used to develop prototype of a Web Based Cultural Heritage Inventory. In ARCCH's strive to align ICT with its business process, the automation of the inventory process at a grassroots level should be given due attention. The research demonstrated what can be achieved using low cost open source technologies.

Given the federal government structure and the strong stake the regional governments have in the Cultural Heritage Inventory, they need to manage their local heritage data. Thus the system needs to work in distributed database while the central database is located at ARCCH. As future research, distributed database design and implementation issues taking into consideration the available IT infrastructure need to be addressed.

References

[1] EICTDA, Ethiopian Government Information Communication Technology Policy, pp. 12, 15-16.

[2] Federal Negarit Gazeta, Research and Conservation of Cultural Heritage Proclamation No 209, Addis Ababa, June 27, 2000.

[3] Kristoffersen, Steinar and Ljungberg, Fredrik, "Your mobile Computer is a Stationary Computer", Viktori Research Institute, Sweden.

[4] Buhagiar, C. Michelle, Bailey, Tony *et al.*, "CHIMS: The Cultural Heritage Inventory Management System for the Maltese Islands", The 7th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST (2006)

[5] Çayırezmez, N. Atalan, "Cultural Heritage Inventory System of Turkey on the Web", www.isprs.org/proceedings/XXXVI/5-C53/papers/FP040.pdf last accessed on January 1, 2012.

[6] Meye, Élise, Grussenmeyer, Pierre, *et al.*, "Integration of Heterogeneous Cultural Heritage Data in a Web-based Information System: A Case Study from Vianden Castle, Luxembourg," CAA Proceedings, 2006.

[7] Kenneth E. Foote and Anthony P. Kirvan, "Web GIS," University of Texas, 1997 http://www.ncgia.ucsb.edu/giscc/units/uI33f.html, last accessed on January 12, 2012.

[8] M.A.Brovelli and D.Magni, "Open Source Mobile GIS Solutions for Different Application Fields" Italy, http://www.cirgeo.unipd.it/cirgeo?convegni/mmt2007/proceedings/papers/magni_diego.pdf, last accessed on January 12, 2012.

Wave Process Model for Open Source Software Development

Ermias Kebede
Software Engineer, Switzerland
ermi.kbd@gmail.com

Oliver Petzoldt
iceaddis Innovation Hub, Addis Ababa, Ethiopia
oliver.petzoldt@cimonline.de

Abstract

Wave Process Model (WPM) is a new form of modeling approach to represent the relationships between processes and activities in the form of a wave graph. In this explanatory research, WPM is applied to describe the open source software further development process. As being a lightweight model, on the y-axis we placed processes of the selection, the reuse, the modification, the extension and the contribution. On the x-axis there are activities which can be realization of internal requirements or design patterns. The intersection of processes and activities are tasks to be accomplished by the software enhancement.

In the research result, we find an open-end description of the open source software further development and its process model having parallel and concurrent WPM parts. With a prototype development, we could see that the WPM concept can be applied for project management and controlling tool. These facts are more elaborated with case studies of deploying, internal requirement, redesign and source code further implementation parts to enhance open source software for additional use.

Keywords: Process Model; Software Development; Open Source Software.

1. Introduction

Process models have a view of interest, which are activity, product, resource or role based models [4]. Software development life-cycle models are divided into descriptive or prescriptive [12]. The methodology or models which are used to develop computer systems are said to be related to the activity chain of the development process namely from planning till release [2, 3]. We preferred to describe the WPM as a chain of activity in the process where activities and processes are related with tasks or assignments.

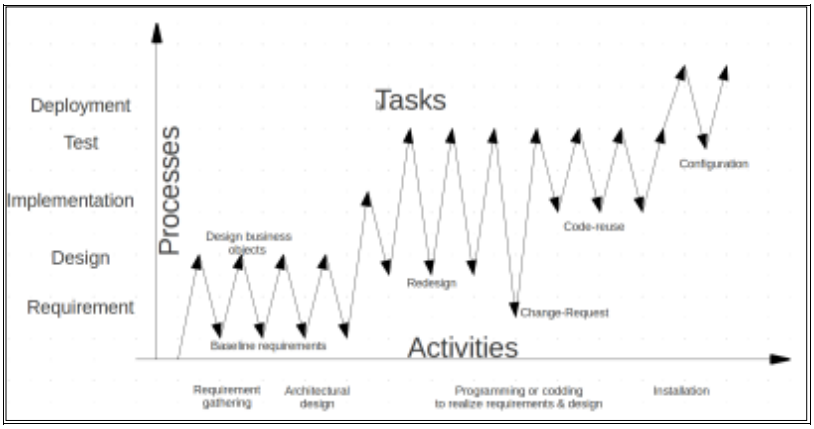


Figure 1: WPM-Sample

Software development process models like the waterfall model, spiral model and agile method seem to lack describing some process and activities involved in the open source software further development OSSFD [8, 9]. Activities by OSSFD were not best compatible to be managed with existing process models [5].

As the primary purpose of the life cycle model is to communicate the work to be done among human beings, work unit leveling is achieved by decomposing [7]. We took the decomposition of work units into waves, waves into activities and activities into tasks. This means a wave describes a kind of phase. What we add in the WPM is the tasks

to demonstrate the relationship between processes and activities as shown in Figure 1.

Requirement engineering is essential when an institute wants to adopt an OSS, because when we customize or add features, we are dealing with requirements which may not exist in the OSS. The other use of it is also if we should select an OSS we may evaluate it with our internal requirement. In this case we added internal requirement in case of OSS further development process. The important part of our research is the concept of the WPM itself which was enforced by the prototype development and the case studies.

2. Background

From the classifications of OSS development framework [11], deploying OSS product, using the OSS CASE Tools and integrating OSS Components talks in general about usage. The other three parts participating in the OSS products are related to development. OSS is there in the Internet cloud; we can use it or develop it further. In OSS, we do not only have software, but also communities [1]. Additionally, How-To, FAQ, and Blogs are where we find Information [8]. Information are like bug request (as requirement), definitions and even design documents.

To select an OSS, we may evaluate it to choose the appropriate one [11]. After selection we can deploy an OSS to use it or to develop it further. Basically there are contributors which are users and developers. Users are those who use OSS and developers are those who provide or develop OSS further. There are individuals, companies, or organizations who participate in OSS development [10].

Those who provide OSS may start from scratch at the beginning but when they released it, OSS can be selected and further developed from interested participants. Further development activities are done with reusing, modification, and extension. Reusing can mean to integrate code-example by copy and paste or calling a predefined method or class. Usage means using OSS applications, tools, concepts, and communication media.

Communications are in chat-rooms, e-mail or community websites. Contribution can be anything like feedback, bug request or financial matter [8]. There is no as such a defined life cycle, rather an open-end further development process as shown in Figure 2.

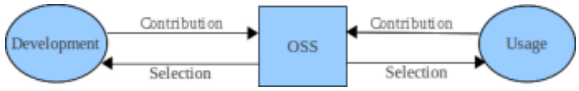


Figure 2: Open-Source Software Life Cycle

If we are going to further develop an OSS, it means we are going to reuse what is there and add more modules or features by modifying and extending some parts [6]. If we are going to have our own individual requirements, we can do it by ourselves. If we are going to have our own development process model, we should not forget to include selection and contribution. Selection is because of using OSS tools and other products [11]. Contribution is not only because of the philosophy but also for information storage.

3. The Proposed Solution

The life cycle for OSSFD can be described in the open-end interaction as shown in Figure 1. Like production and consumption of a product, we have OSS development and usage.

The process modeling approach for further development of OSS is a group of processes which have their own WPM. They can communicate with selection and contribution at any time.

The selection process is there to get information from OSS communities, FAQ, How-To, Repositories, issue lists, code-examples, etc. It suggests contributions of comments, blogs, multimedia materials, etc. The WPM features for the OSS further development are selection, reuse, modification, extension, test and contribution as shown in Figure 4 in each WPM graph of deployment, requirement, redesign and implementation. The activities can be gathering requirements, database design, customization, realization of specific class, and others where we get the input for specific assignments or tasks.

Table 1: Process Names

Process	*Reusing*	*Modification*	*Extension*	*Test*
Deployment	Installation	Configuration	Plug-In, Add-on	Executing,
Requirement	Use-Baseline	Change-request	Feature-request	Validation
Redesign	Use-pattern	Modify	Extend	Validation
Further Implementation	Copy-Paste, Class-Reuse	Customization, Re-factoring	Add-New class, method or else	Run, review, Experiment

The WPM graphs for the parallel processes in Figure 3 can have appropriate names.

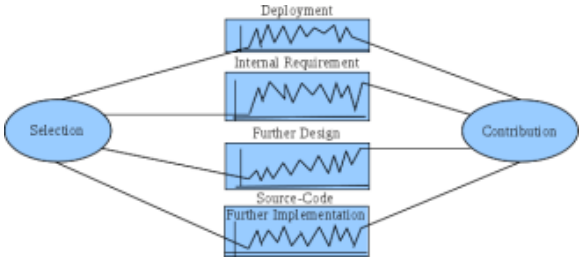


Figure 3: WPM approach for OSSFD

Let us clarify the process for Figure 3 and Table 1. In case of deployment why we choose installation as reusing is because it is the first thing what we do after selecting an OSS. By the configuration tasks, we may modify configuration files. It is also true that some OSS may need other plug-in installation if we need to have additional features. A test can be simply executing the installed application; if it fails we may need to troubleshoot by selecting again a solution in the community website or by applying our previous experience.

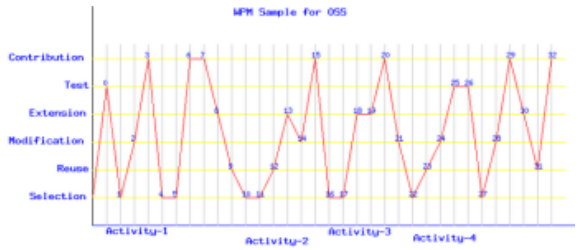


Figure 4: WPM for OSSFD Processes

Internal requirements are there if we would like to customize or add new features in the original OSS. In this case we may gather additional requirements. Baseline requirements could be existing artefacts, known business processes like tax calculations given from the government, etc. We can reuse a document from previous work process, reuse universal formulas, facts and domain specific requirements. But when we would like to modify some behaviors we are talking about a change-request. The other part what we mean with the extension is a kind of new feature request. Since we have a test part, the appropriate word for that is validation. Contributions can be then communicating or writing the requests and issue lists in the form of a notice, a table or other form of a document.

Design patterns are there to be reused. Normally in the OSS itself we have already the design in the source code. But in case of forking we may even need to redesign. Modification may take place when we have to change the user interface. Extension examples are there when we have to add additional tables and relationships in the database design.

By source code enhancement or further implementation, we can select an internal requirement, a bug-request, a source code, a design concept, an interface, etc; by exploring, inspecting, analyzing etc; as an input for translating a concept to the source code. Reusing can be performed with copying and pasting a code example, see how class can be implemented in the programming language, how to call a specific method; etc. Modification can be applied by changing configuration values or renaming some attributes to use a specific variable for other purposes. Moreover, we can add new classes, new methods, new algorithms, new scripts, new concepts or else, to add new features, to realize a plug-in or to append additional modules. The test part can be just running the application to find out if the reuse, the modification or the extension we have implemented has worked properly. Contributions can be done with documenting in the source code, commenting while committing and more.

4. Prototype and Case-Studies

4.1 Prototype

The prototyping let us demonstrate how we can show the application of the WPM. Moreover, it supported us to refine the WPM. For example, associating the tasks with activities let us notice that tasks are repeated, feeding the case study data in to the database to display them and practicing to select and reuse OSS products and tools.

The requirement were to realize the WPM graph for recording and displaying the processes, activities and their relationship with the tasks or self assignments. For that matter, we had three tables respectively and their relationships with the corresponding work units on the database and in the source code we had also the mapping classes. How the prototype is realized was also by applying the reuse, modify and extension methodology.

In the development of the prototype, we used the three tier architecture of the application as an input for the activities. Activity-1 = user-interface realization, the second is the application logic for the WPM graph, the third is the database design with source-code mapping, and the forth is their integration as further development activity.

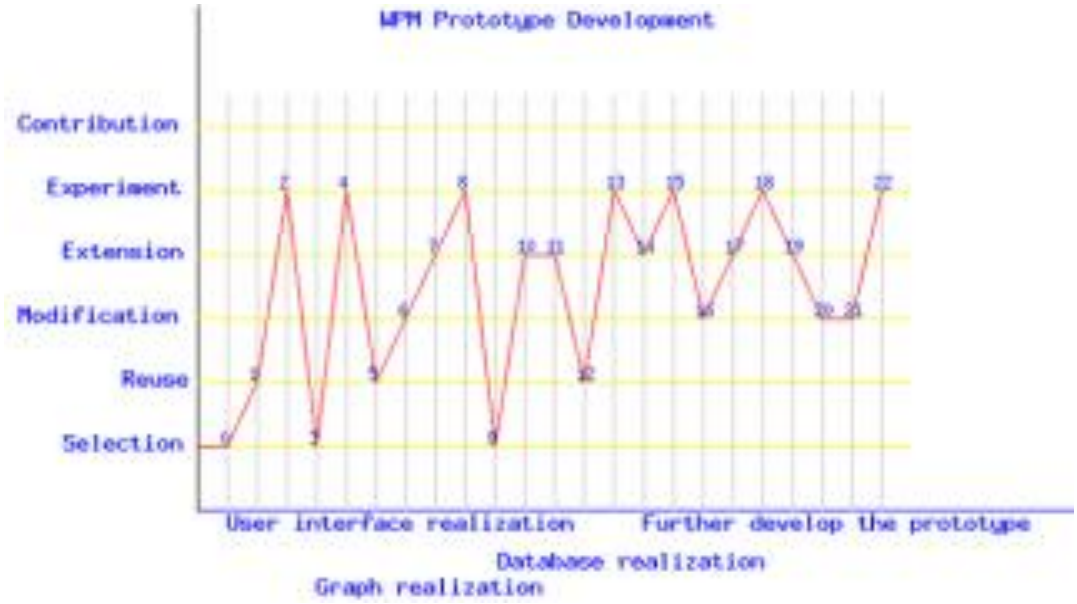


Figure 5: WPM for Prototype Realization

We can see the WPM graph instance in the prototype development in Figure 5. The numbers are those assignments done in the development activities. Those activities were recorded in the database and displayed in a list-box. For example, the following notices are for the tasks numbered here from 3 to 8.

3. Take code example for drawing lines.
4. Test the selected code-example to display a simple graph.
5. Reuse a simple code-example of PHP class concept for creating a WaveGraph.
6. Modify the created class to have the graph attributes (String, int, Array, image) and group the structural code to different functions (Coloring, axis, lines).
7. Extend the WaveGraph class by adding functions for writing strings on the x-axis (Process), on the y-axis (activity), intersection lines from-to (for tasks), and for having extra dimensions
8. Experiment with the new graph image output.

4.2 Case Studies

The case-studies, which are listed in Table 2, were conducted as an action research to practice the methodologies of selection, reuse, modify, extend, test, and contribution processes. In that way we could also record those tasks in the prototype database to display them as a WPM graph. The case-studies produce plenty of information and WPM graph for

deploying different OSS products, gathering internal-requirement, customization and plug-in development over existing OrangeHRM OSS for human resource management. The WPM graph had different forms like a sound wave or sometimes like a heart bit.

Table 2: Activity Samples

Case Study	*Activity Example*
Deployment	Installation of Ubuntu, Dspace, CVS and OrangeHRM
Requirement	Customization request of different user-interface and realization of payroll module from baseline requirement.
Design	Extending the payroll module for each Model, View and Controller by following the design pattern in the OrangeHRM OSS.
Implementation	Modifying the source-code for each customization request. Extending the source-code by reusing existing code-examples in the OrangeHRM itself.

5. Related Work

For WPM, it was not possible to find a related work, since this is a new concept. We have presented somehow related works for OSSD.

In our literature review "Do Best Practice Frameworks fit Open Source Software Customization" [5], we have seen the efforts made to use existing frameworks for the open-source customization in case of ITIL, SPICE and V-Model.

Another thing we discussed is the part which tried to use agile method for their open source further development activities in the thesis research paper of "Using Open-Source Solutions in Agile Software Development" [9].

What we find is also a generalized process modeling in the form of UML for describing the OSS development activities [10].

The above works are nearly related to our specific objective of our research, which we have already included them in the literature review part.

6. Conclusion

We have validated that WPM can be used to detect activities and to control what we were doing. More precisely, we described the further development process using WPM. In finding out the types of activities, in case of internal requirement gathering, which is inspecting and then feedback, lets us to see a methodology of implementing the requirement as they came. The other methodology we identified to have the activities per the design pattern is also another methodology how the design concept can be an input for the activities. The data selected by the case studies are not for evaluating purpose but for describing the WPM concept as a whole and that the WPM is a valued concept. As a future work, the WPM concept can be researched to apply it for other types of software development processes or for any kind of work process management and controlling.

References

[1] Gautam Guliani and Dan Wood, "Open Source for the Enterprise", O'Reilly, July 2005.

[2] Jeremy Lewis, "Software Development Life Cycle (SDLC) 100 Most Asked Questions, SDLC Methodologies, Tools, Process and Business Models", 2008.

[3] Sommer Vill, "Software Engineering", 8th Edition, Addison-Wesley, 2007.

[4] Jean Claude Derniame, Badara Ali Kaba, and David Wastell, "Software Process Model Principles, Methodology and Technology", Lecture Notes in Computer Science, Springer Verlag, 1999.

[5] Keßler Steffen and Alpar, Pau, "Do Best Practice Frameworks Fit Open Source Software Customization?", Institute of Information Systems, Philipps-Universität Marburg, Germany, 2009.

[6] Markus Pizka, "Adaptation of Large-Scale Open Source Software - An Experience Report", Technische Universitaet Muenchen, Institut Fuer Informatik, Munich, 2004.

[7] Life cycle Model, The concept behind the design of Software Life Cycle Model http://www.chambers.com.au/sample_p/c_pmodel.htm, Last accessed on April 05 2012.

[8] Free/Open Source Software Development: Results and Research Methods Master's Thesis, Otso Kivekaes, University of Helsinki, Department of Computer Science, June 1, 2008.

[9] Tuukka Haapasalo, "Using Open-Source Solutions in Agile Software Development", Helsinki University of Technology, Department of Computer Science and Engineering Laboratory of Software Technology, January 17, 2007.

[10] Stefan Dietze, "Modell und Optimierungsansatz für Open Source-Softwareentwicklungsprozesse", Potsdam University, 2004.

[11] Adoption of Open Source Software in Software-Intensive Industry, Department of Computer and Information Science, Norwegian University, Øyvind Hauge, NTNU-trykk, Trondheim, June 17th, 2010.

[12] Walt Scacchi, "Process Models in Software Engineering", Institute for Software Research, University of California, Irvine, J.J. Marciniak (ed.), Encyclopedia of Software Engineering, 2nd Edition, John Wiley and Sons, December 2001.

Semantic Web-Based Digital Business Ecosystem: The Case of Amharic Semantic Blogging

Eyoss Girma
MFI Trading and Office Solutions/Alcatel-Lucent Department, Addis Ababa, Ethiopia
eyosg@yahoo.com

Workshet Lamenew
School of Information Science, Addis Ababa University, Ethiopia
workshet@gmail.com

Abstract

While few researches are conducted elsewhere, semantic blogging specific to a local language such as Amharic was not given an attention. This research is aimed at investigating the current situation of business ecosystem in Ethiopia with the view to design and develop Amharic semantic web-based blogging system in web-based DBE.

Various methodologies are used to gather primary and secondary data from IT-based SMEs like profiles, events, and items. Classes, subclasses, and the relationships between concepts are defined and an architecture of semantic blog searching for IT-based SMEs is developed.

The proposed semantic blog searching architecture consists of four modules: preprocessing, query processor, knowledgebase, and crawler. The preprocessing module handles most of the Amharic unique features. The query processor module lets the user enter queries in Amharic and retrieve the relevant result for the given query. The knowledgebase module handles the ontological structures of the business of concepts. The crawler component traverses through the World Wide Web and fetches those web page blogs with Amharic content.

The query and retrieval of the semantic blogging system is measured in terms of precision and recall and the summary of the results are presented. The study shows that the use of Amharic semantic blog searcher results in 76 % accuracy which is a promising outcome.

Keywords: Digital Business Ecosystem; Small and Medium Enterprise; Semantics; Ontology

1. Introduction

Semantic web services are designed to create easy communication between computers and humans so that they can accomplish certain tasks and services. These services are often used in digital environments - so called Digital Business Ecosystems to implement and organize services of businesses like SME's (Small and Medium Enterprises) interaction for demand and supply.

SMEs can lead to effective knowledge creation and growth. Relevant information at the right time could provide SMEs with the appropriate tools to make more economically sound decisions. This is the main objective of the paper: "To investigate the current situation of IT-based SMEs business ecosystem in Ethiopia with the view to design and develop architecture for Amharic semantic web-based blogging system in web-based DBE".

The rest of the paper is organized as follows. Section 2 covers literature review which includes the description of SMEs, semantic web-based digital business ecosystem and its major components. Section 3 covers the review of related works. Section 4 covers the architectural design and implementation of the proposed architecture for semantic blogging system and Amharic blog searching system in digital business ecosystem. Section 5 presents the experimental result of the proposed design for semantic blogging and searching in digital business ecosystem for IT-based SMEs. Finally Section 6 presents the result and conclusion.

2. Background

The success of an SME in the Business Ecosystem depends on its ability to find the right partners in the Business Ecosystem, to operate across the Business Ecosystem. SMEs are increasingly adopting Internet technologies and are recognizing the opportunities offered by e-Commerce and use it to offer their products and services online.

The Digital Business Ecosystem is a special Internet-based environment in which businesses can interact with each other in effective and efficient ways. Being part of the Digital Business Ecosystem means that we are aware of the range of products and services available from all of the other partners and can easily match them with our business requirements.

The exposure is twofold which means our products and services are also being showcased to other SMEs so they can identify us as potential business partners. The Digital Business Ecosystem brings together the best of all of the Internet technologies, tools and applications as well as legal, business and revenue information that SMEs need if they are to have the competitive edge in the market.

Blogging's greatest benefit is in strengthening the social and business relationships in social and DBE's environment of the world society. First, ease of use makes it likely that more people will publish and publish more often, and that more information will be communicated. The structure of the information is often different from more static home pages, more like online journals.

There are around 80 languages in Ethiopia which are used in day-to-day communication. Although many languages are spoken in Ethiopia, Amharic is the dominant one since it is spoken by the substantial part of the Ethiopian population. Based on Tessema Mindaye [7], Amharic language is the working language of the Federal government and it is a Semitic Language of the Afro-Asiatic language group that is related to Hebrew, Arabic, and Syrian.

The unique feature of Amharic language needs to be handled properly to be used in the DBEs. Handling these features is mandatory so that the input texts are processed properly before they are manipulated in different steps of natural language processing and stored in the index or retrieved from a database.

3. Related Work

This section covers the various researches and projects done in the areas of digital business ecosystem [3], web based semantics and semantic blogging systems [2, 5], Amharic language search engine [7], and document categorization in Amharic language [6].

Batra and Salzburg [2] made some development in the area of ontology systems, specifically in the field of medicine which refers to drugs, practice or diagnosis. Chang and Boley [3] made a comparison of different forms of DBE with most advanced communication platforms or environments such as client-server, P2P and web services. They also described how digital ecosystems can benefit from semantic web ontologies. When we come to the language of Amharic, Surafel Lemma [4] worked on semantics in the area of multimedia and the respective web services and prepared a prototype which allows a user to advertise a web service using the ontology and look for a specific Web service. Meron Sahlemariam [6] has also attempted to look into the techniques of automatic classification and the study focuses in categorizing a given Amharic document into a predefined category by passing it through the preprocessing and classification processes.

Blogging is one component of digital ecosystem and there are multiple researches made in this area and one of them is by Cayzer [5] with the focus on semantic blogging and decentralized knowledge management which develops architecture for importing, export, view, navigate and query the ontology repository. This architecture is the starting point for our digital business ecosystem architecture. The author also described the impossibility of future blogging without semantics.

A project called Digital Ecosystem for Agriculture and Rural Livelihood (DEAL) in [1] was an implementation of web based initiative that coordinates back-end infrastructure, media technology, and knowledge in order to make agricultural content accessible through multiple channels in rural India.

The development of DBE throughout EU and Asia is aggressive as compared to that of Africa. As seen from the related works, its value for the development of SME in the developing countries is priceless.

4. The Proposed Solution

4.1 Architecture of Semantic Blogging

The major goal of this research work is to make use of the content related semantic metadata which are relevant in the DBE environment of blogging and specially in Amharic language. Figure 1 shows the architecture of the proposed semantic blogging web-based system. The semantic view is the part of the architecture which is used to view the blog content in the way that helps the semantically enriched blog metadata to the user.

The semantic navigation part of the architecture helps to efficiently find blog items of our interest. For ease of navigation, explorer or tree type interface is used to browse related items and users can also follow labeled links.

One part of this architecture is semantic query. We used SPARQL as a language to test the retrieval and manipulation of the data stored in OWL or RDF format.

In our consideration of import and export for a blog, we took the implementation through RSS (Really Simple Syndication) with the help from W3C schools RSS standards [8], which is used to distribute up-to-date web content from one web site to thousands of other web sites around the world. The advantage is that it will reduce the time spent on sites looking for news, events, and updates.

The option of using RSS as a file to import and export the blog is possible through companies called aggregators which need registration and this searches the registered web sites for RSS documents. The aggregator also verifies the link and displays information about the feeds so clients can link to documents that interest them.

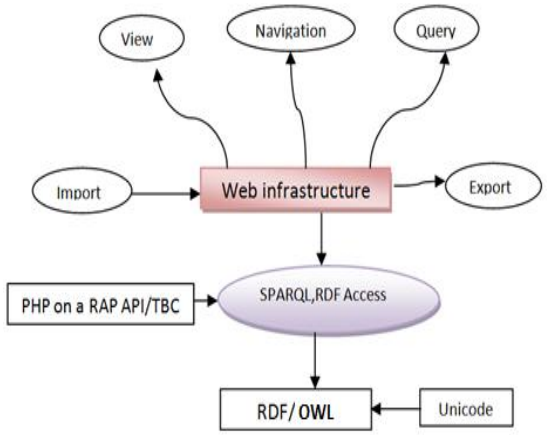


Figure 1: Revised Architecture of Semantic Blogging

4.2 Architecture of semantic Amharic blog Searching in Digital Business Ecosystem

Blog searching system architecture of a specific language requires a crawler, preprocessing, query analyzer and indexing as basic components. This session will cover the part which explains the architecture of Amharic blog searching and display.

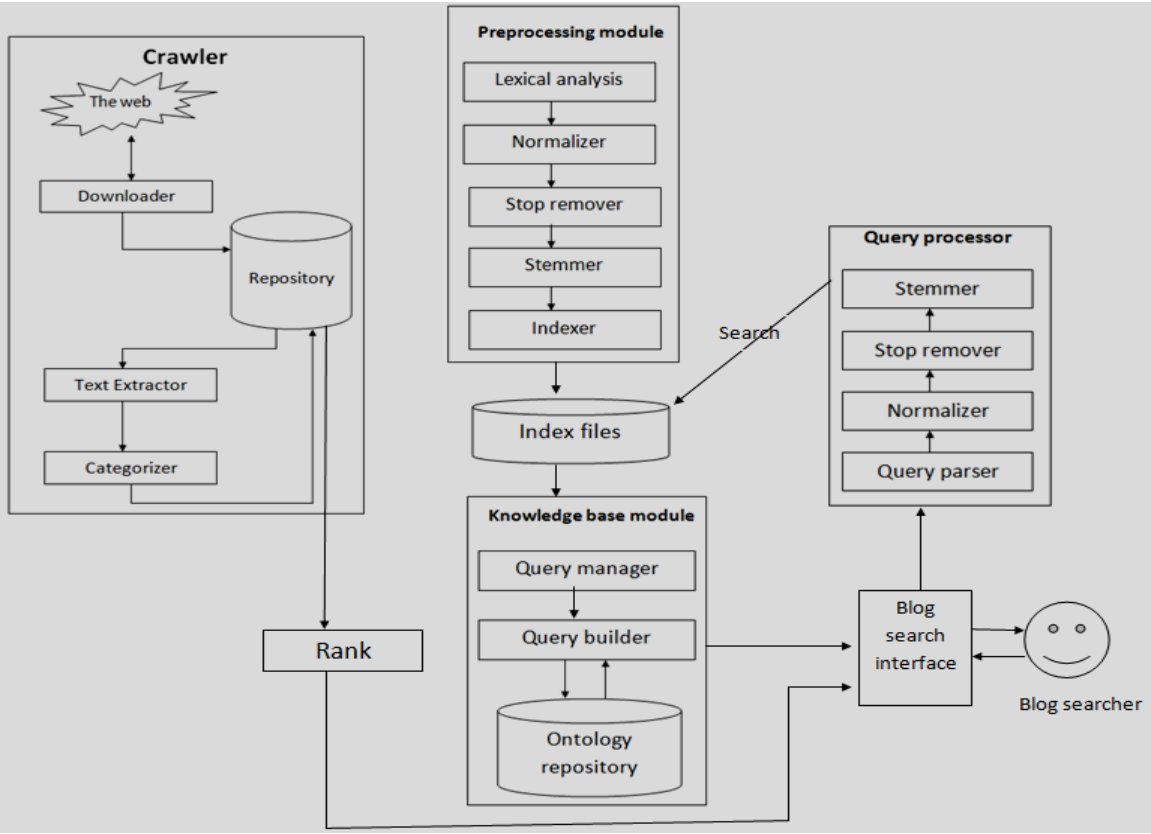


Figure 2: Semantic Amharic Blog Searching Architecture

The Semantic Amharic Blog Searching Architecture is shown in Figure 2. The Preprocessing module handles most of the Amharic language unique features. The input texts from blog users and the crawler are passed through different processes before they are stored in the index. The character streams are tokenized into words by taking white space and Amharic punctuation marks as word demarcations. Shorter forms of a word are replaced by its expanded form. Then, the word is checked for stop words. If the word is a stop word or if it is a variant of a stop word, it is removed.

The remaining non-stop words are stemmed to reduce them to their common form. The final result of all these processes is stored as an index in a structure that is appropriate for fast searching. The process of preprocessing module includes tokenization handling the repetitive alphabets, short words and stemming.

The tokeniser splits the text into very simple tokens such as numbers, punctuation and words of different types. In Amharic language, there are multiple punctuation marks that are used in between and at the end of sentences. For instance, colon ":" in English is called "Hulet Netib" in Amharic which is used to separate words. The other mark is ፣ "Netela Serez" which is used as a coma in English language where in case of Amharic language it demarcates a list of things, items, etc. Two colons side by side "::" "Arat Netib" is used at the end of a sentence and demarcates the end of a sentence. This tokenization component and the algorithm are adopted from Tesemma's work [7] which is designed to split words by identifying punctuation marks and white spaces.

The Amharic indexer has two inputs. One of the inputs is for the crawler component which passes through the normalizer, stop word remover, stemmer, and indexer. The other is for the input text form the blog searcher which passes through similar components.

The index repository is the component that stores (indexes) a word (term) with some necessary information such as Term Frequency (TF), Inverse Document Frequency (IDF), etc. We also used Lucene for this purpose and Lucene has an open source IR library which plays a pivotal role in the indexing process. One of the tasks in indexing component of our blog search is to integrate the

Amharic language features with that of Lucene library with the application of GATE tool.

The query processor module accepts input queries from users through an interface. This module lets the user enter queries in Amharic and retrieve the relevant documents for the given query.

Lucene has an important role to play in this module. One of the tasks in the indexing component of our blog searching is to integrate the Amharic language features with that of Lucene library. GATE has a data store to store language resources. Data stores are an abstract model of disk-based persistence, which can be implemented by various types of storage mechanisms. Lucene's data store is a full-featured annotation indexing and retrieval system. It is provided as part of an extension of the serial data stores.

The knowledgebase module serves as a knowledgebase to define the ontological structures of the business of SMEs. Knowledge representation in the ontology is mainly based on the concepts. The concepts are structured in the ontology as class, sub-classes, and relationships between classes. The ontology is like a database which constitutes the concepts, words, and categories and how they relate to each other.

The ontology contains the represented knowledge in the domain and this is designed in a way to access and use the knowledgebase in the ontology. We, therefore, mapped the concepts and the relationships between them with textual expressions. Using this relationship, we can query the semantic relations found among concepts.

The crawler is directly adopted from Tesemma's work [7]. The advantage of adopting the work is that the author developed a language-focused crawler which has the capacity to traverse through a country domain (ccTLD) which is a way to get web pages of a certain language. The crawler component traverses through the World Wide Web and fetches those web page blogs with Amharic content. The crawler has two sub components: the Crawler and Categorizer. The crawler is responsible for downloading a page from a web site and the categorizer will categorize the web as Amharic blogs or not. The Amharic web blogs are processed and the rest are discarded.

5. Prototype

GATE tool is used to customize the crawler. It is based on Websphinx, a JAVA-based, customizable, multi-threaded web crawler, which can be customized based on our requirement.

The basic idea is to specify a source URL of a blog web site (or set of documents created from web URLs) and a depth and maximum number of documents to build the initial data upon which further processing could be done. The PR itself provides a number of other parameters to regulate the crawling. In our experiment, we used blog pages that we developed through Dream Weaver and we also used Wamp server as our web server.

There are no well organized blogging sites in Amharic language so we needed to include blog web construction by our own. The blog web construction is based on the data we collected from web sites written in Amharic language like www. ethiopianreporter.com/, http://www. waltainfo.com/, and www.mcit.gov.eg/.

The knowledgebase is constructed based on the documents collected from IT-based SMEs with a focus on the organizational structure, services, and products.

Based on the above example, we have made our experiment on 21 blogs with a total of 3014 words and 26 sentences. Test was made for 8 queries.

5.1 Evaluation

Evaluation of the blog searching is done with the evaluation parameter that compares the number of web blogs which are classified correctly and incorrectly. Typically, the comparison is done based on the blogs which are retrieved based on the search term correctly and incorrectly.

Precision and recall, which are the evaluation parameters of Information Retrieval, are used in web based blog searching. In pattern recognition and information retrieval, precision is the fraction of retrieved instances that are relevant, while recall is the fraction of relevant instances that are retrieved. Both precision and recall are, therefore, based on an understanding and measure of relevance. When a search engine returns 30 pages only 20 of which were relevant while failing to return 40 additional

relevant pages, its precision is 20/30 = 2/3 while its recall is 20/60 = 1/3.

Precision =Tot-Ret-Rel ÷ Tot-Ret

Recall = Tot-Ret-Rel ÷ Tot-Rel

where Tot-Ret-Rel is the total number of returned documents that are relevant to the query, Tot-Ret is the total number of documents that are retrieved by the system, and Tot-Rel is the total number of documents that are relevant to the query

5.2 Result

The experiments are carried out on the Amharic web blog documents collected from the crawler and we used Amharic search terms and recorded the result of the retrieval.

Table 1: Blog Search Result

Query	Tot-Rel	Tot-Ret	Tot-Ret-Rel	Tot-Ret-Ire	Precision	Recall	Accuracy Percentage %
ኢንፎርሜሽን	5	5	5	0	1	1	100
ኢንፎርሜሽን ኮሙዩኒኬሽን ቴክኖሎጂ	4	7	2	5	0.28	0.5	58
ሞባይል ስልክ	4	9	3	6	0.33	0.75	60
ሶፍትዌር መፍጠር	1	0	0	0	0	0	0
ሳተላይት	5	4	4	0	1	0.8	100
አዲስ አበባ ዩኒቨርሲቲ	6	6	5	1	0.84	0.84	85
ኤሌክትሮኒክስ	2	2	2	0	1	1	100
አይሲቲ ልቀት ማዕከል	3	5	3	2	0.6	1	71
ቴክኖሎጂ	12	14	9	5	0.64	0.75	73
ፀረ ኮምፒውተር ቫይረስ	5	5	5	0	1	1	100
አፍሪካ	10	11	9	2	0.81	0.9	85
Average							75.63636364

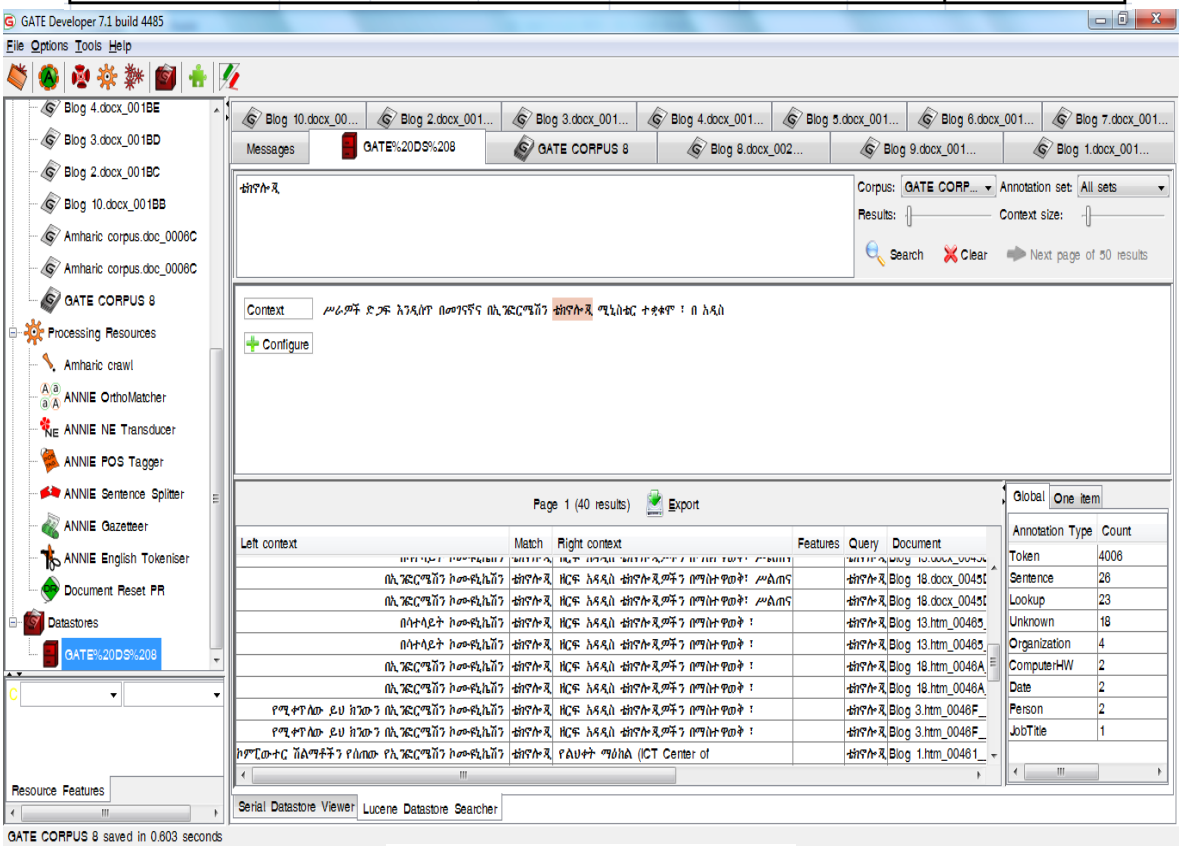


Figure 3: Query result for ቴክኖሎጂ

The experiment was made on 45 blogs with a total of 7114 words (see Table 1 and Figure 3 for a screenshot of an example query). The test was made for 11 queries. Based on result, out of 11 queries 4 of the results were found to have 100% precision and 3 of the queries were observed to have precision less than 50%. Ten of the recall values show a query result of 50% and above.

Average precision and average recall for combination of words which are associated with the "AND" operator is 51% and 68%, respectively. For single word query the average precision and recall is 89% and 89%, respectively.

6. Conclusion

A digital ecosystem uses the working principles of nature's ecosystem which will have a network of co-existing elements that depend on each other in order to survive. Every Small-to-Medium Enterprise is part of this business ecosystem working together with many other elements in the system like other SMEs and clients.

There are two architectures in our design. One of the architecture is the proposed semantic blogging web-based system with major components of import, export, view, navigation, and query. The other architecture is Blog searching system architecture for Amharic which has crawler, preprocessing, query analyzer and indexer as basic components. Amharic language has many distinct features that affect the information retrieval of the language's documents on the Web.

An experiment is conducted using GATE tool by customizing the crawler to gather the required documents. The experiment uses blog pages that we developed and we also used Wamp server as our web server. The blog searching system is found to have a better retrieval than general purpose blog search engines.

References

[1] Francesco Nachira and Andrea Nicolai, "Digital Business Ecosystems", The European Commission, 2010.

[2] Sudhir Batra and F. H. Salzburg, "Semantic Web Services in Digital Ecosystems", *Proceedings of the Digital EcoSystems and Technologies Conference*, Canada, 21-23 Feb 2007.

[3] Elizabeth Chang and Harold Boley, "Digital Ecosystems: Principles and Semantics", *in Proceedings of the 2007 Inaugural IEEE International Conference on Digital Ecosystems on Technologies*, Aug 14, 2007.

[4] Surafel Lemma, "Semantic Description of Multimedia Content Adaptation Web Services", Unpublished Master's Thesis, Addis Ababa University, 2005.

[5] Steve Cayzer, "Semantic Blogging and Decentralized Knowledge Management", *Communications of the ACM Journal*, Vol. 47, No. 12, December 2004.

[6] Meron Sahlemariam, **"**Concept-Based Automatic Amharic Document Categorization", Unpublished Master's Thesis, Addis Ababa University, January 2009.

[7] Tessema Mindaye, "Design and Implementation of Amharic Search Engine", Unpublished Master's Thesis, Addis Ababa University, July 2007.

[8] "RSS basic," Retrieved from http://www.w3schools.com/rss/rss_intro.asp, last accessed on Oct 04, 2012.

Data Exchange Using Web Services: The Cases of Ethiopian Revenue and Customs Authority and Transport Authority

Firesew Mikreyohannes
Custor Computing, Addis Ababa, Ethiopia
firesew@gmail.com

Abrehet Mohammed Omer
Department of Computer Science and IT, Addis Ababa Science and Technology University, Ethiopia
abrehet@gmail.com

Abstract

At present, the use of the Internet and its volume of information is growing at an alarming rate throughout the world. In particular, the Internet has contributed to the emergence of different applications that are being developed using different tools. Recognizing the significant role of these applications, organizations tend to develop their own domain-specific applications that automate their day-to-day activities. However, the heterogeneity of these different systems represents a major obstacle to data accessibility and data exchange. This paper presents the proposed approach to address such kind of challenges.

Ethiopia is currently well aware of its present position and putting all rounded efforts to improve the services provided through the use of information and communication technology. As a result, the present portal systems or e-Government service initiatives are expected to help it leapfrog ahead towards new development. Some of the organizations are engaged in the process of design and development of web based applications that automate the service delivery. These applications are expected to be interoperable with other autonomous systems. Despite the efforts made by different public bodies to automate their operations, integration of such automated systems remains a major challenge.

This paper intends to deal with such kind of challenges by selecting specific cases with data exchange problem that demonstrates the methodology followed to find a solution. To do so, first the current efforts and technologies towards integration of different systems have been assessed. Then information about existing systems and system requirements for data exchange is gathered. Based on the assessment and system requirement, a system is designed and a prototype is implemented with a selected technology.

Keywords: Data Exchange; Web Service; Heterogeneous Data

1. Introduction

Applications are now being developed by different organizations to facilitate different services. The scope of these applications may not cover all requirements of the organizations. In addition, the output of these applications might be required by other organizations. However, the schema and data structure of these applications are usually different and data exchange among organizations is a major problem. The Transport Authority (TA), now called Road and Transport Bureau was established through Proclamation No 15/2009 in 2003. It provides vehicle registration, driving license renewal, vehicle inspection, vehicle ownership transfer, plate issuance, vehicle value estimation, and other services [2].

The Ethiopian Revenue and Customs Authority (ERCA) was established through the Proclamation No. 587/2008. The Authority came into existence on 14 July 2008 and is responsible for collecting revenue from customs duties and domestic taxes [3].

This paper focuses on assessing and identifying existing problems associated with services provided by TA and ERCA such as vehicle registration, ownership transfer, and tax collection with a view towards improving the service process efficiently. In order to achieve this, existing methods and technologies are assessed and compared based on their opportunities and limitations for creating enabling environment for making efficient and flexible data exchange among applications. Moreover, a web service-based application is designed and implemented to demonstrate the

selected technology applicability for efficient and flexible data exchange in heterogeneous environments.

Applications are now being developed by different organizations to facilitate different services. The scope of these applications may not cover all requirements of the organizations. In addition, the output of these applications might be required by other organizations. However, the schema and data structure of these applications are usually different and data exchange among organizations is a major problem.

TA requires declaration of data as well as proof of tax settlement from ERCA for every vehicle imported as a precondition for providing vehicle registration services. TA requires vehicle's declaration information directly from ERCA not only to use it as an input to register a vehicle, but also to check whether the vehicle imported has gone through all legal procedures. Since both organizations use different applications and data storage systems, this has been creating a bottleneck for exchanging data.

At present, data exchange regarding vehicle data is performed manually. Customers are required to bring hardcopy documents to the TA to register their vehicle. TA also requires an official confirmation document directly from ERCA. To complicate matters, the TA system is distributed over different sites that are disconnected to each other. According to discussions held with ERCA staff, ERCA have been using Automated System for Customs and Data (ASYCUDA++) for over ten years. As it is a very old system, it neither allows integrating with other systems nor extracting information to provide service to other stakeholders like TA is possible. This forced ERCA to extract information manually. But, ERCA has now a plan to upgrade the existing system to a modern system.

The lack of data exchange between these two organization's systems has created the following problems:

- Inconvenience to customers due to delay of service delivery
- Use of forged vehicle's declaration that leads to illegal use of a vehicle
- TA is unable to get up-to-date and genuine vehicle's imported date as a result of disparate systems

2. Related Work

In today's software development world, multiple vendors have been implementing their own data model through practicing the relational theory. Therefore, the data models dealing with the same real world objects are often different. The data exchange between these different software systems is extremely difficult due to the difference of the data model. To resolve the data exchange problem in integration and information sharing among different software systems, XML (Extensible Markup Language) becomes a natural selection as a means to relay the data in the information exchange process [5].

In this context, data is transferred between two sample school administration database systems, which are semantically similar, but have different data models. The transferring is carried out through an intermediate data model as the bridge between these two data sources. The intermediate data model is in XML data format. The mappings are via two XML schemas, one from the source school database to the intermediate data file, another is from the intermediate to the destination school database. The programs are developed using Microsoft .NET and C# programming language. The databases are built on Microsoft SQL server. As shown in Figures 3 and 4, this paper presents two heterogeneous database systems: School A and School B, and data needs to be transferred from one to the other [5].

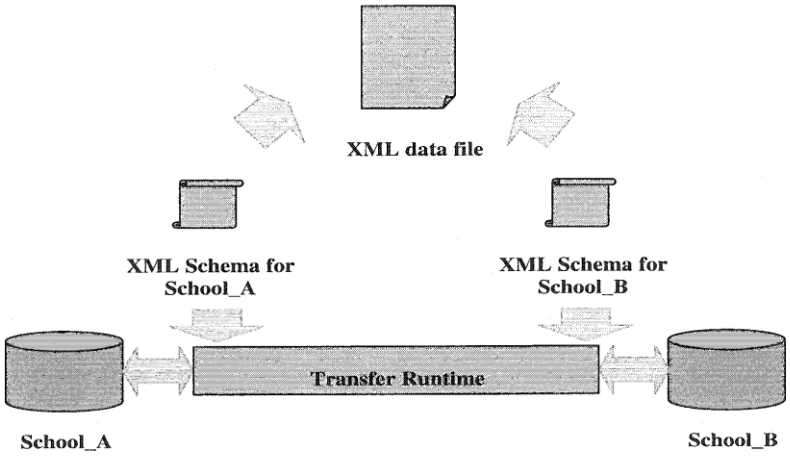


Figure 1: System Architecture

In order to solve the data transferring between the heterogeneous schema problems of these schools, the selected solution is to build a middle-tier data model as a bridge between the transferring and receiving side. The middle-tier data model complies with the semantic meaning of this specific application. It is responsible for the mapping from the source to the destination schema, providing a representation of the relevant data relationship on the transportation pipeline. This middle-tier data model is represented by XML, in particular, an XML data file, and two XML Schemas, each of which is responsible for matching and extracting data from the source data, and mapping and transferring the data to the destined data source [5].

3. Solution Design

TA requires declaration of data as well as proof of tax settlement from ERCA for every vehicle imported as a precondition for providing vehicle registration services. TA requires vehicle's declaration information directly from ERCA not only to use it as an input to register a vehicle, but also to check whether the vehicle imported has gone through all legal procedures. Since both organizations use different applications and data storage systems, this has been creating a bottleneck for exchanging data.

Since both TA and ERCA are currently using a WAN and eventually implement a network using fiber optics as a backup line, availability will not be an issue. In addition, the matching requirement can be addressed through implementing a data mapping for TA and ERCA independently. Therefore, multiple approaches for data exchange have evolved over time. However, the web service is chosen for the following main benefits [1]:

- Hide heterogeneity by developing an application that allows data exchange of heterogeneous systems
- Provides valuable resources, which are data sources that are available for users
- Accesses and retrieves data from multiple data sources at the shortest possible time and lowest cost (such as decreasing transportation, resource cost, etc.)
- Promotes innovation and potential new data uses
- Leads to new collaborations between data providers and data consumers in distributed computing

The architecture chosen for data exchange using Web Services is a combination of architectures since both TA and ERCA have their own heterogeneous systems. Both organizations use three-layered architecture; but the web service architecture that integrates the distributed systems is different. In distributed systems, there are three types of commonly used architectures. These architectures are considered for this web service solution.

The architecture is based on three layered service applications: presentation, business, and data. This pattern presents an overview of the responsibilities of each layer and the components that compose each layer. These are [6]:

- Presentation Layer: provides the application's user interface. Typically, this involves the use of Windows Forms for smart client interaction, and ASP.NET technologies for browser-based interaction.
- Business Layer: implements the business functionality of the application. The domain layer is typically composed of a number of components implemented using one or more .NET-enabled programming languages. These components may be augmented with Microsoft .NET Enterprise Services for scalable distributed component solutions.
- Data Layer: provides access to external systems such as databases. The primary .NET technology involved at this layer is ADO.NET. However, it is not uncommon to use some .NET XML capabilities here as well.

The architecture, as depicted in Figure 2, is well suited because it separates the business logic from the presentation or data access layers. A given layer can be changed without significantly affecting the other tiers and any change to the business requirements and policies can also be easily accommodated.

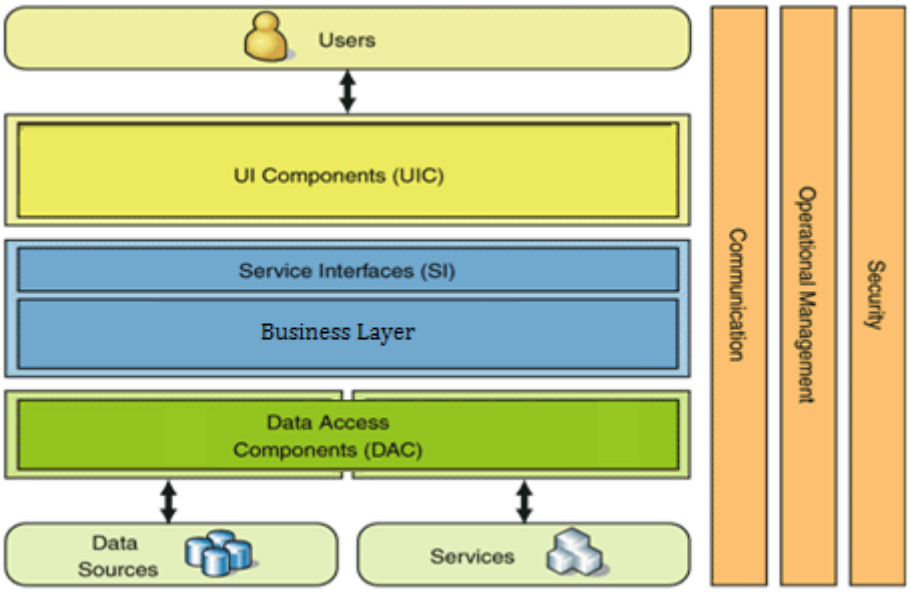


Figure 2: Three-Layered Services Application

The service oriented architecture works on the assumption that the functionality made available by an organization will be exposed as a service. In middleware terms, a service is a procedure, method, or object with a stable, published interface that can be invoked by clients. The interactions between organizations occur in a peer-to-peer fashion where each party exposes its internal operations as (Web) services, which therefore act as entry points to the local information systems [4].

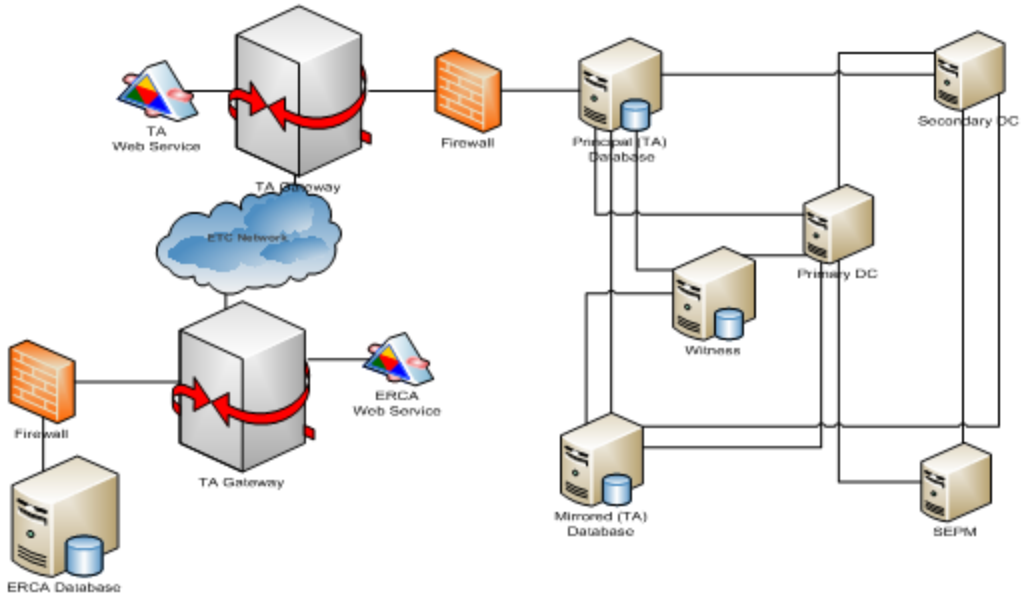


Figure 3: Deployment diagram for DEWS

Likewise, the TA web calls a service at ERCA through the TA business. The TA business is responsible for serializing the request and sends the SOAP request to the ERCA service. The ERCA service transfers the request to the ERCA business to de-serialize the request and queries the ERCA database. Once the ERCA business retrieves a record from the database, it serializes the response to send it back to the consumer as a service response. Finally, the TA business de-serializes the service response to map the data to the corresponding objects. Since the data exchange between the TA and ERCA is bi-directional, the process from ERCA side also works in a similar manner.

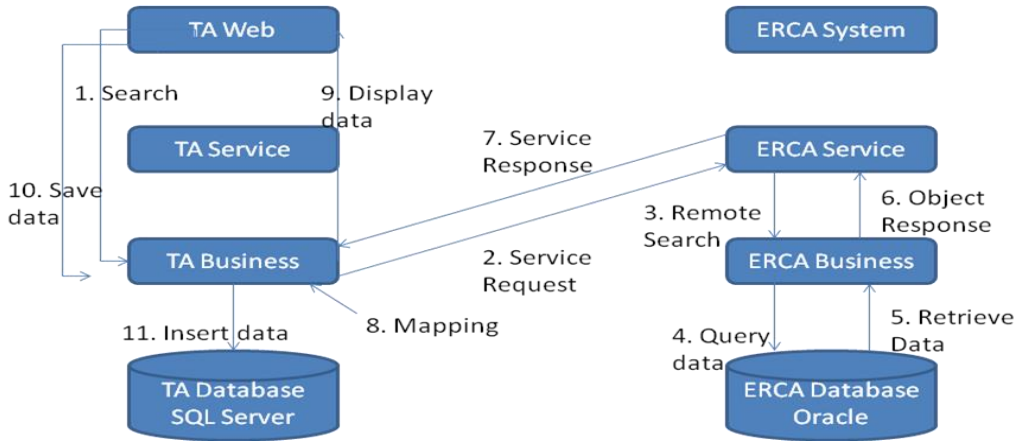


Figure 4: Implementation Architecture of DEWS

4. Conclusion and Recommendations

This paper aims at exploring technologies for data exchange and addressing the problem of data exchange between one of the branches of TA and ERCA. Out of the different approaches and implementation technologies explored, web service-based data exchange mechanism is implemented for the vehicle registration and ownership transfer processes. The prototype developed in this paper demonstrates how a service implementation facilitates exchange of data between TA and ERCA through the use of their systems. Service providers of the TA can search for a vehicle directly from ERCA to carry out the vehicle registration process. It would also allow ERCA to follow-up a vehicle ownership status and tax collection.

In addition, the TA has currently five branches where vehicle registration and other services are given. But, the work in this paper must be somehow modified with further analysis and minor modification of the existing systems on each branch

to provide a comprehensive solution for sharing data sources that hides the heterogeneity between different data sources and between ERCA and other stakeholders as well.

Generally, organizations still tend to automate their process without worrying to interface with other systems. But when there is a requirement to interface with other organization's systems or services, the same project is done over and over again. This has an effect on the organization's time, cost and other resources as well. Therefore, it is recommended that interfacing issues like network infrastructure, security and, particularly they type of data that might be required by other stakeholders, must always be taken into consideration up on project initiation and planning.

The result of this work mainly alleviates processes related to initial vehicle registration and tax collection of ownership transfer of duty free vehicles which are the existing major problems of TA and ERCA, respectively. However, since both organizations might have additional branches which are not considered in this work, adapting the web service to incorporate requirements of these branches and other stakeholders can be further studied.

In addition, due to the time limitation, this work is developed using a standard web service with the implementation of authentication and authorization. Other security requirements like confidentiality, integrity, auditing, and availability must be implemented in order to minimize a risk on both organizations.

Therefore, expansion of the web service to accommodate the requirements of the remaining branches of TA and other stakeholders and consideration of other security aspects are open for future work.

References

[1] Sofien Gannouni, Hassan Mathkour, and Mutaz Beraka, "A comparative Survey of Data Sharing Approaches and their Applications in Distributed Computing Environment", *Journal of Theoretical and Applied Information Technology*, Vol. 33, No. 1, pp. 42-53, November 15, 2011.

[2] City Government of Addis Ababa, http://www.addisababacity.gov.et/index.php/en/economic-sector/road-transport-bureau, last accessed on November 08, 2012.

[3] Ethiopian Revenue and Customs Authority, http://www.erca.gov.et/index.jsp?id=aboutus, last accessed on September 04, 2012

[4] Gustavo Alonoso, Fabio Casati, Harumi Kuno, and Vijay Machiraju, "Web Services-Concepts, Architectures and Applications", Springer Verlag, Berlin Heidelberg, Vol. 33, No. 1, pp. 125, 2004.

[5] Qinghong Lin, "XML Data Transfer Between Heterogeneous Database Systems", Concordia University, September 2003.

[6] Three Layered Service Application, http://msdn.microsoft.com/en-us/library/ff648105.aspx?rssCatalog, last accessed on January 05, 2013.

A Framework for Integrating Software Usability into Software Development Process

Hayat Dino
AFRICOM Technologies, Addis Ababa, Ethiopia
hayudb@gmail.com

Rahel Bekele
School of Information Science, Addis Ababa University, Ethiopia
rahtesf@yahoo.com

Abstract

This paper reviews the status of software development companies to what degree they adapt formal software process model and perceive & integrate usability throughout the development process. The researchers believe that usability must be tightly integrated in the software development process and makes its objective to examine how usability can be integrated more tightly into a specific software development process model that is being applied by the majority of software development companies. Process models proposed by software engineering and some of user-centered design models proposed by usability engineering are discussed. Moreover, literature study, related works, survey and its findings are discussed. Finally, a framework is developed for use by software development teams wishing to integrate user-centered design practices with a particular software development process model by adding some additional features.

Keywords: Software Development; Software Process Model; Usability; Software Engineering; Usability Engineering; User-Centered Design

1. Introduction

As many research works show, usability should be considered throughout the development process, from the requirement gathering phase to the deployment of the product. It might also be considered further. This is because product of good usability can only be achieved if its development process integrates usability. Integration of usability should be considered by the all members who are involved in the software development process. From software development company perspectives, developing software which is usable can increase sales and productivity. From customers' point of view, it makes the end-user satisfied, also increases the productivity. Bear in mind that companies developing software have to make sure that their development process meets usability requirements. Customers should not only demand software which provides the main functionality but also they have to take into account usability as a critical feature.

Along with the concept of usability, there is the Usability Engineering (UE) discipline that is concerned with the question of how to design software that is easy to use (usable). UE is an approach to the development of software and systems which involves user participation from the outset and guarantees the efficacy of the product through the use of usability specification and metrics [1]. As an integral part of software engineering, UE seeks to understand the client's, customer's, and users' wants and needs; translating these needs into system requirements; and designing a system to satisfy these requirements and testing to help ensure their realization in the final product. However, at each level of software development process, the two disciplines have many differences such as levels of iteration and evaluation, terminology, requirements representation, activities, techniques, timelines, scope, roles, procedures, and focus as reported in [2]. Many researchers have designed UE process models which help to develop software which is usable and user-centered. Some of the models will be discussed in the next section.

All software applications developed based on SE and UE lifecycles are tools that help users accomplish certain tasks. However, before developing a software product or designing UI, one should consider information about the people who will use the tool such as who are the system users? What will they need to accomplish? What will they

need from the system to accomplish this? How should the system supply what they need? [3].

2. Statement of the Problem

Research results [2, 4, 5, 6] in the past few years recommend for software organizations, developers, user interface designers, and other professionals involved in developing software to integrate software usability on software development lifecycle.

Rubin and Chisnel [7] also indicated that usability is a quality that many products possess, but many, many more lack. The lack of usability may lead the end-user to need further assistance and having bad perception about the application or the software system, in essence that the end-user becomes less satisfied to use the application. As an example, let us take mobile devices, in which people sometimes lack knowledge about how to use those devices. For instance, one might not know where to set an alarm clock, how to browse the Internet using the device, even how to switch off can be difficult. Hence, these kinds of usability issues should be considered in another case as well.

If the end-users find the software too difficult to understand, then an otherwise excellent product could be ruined to failure. The system may not be usable if the software system works as intended by the developer but it cannot support the task, or the user cannot figure out how to do it, or doesn't like the system, or the user interface is inconvenient or hard to understand and is not easy to use [8].

The main reason for not testing usability is because it is not considered as a critical feature of an application. Not only software developers in organizations but also students in schools do not include usability as one of the quality attributes of applications to be developed. In most of software organizations, usability is not integrated or considered unless it is a critical feature from the organization's perspective, although many researches indicated that usability should be integrated in the software development process.

The situation is the same for the case of Ethiopia. Based on the information and experience the researchers have, developers concentrate on the main functional requirements of the applications and do not care whether the system is interactive with the user or not. The development team spends days and weeks figuring out features and specifications without one mentioning the user; to mean that developers are not focusing on information about users.

On the other hand, not adapting usability engineering lifecycle in the software engineering lifecycle is one problem though some software development companies are not practicing specific software process model. This is because usability is wrongly identified to be a Graphical User Interface issue which can be addressed after the main part of the functionality has already been developed and lack of knowledge about the overall principles of usability engineering discipline.

The other reason for lack of developing a usable product is the issue of usability testing. Based on the information the researchers found, during the testing phase, some end-users participate in order to test the software system whether or not it fulfills the functional requirements; they are not there to verify the software ease of use, learnability, the time to complete a specific task and other usability criteria. This is because there is lack of knowledge as mentioned earlier. Hence, developers could not figure out errors and problems, which happened because of lack of usability since they do not test the software by considering the major factors of usability. In this way, it becomes very difficult to find software that fails because of usability issues in the Ethiopian context.

Since Ethiopia has many nations and nationalities with different cultures and languages, the researchers intended to work on in this emerging research area by addressing some research questions. One of the research questions is adapted from another study that was conducted in Norway [9]. This would help to practice development of usable applications whether web-based or desktop by taking into account some critical issues that make a difference on using a system.

Based on the above statements of the problem, the main objective of this research is developing a framework that integrates software usability into the software development process model. Under this, the

following specific objectives have been tried to be achieved:

- to find out which process models have been used,
- to find out which process models are being applied currently,
- to examine the degree to which usability is integrated into software process model,
- to study the problems of those process models from the view point of usability,
- to analyze the level developers and project managers understanding about usability engineering,
- to investigate the major challenges of integrating usability into software engineering lifecycle, and
- to develop a framework that guides the way of integrating usability into a particular software development process.

3. The Proposed Solution

In order for developing a framework, a survey research was conducted to analyse how software development companies perceive software process model as well as usability throughout the development process. Furthermore, a particular process model that is being applied by the majority of the software development companies is selected to come up with the framework. Thus, waterfall process model is found to be the mostly applied approach. Some of the additional components added to the traditional waterfall model in the proposed framework (see Figure 1) will be discussed as follows.

3.1 Requirement Phase

User Profile: There are some types of information that are important to put into user profile. These include (i) characteristics of the real users who are going to use the system (e.g., age, culture, language, social issues, disability issues, etc.); (ii) what they want to do with the system; (iii) how they are going to perform specific tasks using the interface; (iv) users' experience for a certain task; (v) users' motivation, and so on.

Task and context Analysis: Analysis of tasks helps for understanding the user's tasks at both a high-level and detailed level. According to [10], during task analysis, the problem to be solved and user's tasks should be documented. Otherwise it would not be enough to design a solution for undocumented problems. In the case of context analysis, three critical aspects should be explored and documented. These are usability expectation of users, their needs, and feasibility (i.e., specifications including costs and resources required to design the interface and constraints and assumptions regarding the user interface to be designed).

Usability Goals: The product should have an agreed set of qualitative and quantitative usability goals that can be used to measure how useful the product is.

After identifying the above mentioned aspects, evaluation will be made whether or not they are identified based on users' needs.

3.2 Design Phase

Interaction and Interface Design: the interaction design is the look and feel of the user interface as well as the design of the individual interaction. The interface design is about the design of the actual software that drives the interaction. These components help the developer to come up with a good user interface that can be accessible by any user.

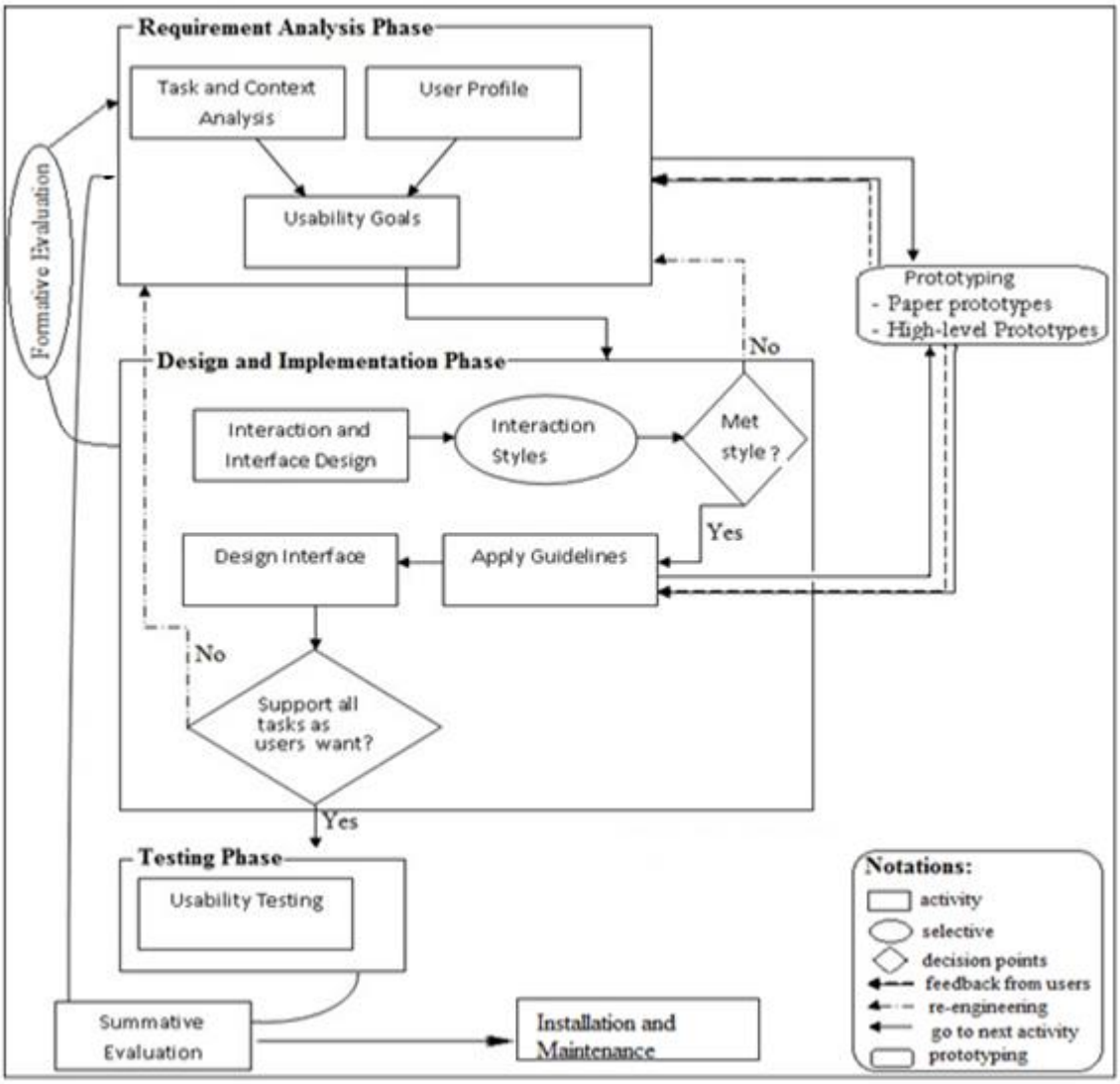


Figure 1: The Proposed Framework that integrates usability into the waterfall process model

Interaction Styles: help to identify how a user interacts with the interface from user's perspectives. Interaction styles are chosen depending on the task and user characteristics. If the interaction styles are not selected depending on these, at the end of the day, the user may not be comfortable with the design.

Guidelines: these help to improve the design of the interaction so that one should apply guidelines for developing individual interactions within a given interaction style.

3.3 Testing Phase

According to the proposed framework, usability testing must be conducted rather taking it as an optional testing type like others since the purpose of the framework is for integrating usability issues into the waterfall model and developing usable products. Hence, usability testing is the most critical part to be considered. Software development companies or individual software developers can adapt appropriate testing techniques based on the project and their characteristics.

Mainly there are three types of usability evaluation methods such as Testing, Inspection, and Inquiry. In the case of usability testing approach, representative users work on typical tasks using the system (or the prototype) and the evaluators use the results to see how the user interface supports the users to do their tasks. Testing methods include the following techniques: Coaching Method, Co-discovery Learning, Performance Measurement, Question-asking Protocol, Remote Testing, and Thinking-Aloud Protocol. The Inspection methods include Cognitive Walkthroughs, Feature Inspection, and Heuristic Evaluation. The scope of the research does not allow to briefly describe the mentioned

usability evaluation methods. For the case of formative and summative evaluation, testing techniques which have been previously mentioned can be applied.

Prototyping: this component helps to prepare a prototype in each phase of the proposed model. A prototype can enable not only the development team to better understand the requirements of the desired system –both the functional and usability - but also to clarify users' needs iteratively and early in the development process. Also prototyping at the early stage of the development process helps users to express their needs to be considered, problems they want to be solved, and their expectations from the system. It also helps to decide whether to move forward in the development process or to iterate back to the previous state. Furthermore, it is stated in [10] that the use of prototypes can potentially improve usability and user satisfaction and may reduce the resources needed for development effort. It is also identified that for supporting iterative development, software developers must be able to generate prototypes more quickly than a finished project [10].

It is mentioned in [10] that prototypes may be distinguished by the amount of detail that they include and the goal of the prototype. Thus, different scholars divide prototypes into different ways. For instance, some of them divide it into three types namely: scenario/storyboard prototypes, demonstration prototypes, and version 0 prototypes. The others divide prototypes into two common types: (i) Low-fidelity prototype – useful in the early stage of the development process and it can be presented as paper-based prototype. It helps to evaluate limited functionality and user interactions of the system, and (ii) High-fidelity prototype – this is complete and or nearly complete functionality and is interactive. It also helps the designers to gather realistic feedback from users.

The above mentioned components found in each phase of the framework are to be considered iteratively until the development team gets the right design as well as the users' needs are fulfilled as they want it to be. Furthermore, the framework forces the development team to involve users from the customer side. This will make the final product easy to use by any users. Besides, users will have high interaction with the development team throughout the development process.

4. Related Work

Many researches have been made regarding integration of usability. Most of the researchers believe that integrating usability in the software development process lifecycle is very crucial for user satisfaction, effectiveness, efficiency, and accessibility of a software system or an application.

Pyla *et al.* [2] presented a model based framework for integrating usability engineering in the software engineering lifecycle. They discussed that interactive design requires specialized UE knowledge, training, and experiences in topics such as human psychology, task analysis, visual perception, cognitive load, and so forth. They consider UE and SE as two different lifecycles and each requires their own experienced professionals. Even if the application's user interaction is tested by usability engineers and came up with some usability problems, the software development organization to which the system is tested for will not be willing to improve the problems because of budget and time constraints. They also reported that lack of coordinating the two development lifecycles leads to conflicts, gaps, misinterpreting the users' needs, lack of provision for change, lack of communicating among different development roles, and the like.

In order to overcome the mentioned problems they proposed process model as a solution that helps to integrate the two development lifecycles and also enables the software engineers and usability engineers to work together without affecting or conflicting to one another.

Nebe and Zimmermann [6] proposed an approach that identifies integration points between software engineering and usability engineering on three different levels of abstractions. Before identifying the integration points they analyzed the two usability engineering standards DIN EN ISO 13407 and the ISO/PAS 18152 in order to acquire detailed knowledge about usability engineering activities, methods, deliverables and their quality aspects. They identified that ISO/PAS 18152 defines detailed base

practices that specify the tasks for creating usable products. These base practices have been used as a foundation to derive requirements that represent the 'common activities' from the usability engineering perspective. The quantity of fulfilled requirements for each activity of the framework informs about the level of compliance of the software engineering model satisfying the base practices and therewith the usability perspective of activities. DIN EN ISO 13407 introduces a process framework for the human-centered design of interactive systems. Its aim is to support the definition and management of human-centered design activities.

Seffah *et al*. [11] reported statistical evidence that shows the importance of user involvement throughout the software development process, the cost-benefits of integrating usability and how a product can be less usable because of lack of user involvement.

Juristo *et al*. [12] mentioned that software usability should be integrated into the software development process before the design phase is started. They implicitly claimed that usability of software should be considered as one quality attribute.

Ferré *et al*. [3] proposed research regarding how to integrate usability in the software development process lifecycle by reporting some usability techniques that can be most valuable for integration purpose. They suggested from those usability techniques a particular set of techniques that can be selected based on a specific characteristic of the software development process model followed by organizations and the application being developed.

5. Conclusion and Recommendation

This paper focuses on finding ways for software developers and customers to integrate usability into the software development process in order to develop usable software. Based on theories on software development methods, user-centered design and related works, the paper has proposed a framework for integrating usability in the waterfall software development lifecycle. The waterfall method is selected because the majority of customers signed the contract under the activities of waterfall development lifecycle. Furthermore, software development team in a particular company apply waterfall combining with other methods.

The literature study documents that waterfall software development process model is not likely to take usability aspects and the user interface itself into account at the required level. It focuses on the functional aspects of software development and preparing documents about the overall structure of the project starting from requirement analysis. The waterfall method does not guarantee user participation throughout the development process as well as iterative process. So that, the framework is designed in order for making the waterfall method a short time span from the requirement phase to actually working software and enable it to respond to changes dynamically when identifying challenges and defects as the project evolves.

Comments from customers or end-users is very important and fundamental because it addresses functional challenges and errors. Moreover, it should be taken as a routine activity of the development process. This routine can give a way of how to apply valuable aspects of usability and how usability can contribute in the process for delivering it as feedback about challenges and errors in the user interaction activities. Thus, this paper has tailored the focus of enabling the software developers in developing products with integration of usability aspects in mind. That is, the software developers themselves should be able to bring a concerted action involving an interdisciplinary approach to software development including optimizing usability. Also in order to make effective use of usability, the paper examined user-centered design methods where user participation is considered absolutely central.

Numerous usability scholars consider some critical aspects of interaction design to be considered in order to come up with a usable and interactive product. So the issues to be considered in order for developing usable products are: who are the users, how do they use the software, what they want to achieve with the system, and how can one evaluate the software. By using these aspects of user-centered design, the proposed framework is more likely to provide an important knowledge about the users that

proves to be crucial to creating a software product that is in fact usable and not only just functional. After all, the software developers themselves also agreed that the end-users should be involved in the development process starting from the initial phase till the end even though they do not involve them practically.

The researcher was intending to apply the proposed framework into real point of sale application system for validating the framework. Because of some unexpected problems that happened in the company (the one who developed the POS system and delivered for its customers), the validation process could not be completed. So that the researcher intends to precede the validation process as soon as the problems are overcome.

References

[1] K. Nebe, D. Zimmermann, and V. Paelke, "Integrating Software Engineering and Usability Engineering," *Engineering*, Vol. 2, 2002.

[2] P. S. Pyla, M. A. Pérez-quiñones, J. D. Arthur, H. R. Hartson, V. Tech, and M. Hall, "Towards a Model-Based Framework for Integrating Usability and Software Engineering Life Cycles".

[3] X. Ferré, N. Juristo, U. P. D. Madrid, H. Windl, S. Ag, and L. Constantine, "usability engineering Usability Basics for Software Engineering," *IEEE Software*, February, 2001.

[4] H. Wagner, "Integrating Usability into the Software Development," University of Hamburg, 2004.

[5] W. J. Giraldo and T. Granollers, "Integrating Usability Techniques into Rational Unified Process," *Human-Computer Interaction*.

[6] K. Nebe and D. Zimmermann, "Aspects of Integrating User Centered Design into Software Engineering Processes," *Engineering*.

[7] J. Rubin and D. Chisnell, Handbook of Usability Testing_ How to Plan, Design anad Conduct Effective Tests, Wiley Publishing Inc., 2008.

[8] S. Lauesen, "User Interface Design_A Software Engineering Perspectives", Addison-Wesley, 2007.

[9] "Software Development Methods and Usability, Perspectives from a Survey in the Software Industry in Norway," *Methods*, No. 2003, pp. 1-18, 2004.

[10] L. Leventhal and J. Barnes, "Usability Engineering_Process, Products, and Examples".

[11] A. Seffah, A. Benyahmed, and M. Khalil, "Integrating Usability and Human Factors in Software Engineering Lifecycle and Organizations Software Usability Economics," 2003.

[12] N. Juristo, A. M. Moreno, Maria-Isabel, and S. Segura, "Analysing the Impact of Usability on Software Design", The Journal of Systems and Software, pp. 1506–1516, 2007.

Test Case Generation from UML Activity Diagram: The User Perspective Approach

Henok Bekele
Save the Children, Addis Ababa, Ethiopia
henzena@gmail.com

Mesfin Kifle
Department of Computer Science, Addis Ababa University, Ethiopia
kiflemestir95@gmail.com

Abstract

Prior to the release of a software to the customer, user acceptance test (UAT) will be carried out. UAT is the phase of testing which is used to determine whether a system satisfies the system requirements from the user perspective. The goal of user perspective testing is to make sure that the system completely supports the day-to-day business scenarios. But this manual UAT process has many limitations.

Unified Modeling Language (UML) models have emerged as an industrial standard for modeling software systems due to the wide acceptance of UML modeling language. More recently, they are being used in the generation of usage scenarios and test case scenarios. UML activity diagrams have got significant attention in test case generation. They have also become an ideal UML diagram to model the business process and its related concepts at higher level.

In this paper, a new approach is proposed to enable the end-user to automatically generate user level test cases from the UML Activity Diagram. The approach basically employs a series of transformations and a newly developed algorithm to get the user level test cases from the Activity Diagram. The Approach uses all path coverage criteria to generate the test cases.

A case study was conducted on point of sales (POS) system users to collect user perspective criteria to test the software system. The result of the case study has indicated that user perspective testing should focus on assessing if the system can support the day-to-day business scenarios. A prototype tool is also developed that validates the proposed approach.

Keywords: User Level Test Case; End-users; User Activity Graph; Transition List; User Activity Path Transition List

1. Introduction

Software Testing is an important phase in the software development life cycle that has to be managed very well as it plays a crucial role in assuring software quality [1]. The different phases of a software life-cycle involve different groups of stakeholders whose knowledge and understanding of software engineering differ heavily. For example, the software developer or architect has advanced knowledge about the software engineering and represents the system on his/her own way from the requirements of the end-users. On the other hand, the end-user has a deep knowledge of the domain with little or no understanding of software engineering concepts. Accordingly, at the time of testing, the system developers and end-users will have their own perspective to test the delivered system.

When end-users test the system from their point of view, they are mostly interested in testing the functionality of a system based on the acceptance or user perspective testing criteria as documented in a contract by focusing solely on the outputs generated in response to their input, ignoring the internal working mechanism of a system. In other words, they focus on checking the behavior of the system under different scenarios which they face every day in real world while doing the same task manually. This is purely a black box testing which we would like to concentrate in this research.

Sometimes, system developers involve end-users in testing the delivered system on user acceptance testing stage but this manual process has many

problems. For example, it depends on the quality of the people involved, mostly prone to error, consumes huge amount of time, and may not generate enough test cases to test the system effectively. End-users can be also influenced by the system developers in the process of producing test cases.

So in this research, we propose an approach to enable end-users to automatically generate user level test cases from the system model, particularly from the UML Activity Diagram based on user perspective criteria with minimum assistance from the system developer.

The rest of the paper is organized as follows: Section 2 discuses the background information on the basic ideas of the paper. The third section describes a case study which is carried out to gather the user perspective criteria from end-users and the fourth section discusses the proposed approach for generating test case from UML Activity Diagram focusing on the end-user perspective. The fifth section describes the implementation of a prototype. The sixth section assesses related works on the area of test case generation from UML Diagrams and finally section seven concludes the paper.

2. Background

Development of a reliable software system has to be approached in a systematic way. This requires use of appropriate tools and mechanisms to ensure a high level of quality for the system under development. Software testing is a process of verifying and validating that a software application or program meets the business and requirements that guided its design and development [3].

There are many types of software testing techniques but they can be defined in three dimensions. One dimension is showing the scale of System Under Test (SUT), which ranges from a small unit up to a whole system. The second dimension shows the different characteristics that we may want to test. The third dimension shows the kind of information we may want to use during software testing [2].The testing types under each dimension are not totally exclusive and there are also many other types of testing which are based on these fundamental testing types. In this research, we have focused on User Perspective Testing.

Software testing is usually done by the system developers but end-users have to test the system from their view before they use the delivered software in live working environment, which can be referred to as user perspective testing. The User Perspective Testing aims at verifying and validating the implemented system in the real world by the end-user according to the original needs [4]. In other words, the goal of software testing from the system developer perspective is “to make sure the software meets the specification” or “to make sure that the developed software is bug free”. Whereas the goal from the user perspective testing is “to make sure that the system completely supports the day to day business scenarios along with other known possible scenarios that may create hurdle in business operations, and to make sure that the software will not hurt the LIVE environment ” [5].

User Acceptance Test (UAT) can be considered as a typical example of User Perspective Testing. UAT is the phase of testing which is used to determine whether a system satisfies the requirements specified in the requirements analysis phase. The acceptance test design is derived from the requirements document which describes the system’s functionality, interface, performance, data, security, etc.

In Model Based Testing (MBT), software architects use UML models to represent a system. These models are used in a software specification document which is a baseline for the subsequent software development process including software testing. Normally both the developer and the end-user should test the system to make sure that the system works in accordance with the requirements. Mostly in MBT, when the system developers test the system, they use the test cases generated automatically from the UML models. On the other hand, the end-users want the test cases generated from the models to be high level so that they can understand them clearly at the time of testing. So the generated test cases can be referred to as user level test cases because they can be used by end-users to

test a software system with possible minimum support from the system developer.

UML Activity Diagram has been used more frequently to model the high level business process as it requires minimal effort and time to develop and understand. It also represents the business flow in more abstract form than other UML Diagrams [6, 7, 8].

3. Case Study

The main purpose of this case study is to gather the user perspective criteria to test a system from the actual end-users. In the case study, both qualitative and quantitative approaches have been used. We have focused on the users of a Point of Sales (POS) system and a systematic random sampling technique is used to select representative POS system user companies. An interview is also used as a data collection method and finally the primary data collected by interviewing the end-users of representative POS users has been organized and analyzed.

According to the quantitative analysis

- 85% of the respondents think that the software system should be tested by the end-users before these users start using it in live working environment.
- Only 10% of the respondents have tried to conduct an informal test to check the system works as intended by themselves.
- Almost 90% of them think that conducting a software test from the user perspective would make them very comfortable on live environment.
- 80% the respondents think that if enough training is given on the system, it is better that the system developer doesn't involve in the actual end-user test process.
- Only 5% of the end-users are fairly encouraged to test the system from their perspective.

The qualitative analysis has also showed that the majority of the respondents wanted to test the system from their point of view and they also mentioned that user perspective testing should focus on checking whether the system supports the day to day business scenarios. But mostly they can't practically perform this type of test because: there is no established mechanism to perform the test, luck of software knowledge to adopt the system developer approach to test the system, and the normal manual process would take a lot of time.

4. The Proposed Solution

In this approach, UML activity diagrams are used to generate user level test cases for testing a software system from the end-user perspective. The approach basically employs a series of transformations to get the final user level test cases from the specific Activity Diagram. Below, the basic steps of the proposed approach are listed.

Step 1: Transform the Activity Diagram into an intermediate representation.

- Export the Activity Diagram to XMI file.
- Parse the XMI file to get the required Model Information.
- Analyze the Model Information and transform it into a Transition List (TL).

Step 2: Develop a new algorithm to build a User Activity Graph (UAG).

Step 3: Extract all paths from the resulting UAG by applying all path coverage criteria. The resulting paths will be analyzed and used as test cases.

In the first step, the activity diagram will be transformed to an intermediate representation which contains only related information that is used in the process of test case generation from the user perspective. The intermediate representation is needed because even though the activity diagram presents the system flow in high level compared with other UML activity diagrams, it still contains detailed information for the purpose of generating test cases from the user perspective.

In this step, the activity diagram is firstly converted into an XML Metadata Interchange (XMI) file which is the OMG standard for exchanging metadata information via Extensible Markup Language (XML). In the next stage, the XMI file will be parsed to get the intended model information because the XMI file contains other extra details to redraw the UML Model diagrams in addition to the

model information. In the last stage, node based analysis will be carried out on the parsed model information to get the final intermediate representation which only contains a transition list that connects user related activity nodes. This is achieved by removing unnecessary nodes from the model information without affecting the original business flow of the system.

In the second step, User Activity Graph (UAG) is constructed by taking the transition lists as input from the first step. UAG is basically a cycle free directed graph. To get the intended cycle free UAG, an algorithm is developed which assumes any cyclic transitions or transition which causes cycles shouldn't occur more than once along in a given path in a UAG. In other words, the algorithm first identifies the cyclic transition lists from the global transition lists TL by analyzing the relationship of the nodes in the TL and it will then construct the UAG by adding the transitions and the respective nodes from the TL but in the process it avoids adding cyclic transitions if it has already appeared in a particular path in UAG to eliminate the formation of infinite loop in UAG. The algorithm maintains UAG Transition List (UAGTL) to internally represent the UAG that contains the depth of the UAG, unique self and parent identification per depth. UAGTL is essential for two purposes. Firstly, it helps to back track a given path in UAG to check the presence of cyclic transition at the time of UAG construction. Secondly, a user activity path can easily be extracted by following a given path in UAG to generate test cases.

In the third step, all the paths will be extracted from UAG by traversing UAG using modified depth first search or alternatively from UAGTL which was developed at the time of UAG construction. This can be achieved by checking the presence of the leave nodes and following the parent to child information all the way to the root node and repeating this process for all leaf nodes at all depths of a UAG. These generated paths are considered to have all the possible scenarios of the system as they are directly derived from the activity diagram which again contains user requirement information from the user requirement analysis phase. End-users can follow these paths to check whether the system behaves as intended at the time of testing. End-users are also provided with decision information along with a particular path so that they can identify which scenario they are following. Having this information from the tool, end-users can be able to test the system from their perspective with minimum involvement from the system developer.

5. Prototype

A prototype is also developed to demonstrate and validate the new approach. This prototype performs all the transformation steps that are mentioned in the approach and generate user level test cases. It will be used by end-users to test the system from their perspective.

The prototype is window based system which is developed using Microsoft Visual Studio 2010 and C Sharp Programming Language. Visual Studio 2010 provides a .net framework 4 that incorporates LINQ which is an efficient tool to parse the XMI file. SQL Server 2008 database is also used in the implementation to store the intermediate representation at different stages of the system process.

This prototype is designed to take input from UML modeling tools in XMI format. For this purpose, we have used Altova UModel 2011 which can model the business flow of the POS system using Activity Diagram and export the model information in XMI format. The tool will take the exported XMI file and perform two consecutive transformations, i.e., parsing the XMI file using Microsoft LINQ and Node based analysis of the model information which results in Transition Lists that contain only user related activities by removing unnecessary nodes and maintaining the original business flow. The tools will then construct UAG by applying the algorithm on the Transition List from the pervious stage and maintain the UAG transition list which serves as an internal representation of the UAG. Finally, it generates user level test cases from the UAG. Each test case contains user activity path or sequential user activities along with a set of guarding conditions or scenarios as shown in Table 1. The end-users can test the system by carrying out the sequential user related

activates and comparing it with the associated guarding conditions and make sure the system behaves as stated in the user acceptance criteria.

Table 1: Sample Test Cases for Refund Activity Diagram.

TCN	Actor	User Activity Path	Guarding Condition
1	Casher	SN→STM→SDR→SMC→LIT →SSI→FRD→RSI→PRC→FN	• Query need additional criteria • Query returns result , • Valid refund detail is entered
2	Casher	SN→STM→SDR→SMC→LIT→SSI→ FRD→RSI→FRD→RSI→PRC→FN	• Query need additional criteria • Query returns result , • Invalid refund detail is entered
3	Casher	SN→STM→SDR→SMC→LIT→SDR→LIT →SSI→FRD→RSI→PRC→FN	• Query need additional criteria • Query returns no result , • Valid refund detail is entered
4	Casher	SN→STM→SDR→SMC→LIT→SDR→LIT →SSI→FRD→RSI→FRD→RSI→PRC→FN	• Query need additional criteria • Query returns no result , • Invalid refund detail is entered
5	Casher	SN→STM→SDR→SMC→LIT→SDR→ SMC→LIT→SSI→FRD→RSI→PRC→FN	• Query need additional criteria • Query returns no result , • Query need additional criteria • Valid refund detail is entered
6	Casher	SN→STM→SDR→SMC→LIT→SDR→ SMC→LIT→SSI→FRD→RSI→FRD→RSI→PRC→FN	• Query need additional criteria • Query returns no result , • Query need additional criteria • Invalid refund detail is entered

6. Related Work

Due to the popularization of UML modeling techniques, it has emerged as the de facto standard for modeling software systems. UML can be used to describe different aspects of a system including static, dynamic, and use case views of a system. There are many researchers who propose different approaches of automatic test case generation from different UML diagrams. But UML diagrams such as Activity Diagram, State chart and Sequence diagram have got significant attention by many researchers [1, 9, 10, 11, 12]. The researchers have selected specific UML tools based on their objective and motive of their research. For example, the authors in [1, 10] used Activity Diagram to generate test cases which is used to test a system on cluster or system level whereas authors in [9] used state chart diagram to achieve component level testing. On the other hand, the authors in [11, 13] took the advantage of more than one UML diagram for the purpose of generating test cases which have adequate test coverage criteria to test the system. Below, we have tried to briefly explain some of them that are closely related to our work.

Kundu and Samanta [10] have presented an approach of generating test cases from activity diagrams using UML syntax and with use case scope. They considered a test coverage criterion, called *activity path coverage* criterion, which considers both loop testing and concurrency among activities of an activity diagram. But the approach they followed to test the system is purely from the system developer context.

Ieamsaard and Limpiyakorn [4] have presented a method and a system implemented to generate user acceptance tests from use case descriptions once the requirements analysis phase has been committed. The paper is relevant to our work as it focuses on the area of user acceptance testing but test cases are generated from the use case description and finite sate machine is used to generate a test path graph instead of UML activity diagram. It is not also clearly stated that the approach considers only user relevant processes in the course of generating test cases.

Kumar *et el*. [13] have proposed an integrated approach to generate test cases from UML sequence and activity diagrams. They first transform UML diagrams into a graph. Then, they developed an algorithm to generate test scenarios from the constructed graph. Next, the necessary information for test case generation, such as method-activity sequence, associated objects, and constraint conditions, are extracted from test scenario. In this approach, they used sequence diagram in addition to UML activity so that it also considers the message

sequence of objects in the system but this internal communication of objects is not relevant for the purpose of test case generation from the user perspective.

In general, as can be seen from the reviewed articles in the area of automatic test case generation in model based testing, the process of test case generation, whether from the developer or end-user view, can be influenced by the method of modeling the business process of the software system, the intermediate representation of the model to drive the test case generation, and finally the test coverage criteria. From the review, we can easily see that most of the researchers have focused on generating test cases from the developer perspective which as a result, the test case generation mechanism has shown a significant improvement in the past. But on the other hand, the automatic user level test case generation has been given less attention. As a result, the works that have been done in this area are almost in their infant stage.

7. Conclusion

In this paper, a new approach is presented to generate user level test cases from a UML Activity Diagram. The major significance of this approach is to give the end-user the capability to test the software independently of the software developers. In this approach, the UML Activity Diagram, which models the business flow of the system, was first translated in to an XMI file and then a transition list has been developed by parsing the XMI file and analyzing the resulting parsed model information. An algorithm has also been developed to construct the UAG which is loop free directed graph that contains only user related activities and finally user level test case has been generated from the UAG that would be used by end-users to test the system.

References

[1] Xiaoqing Bai, C. Peng Lam, and Huaizhong Li, “An Approach to Generate the Thin-threads from the UML Diagrams”, Proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC’04), 2004.

[2] Yasir Masood Malik, “Model Based Testing: An Evaluation”, Blekinge Institute of Technology, May 2010.

[3] John E. Bentley, Wachovia Bank, and Charlotte NC, “Software Testing Fundamentals Concepts, Roles, and Terminology”, SUGI 30, 2005.

[4] Chavalid Ieamsaard and Yachai Limpiyakorn, “On Integrating User Acceptance Tests Generation to Requirements Management”, International Conference on Information Communication and Management, IACSIT Press, Singapore, 2011.

[5] Abubukar Manuawar, “How to User Acceptance Test (UAT) Tips & Templete for Software Business Analyst”, Pakistan, 2010.

[6] D. Pilone and N. Pitman, “UML 2.0 in a Nutshell”, O'Reilly, June 2005.

[7] Nick Russell, Wilm P. van der Aalst, Arthur H.M. ter Hofstede, and Petia Wohed, “On the Suitability of UML 2.0 Activity Diagrams for Business Process Modeling”, In 3rd Asia Pacific Conference on Conceptual Modelling (APCCM’06), Hobart, Australia, 2006.

[8] Stuart Reid, “Systematic UML testing”, In Proceedings of the Software Quality Systems Conference 2005, Westminster, London, 2005.

[9] G. Booch, J. Rumbaugh, and I. Jacobson. “The Unified Modeling Language, Reference Manual”, Addison-Wesley, Reading, Massachusetts, 1999.

[10] Debasish Kundu and Debasis Samanta, "A Novel Approach to Generate Test Cases from UML Activity Diagrams", in Journal of Object Technology, Vol. 8, No. 3, May 2009, pp. 65-83.

[11] Santosh Kumar Swain, Durga Prasad Mohapatra, and Rajib Mall, “Test Case Generation Based on State and Activity Models”, Journal of Object Technology, 2010.

[12] P. Fröhlich and J. Link, “Automated Test Case Generation from Dynamic Models”, ECOOP 2000, Lecture Notes in Computer Science, Jan 2000, pp. 472–491.

[13] Santosh Kumar Swain and Durga Prasad Mohapatra, “Test Case Generation from Behavioral UML Models”, International Journal of Computer Applications, Vol. 6, No. 8, September 2010.

An MVC Approach to Design Content Management System

Kibreab Gizatu
Clinton Health Access Initiative, Addis Ababa, Ethiopia
kibreabg@gmail.com

Nassir Dino
HiLCoE School of Computer Science and Technology, Addis Ababa, Ethiopia
nassir@hilcoe.com.et

Abstract

In the process of designing and implementing web applications, there exist many challenges. This work aims to identify those challenges and propose solution. Hence, a number of anticipated challenges that could be encountered on the process of developing web applications have been considered to illustrate the problem area of this paper.

The web application chosen for the case study is a content management system that will help us to realize the anticipated privileges of using MVC architecture style over candidate architecture styles like SOA.

The paper concentrates mainly on the candidate architectural aspects of web applications. Architecture Centric Design Methodology (ACDM) is followed to show the architectural aspects of web application. The web framework applied in this case study is Struts 2 and the design pattern is Model View Controller.

Keywords: Web Applications; Software Architecture; Design Pattern; Content Management System; Model View Controller; Web Framework

1. Introduction

The demand of software in the form of web applications is increasingly growing. The demand of business and technology services using web applications like online booking an airline ticket, hotel reservation, becoming part of social networking site, and so on are escalating. This increasing demand on using web applications brings about a strong focus on their efficient development. Therefore software practitioners should strive to produce high quality and standardized web applications that give much value to users.

Consequently, the main purpose of this paper is identifying the difficulties and challenges in designing web applications and proposing a solution for them. One of them is the proper choice of a software architecture style that best fits the web application. The architectural design decisions that we employ to develop our web applications need to be carefully examined to produce the most competent and efficient applications. These decisions directly affect the characteristics of the resulting system like maintainability, flexibility, reuse, agility, interoperability, and scalability. This results with alternative architectures that incorporate loose coupling of the system's modules, maintainability, flexibility, usability, security, and the like.

The proposal of this paper is that the above mentioned and other more challenges can be best taken care of and solved by using the Model View Controller (MVC) architecture. This architectural pattern is best suited for the design of web applications and is implementable by the various programming languages such as ASP.NET, PHP, Java and some more. MVC has now grown into a stage where it became a framework rather than just a design pattern. In this paper the intent is to show how a well-structured web application can be developed using Java's Struts web MVC framework. This will be demonstrated by using the approach of a case study and development of a working prototype of a Content Management System (CMS) web application.

The methodology used in order to verify critical problems and to validate the solution were interviews, literature review, case study, and prototyping. In the development process, everything that the architectural framework has benefited is recorded. The remaining part of the paper is organized as follows. Section 2 states the background focusing on the motivation. Section 3 is about the

solution. Section 4 is on Case study. Section 5 is on related work and Section 6 concludes the paper.

2. Background

The importance and necessity of web applications is well recognized by the country wide software industry practitioners, researchers, the academia and by concerned government agencies. This enlightens us to allot a considerable effort to better the development procedures of web applications. There are many challenges that developers face while in the process of developing web applications. As software developers, we have been experiencing many of these challenges. The following list shows the summary of the problems to be tackled and reasons for the initiation of this research:

- Various challenges of web application development emanate from the fact that the web continuously evolves due to the need for legacy systems integration, traffic handling issues, and existence of infrastructure variation over the World Wide Web and the globe.
- Developers face difficulties trying to convert the sophisticatedly designed architecture documents by the system architects.
- There are struggles on how to simplify various challenges in the development of web applications such as validation, error-handling, decoupling the system's modules, attractive user interfaces, etc. Because of these struggles, a need to automate these processes arises.
- A huge difficulty arises when trying to make web applications compatible to a number of technologies like browsers and the flexibility to implement user interfaces in any technology assumed best fit (like ASP.NET, JSP, Velocity, XLST).
- Simplifying the integration of language packs to web applications.
- The number of difficulties faced trying to test web applications.
- Usually too much effort is spent to promote code reuse.
- The need to automate procedures like database access and session management; because of their nature to be repetitively executed.
- Incorporating a well-designed security module and dealing with bandwidth usage issues for areas with slow Internet connection (which is usually tackled with caching).
- The need to facilitate team projects that would constitute modules which are easily broken down to member developers.

The ultimate aim of this paper is showing a good mechanism to tackle the above mentioned problems that are encountered during web application development efforts.

3. The Proposed Solution

The design of web applications is by far different from designing websites or other form-based applications. A website is a set of related web pages containing content (media), including text, video, music, audio, images, etc. [1]. Form-based applications, on the other hand, are smart clients that are graphically rich, easy to deploy and update, can work when they are connected to or disconnected from the Internet, and can access resources on the local computer in a secure manner [2]. Designing web applications is different from the above mentioned application types because the web brings about so many additional features that make development very difficult. It adds a lot more variables to consider in the process of application development.

A web application is actually a web site where user input, including navigation through the site, and data entry affects the state of the business. In this kind of application, developers should deal with issues like integrating the newly developed system with legacy system databases, handling the constantly evolving nature of the web, and handling the traffic caused by a large number of users. There are, of course, other essential points to consider such as the application's security and aesthetics. In this kind of environment, fast development and quick response time for user interaction are also necessary. The architecture of the whole system also depends upon the variety of hardware devices that are involved such as web servers, client computers, browsers, application servers, the file system, and the database servers. This forces the developer to think

architecture wise and come up with a structure that results in an efficient end product.

The general objective of this paper is suggesting an architectural solution that solves the problems mentioned and illustrating this by implementing a case study web application.

4. Experimentation/Prototype

The implementation of the CMS prototype for in this paper is taken care of using the following technologies.

- Eclipse Java EE IDE for Web Developers,
- Struts 2 MVC web framework,
- Hibernate Object Relational Mapping (ORM) framework, and
- MySQL relational database.

The project structure of the CMS prototype application is composed of several folders to represent the various components of an MVC application. Figure s shows how the project is structured inside Eclipse IDE. The project is entitled CMSAdmin. The main folders inside the project are located inside the Java Resources: src folder and inside WebContent folder. The package folders inside the Java Resources: src folder are named com.cmsAdmin.dao, com.cmsAdmin.domain, and com.cmsAdmin.web. The DAO folder contains data access objects that contain code that queries and writes to the database. The Domain folder contains the entity classes that represent the database tables of the project. The Web folder contains the action classes that serve as the Model in the MVC triad. The rest of the files in the source folder are xml files that serve for the various configurations of the Struts 2 application implementation.

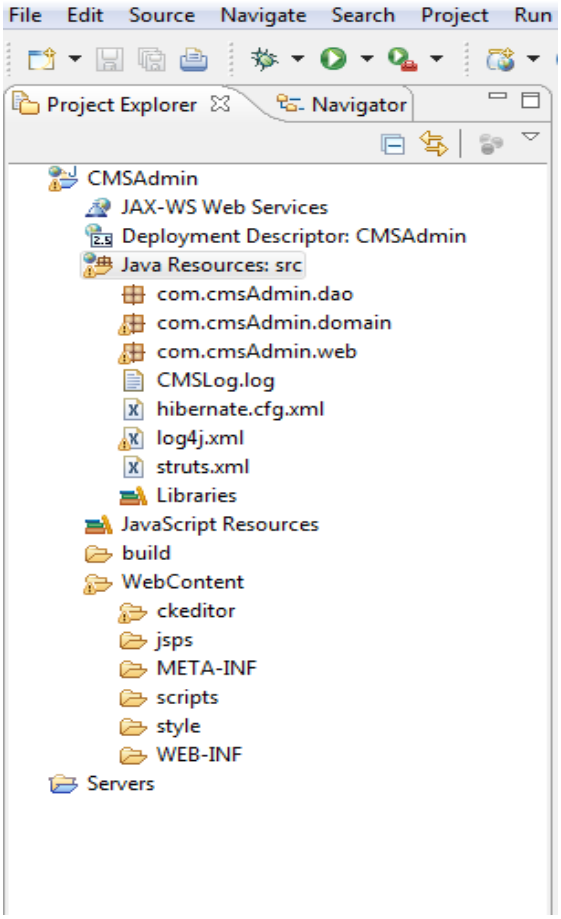


Figure 1: Project Structure inside Eclipse IDE

The other important folder is the WebContent folder. This folder contains most of the resources that relate to the View section of the MVC triads. It contains the JSP pages in the jsps folder, various scripting resources in the scripts folder, cascading style sheets (CSS) in the style folder, and web application configuration xml (web.xml) in the WEB-INF folder.

The very reason why the project is structured the way it is and the fact that the technologies mentioned are utilized is to facilitate the provision of solution to the problem statement.

5. Related Work

Over the course of research for this paper, it has been noticed that not much resource is available that concentrates on the architectural aspects of web applications. It was very difficult to find literature that emphasized upon web frameworks, web application architectures, and current application design best practices. The main sources of literature were books, articles on the Internet, blogs, and

forums. But this doesn't mean that there was absolutely nothing that was available and relating to the soul focus of this project.

There have been some works done to show the importance of careful architecting upon web applications to produce quality products as in [3, [4]. These works clearly show the current trend of web architectures and try to propose architectural solutions that best fit the demands. They show that web applications are more accessible and have lower maintenance and deployment costs than traditional systems. They classify web applications as brochure, service oriented, data intensive, and information systems web applications. These works also point out that web applications are under constant evolution because of the demands of the web and, therefore, require an architecture that best cops with these requirements.

Also Ziemer [5] in an essay for architecture of web applications pointed out that elaborating a sound architecture for a software system is highly crucial. It was also pointed out that the architecture chosen for designing a certain web application should take into consideration the fact that development of web applications is highly evolutional and ever changing.

As explained in [6], the need to incorporate architectural frameworks in the development of applications is becoming mandatory. The need is even doubled when it comes to web applications. The architectural frameworks available today solve much of the overhead involved through the development of web applications. They now come up with their own libraries for data access, session management, templating and promote code reuse. The article also points out that most application frameworks found these days are based on the Model-View-Controller (MVC) architectural pattern. The frameworks use this pattern for the main purpose of decoupling data model with business rules from the user interface. It continues to explain the use of the frameworks by pointing out that they take care of much overhead related to database access mapping, URL mapping, and even in some the provision of tools to create and provide web services.

According to Hennebrueder [7], a complete list of available frameworks found these days would exceed over 200. Over the top of the list are Apache Shale, JavaServer Faces, JBoss Seam, Stripes, Struts 2, Spring, Hibernate, Ajax libraries, JQuery, etc. This clearly shows there is a desperate need to standardize the development of web applications and change our development culture. We have to be able to choose the best architectural framework out there that fits into the kind of web application that we are developing and apply it.

As pointed out in [8], the web application platform has become very powerful and allows for developing all classes of applications that run over the web. Current web applications demand the functional feature set and look and feel of traditional desktop applications. The writer continues to suggest that the core workflow of these applications remained the same. Also a web framework helps developers to be relieved from the mundane concerns of the domain by the provision of reusable architectural solution to the core web application workflows.

6. Conclusion

The specific objectives of this paper were to:

- clearly show the advantages of integrating architectural frameworks in the development of web applications,
- point out the extra benefits of using the MVC framework to design the kind of web applications demanded these days by comparing it with other alternative architectures like Service Oriented-Architecture,
- investigate the requirements of the CMS web application that is going to be used for the case study, and
- test and evaluate the developed system to show that the whole procedure of the development has in fact proved to be a solution to the problems discussed earlier.

This paper attempted to show what challenges exist in the effort of developing web applications. The main purpose of the paper was to uncover what the challenges were and providing a sound method or an approach that solves them.

After the prototype application has been implemented, the findings suggest that through the process of developing the case study application (CMS), most of the challenges identified have been successfully solved. That means a sound approach to how those challenges can be eased up and tackled is identified by applying the web framework Struts 2 which implements the MVC architectural pattern. The rest of the challenges on the other hand could not be treated through the making of the application using Struts 2 framework and MVC architectural pattern. This was because the solution for these problems required a more sophisticated series of procedures that can't be visible enough just by implementing a single case web application. They required a little more study coverage of other areas such as the web's nature, browsers, security issues and the like. These remaining challenges are:

- Legacy system integration
- Handling infrastructure variation over the web
- Traffic handling over the web
- Browser compatibilities
- Security module integration
- Handling low bandwidth issues (caching)

It is the assumption and best hope of this paper that these problems be given some attention and get researched upon in future works. The fact of the matter is these challenges relating to web application design and implementation still exist.

References

[1] Wikipedia, the free encyclopedia, "Website", en.wikipedia.org/wiki/Website, Last-accessed-Mar 12, 2012.

[2] MSDN Library, "Windows Forms and Smart Client Applications", http://msdn.microsoft.com/en-us/library/8bxxy49h.aspx, Last-accessed: Mar 12, 2012.

[3] Roy T. F. and Richard N. T., "Principled Design of the Modern Web Architecture", ICSE 2000 Limerick, Ireland.

[4] Guntram G. and Martin G., "An Evolution-Oriented Architecture for Web Applications", In Second Nordic Workshop on Software Architecture, NOSA, 1999.

[5] Sven Ziemer, "An Architecture for Web Applications Essay in DIF 8914 Distributed Information Systems", November 28, 2002.

[6] Multiple (wiki), "Web application framework", Docforge, Retrieved 2010-01-19, Accessed on October 27, 2011.

[7] Sebastian Hennebrueder, "Choosing Web-Frameworks", http://www.laliluna.de/articles/posts/the-web-framework-evaluation.html, Accessed on October 27, 2011.

[8] Donald B. & Chad M.D., "Struts 2 In Action", 2007.

IPv6 Migration Framework – Case of Institutions in Ethiopia

Kidist Mekonnen
Zemen Bank, Addis Ababa, Ethiopia
kidistmt@yahoo.com

Taye Abdulkadir
HiLCoE School of Computer Science and Technology, Addis Ababa, Ethiopia
tabdulkadir@gmail.com

Abstract

In the 1970s, the IPv4 protocol was considered as a base for the modern Internet. At that time no one anticipated limited address space of IPv4 would have a huge impact on the development of Internet. In the 90s, the problem of limited IPv4 address space was noticed and this became a main reason to work on IPv6. The new users and applications of the Internet have the potential to rapidly increase the utilization of IPv4 address space and deplete it. Since the future growth and global connectivity of the Internet is highly dependent on the availability of IP addresses, it is affected by the late IPv6 deployment.

In this paper, IPv6's additional features and the well known IPv6 transition mechanisms are presented. It has also assessed the level of awareness of IPv6 among IT professionals of different IT stakeholders in the country like Ministry of Communication and Information Technology (MCIT), banks, and educational institutions through conducting a survey. Readiness for IPv6 migration of ethio telecom (the only ISP in the country) has also been assessed. This work has covered the issue of addressing the governing policy of the Ethiopian government in regard to the migration to IPv6.

Finally, the study has proposed IPv6 migration framework/strategy for organizations in the context of Ethiopia.

Keywords: IPv6; IPv6 Awareness; IPv6 Deployment; IPv6 Transition Mechanisms

1. Introduction

Internet Protocol (IP) addresses are required for a computer or a network device to send data to another over the Internet. In the 90s, the problem of available IPv4 address space was noticed and this became the main reason to work on IPv6 [1, 2]. With this in mind, the Internet Engineering Task Force (IETF), a community that works on the evolution of Internet technologies, created IPv6 to accommodate the growing number of users and devices accessing the Internet by predicting the need for more IP address [2]. IPv6 has a very large address space and consists of 128 bits as compared to 32 bits in IPv4. Therefore, IPv6 enables more people and devices to access the Internet by providing IP addresses in higher number.

On February 03, 2011, the Internet Assigned Numbers Authority (IANA) allocated one of the final five blocks, which is about 16 million, to five Regional Internet Registries (RIRs) around the world. Today, there is no remaining IPv4 address space to be distributed in the Asia Pacific region and Australia, and it is expected to run out around the end of 2012 in Europe; in America in 2013 and in Latin America in 2014. In the case of Africa, unless IPv4 addresses use increases in a big rate, the continent will be safe for at least another two or three years.

Similar to the rest of the world, IPv6 adoption is also a challenge for Africa. To support a sustainable Internet growth on the continent, AfriNIC has the opportunity to define its own policies and plans that will help IPv6 adoption. In point of fact, AfriNIC has been conducting trainings for the past five years to attract government representatives and the private sector to take the leading role in IPv6 deployment. Unfortunately, most African governments have not yet taken any action towards IPv6 migration [3].

Coming to Ethiopia, though Ethiopia was one of the countries in the world which has acquired telephone system just a few years after its invention, the telecommunication service penetration rate and development of the required infrastructures has relatively remained far behind as compared to other countries [4]. But the number of Internet users is still

growing very fast. The new users and usage of the Internet have the potential to rapidly increase the utilization of IPv4 address space and deplete it.

Any country or organization that (for whatever reason) does not migrate to IPv6 may not be in danger for the time being mainly due to the parallel co-existence of IPv6 and IPv4. However, in the long run, it will be difficult for them to connect to networks which support IPv6, or for people in other countries to connect to networks which support IPv4. At some point, deployment of IPv6 with proper readiness and migration strategy is a requirement, not an option for Ethiopia, too. To do IPv6 migration in systematic and careful way, it is necessary to have a guiding strategy towards IPv6 deployment in the context of Ethiopia and assess the awareness of IPv6 among IT professionals of selected IPv6 deployment stakeholders and infrastructure readiness of ISP as well as selected IPv6 deployment stakeholder organizations.

2. Transition from IPv4 to IPv6

The Internet requires overall architecture progress to accommodate new technologies.

IPv6 features (extended address space, auto configuration, simple header structure, extension headers, IPSec, mobility, quality of service, route aggregation, and efficient transmission) facilitate to meet the requirements of the new applications over the Internet and open up a broad field of technological development. Despite the fact that IPv6 has a lot of advantages over IPv4, it is not possible to throw away the existing IPv4 network and adopt IPv6 immediately.

Therefore, organizations should have to identify potential risk areas and their mitigation, and be prepared to migrate to IPv6 with an acceptable level of risk.

Security risk before and during maintaining interoperability, economic risk, and implementation risk are the different risks that were identified in this study which are associated with the IPv6 deployment.

Due to the incompatible nature of the two protocols (IPv4 and IPv6), the transition from IPv4 to IPv6 cannot be expected to be smooth. To facilitate IPv6 transition, the IETF has set up a work group, NGtrans (Next Generation Transition), which specifies mechanisms for supporting interoperability between IPv4 and IPv6. The various IPv6 transition strategies can be broadly divided into three categories, dual stack, tunneling, and translation mechanisms [5].

3. Empirical Data and Analysis

A successful migration of IPv6 requires the coordinated effort of all stakeholders throughout the country. Anyone who is associated with the Internet, either as a user or as a service provider, has to be involved in this process. Hence, evaluating the IPv6 awareness as well as readiness of the IT environment in the relevant stakeholders (government, ISPs, software/hardware vendors and end users) is an essential first step in defining the transition framework. The awareness of IPv6 readiness is a crucial point to guide IPv6 deployment in Ethiopia. Questionnaires as well as various publications and interviews were used to compile the qualitative data collected and analyzed in this paper.

a. Government

Governments, as sources of driving policies, have key roles to play in raising awareness and promoting the deployment of IPv6 throughout a country. Lack of awareness with regard to issues of IPv4 depletion and IPv6 adoption could be risky on national competitiveness and continuity of government services. In the case of Ethiopia, this government organ is the Ministry of Communication and Information Technology (MCIT). MCIT has a mission of developing and utilizing the communication and information technology broadly throughout the country, and strives to optimize the role of the sector in political, economical, and social development thereby improving the living standard of the society [6].

Accordingly in this work, the status of MCIT towards IPv6 migration has been assessed. As a result of interviews of relevant employees in the Ministry, IPv6 had no impacts on software development process and because of this understanding, there was no work done by MCIT. It is in the process of starting development of IP

address and DNS Administration and Utilization guideline. It is explained that IPv6 is one of the issues which would be covered in that assignment.

b. Internet Service Provider (ISP)

The telecommunication industry in Ethiopia is currently fully owned by the government. The government owned operator, formerly known as Ethiopian Telecommunication Corporation (ETC) and now ethio telecom, provides PSTN, mobile telephone, Internet, and data communication services. The rate of penetration of telecom services of the country is among the lowest in Africa and even among Sub-Saharan African countries.

ethio telecom, the only Internet Service provider in the country, should play a key role in the national adoption of IPv6. For ethio telecom, IPv6 deployment is even more important to set itself as market leader than for other industries to follow.

In the rest of the world, as IPv6 deployment progresses, those ISPs who have not invested in adopting IPv6 in their networks may find attracting new customers difficult, and may begin to lose existing customers who wish to proceed with their own IPv6 deployment. Despite the fact that there is no other ISP in the country that competes with ethio telecom (and hence no market to lose), IPv6 deployment helps to boost ethio telecom's services using the new IPv6 features.

As a result of interviews of relevant employees, ethio telecom has implemented dual stack on its backbone network (core network) which enables it to support IPv6 services. This clearly shows that the company is on the way towards IPv6 in order to prepare for the future. It was also explained that they assumed that IPv6 will be available as a standard service for all the customers in two years. On the other hand, none of the interviewees have any information as to when the public IPv4 address in their network as well as for customers will run out. However, increasing awareness of the technical benefits of IPv6 among the technical personnel should be one big task for ethio telecom. According to most of the interviewees' understanding, there is no plan to deploy IPv6 fully and also IPv6's new features are not that much of a drive for IPv6 deployment. By exercising proper use of NAT, the interviewees believe that ethio telecom can manage the IPv4 addressing problem without thinking of IPv6 deployment. It was also mentioned that IPv4 address depletion should not be an issue for ethio telecom in any case. As far as the knowledge of the all the interviewees is concerned, there is no nationwide initiative towards IPv6 migration.

c. End Users

For end user organizations, the question of whether their ISP can provide IPv6 services is one of the most significant factors in deciding whether or not to commit resources to their own IPv6 initiatives. The transition from IPv4 to IPv6 will become a must for organizations, especially to those that currently provide online services based on IP addresses. The intention of this survey is to demonstrate how IT professionals of the stakeholders think about IPv6 and how far they have gone in the process of implementing the new protocol. Using a self administrated questionnaire, the information has portrayed the respondents' knowledge of IPv6 technology and status of and plans for IPv6 deployment as well. Knowledge about IPv6, infrastructure readiness of IPv6, training about IPv6, and IPv6 transition plan are the main aspects covered by the questionnaire.

In general, based on the survey result, we have noticed that the knowledge and awareness about IPv6 (what it is and how it works) as well as IPv6 ready equipment are minimal.

As it is noted from the interviewee's response, almost all participant organizations' network hardware and software are IPv6 ready. Most of these IPv6 ready hardware and software are enabled by default.

This study identified that this situation creates a critical security issue for organizations which have not prepared and implemented security policy for IPv6 as the same security degree as of IPv4.

Migration from IPv4 to IPv6 is also a concern in the software development environment. Because IPv6 does not modify the Application Programming Interface (API) between an application and the IP stack, applications which are IP address dependent like banking applications should need modification to work with IPv6 network environment. In this case,

vendors and developers of those applications should be trained well to modify their applications to support IPv6.

4. Proposed IPv6 Institutional Migration Framework

The proposed framework has been designed under a national context taking into account the following factors to be done by the responsible organs. The Government (MCIT) should take the initiative in designing nationwide IPv6 awareness campaign and set policies and procedures for IPv6 migration and controlling compliance of IPv6 ready hardware and software. ethio telecom (the ISP) should work to help facilitate the address space acquisition process.

IPv6 adoption is not easy and it is not a single step that can be achieved in a short time but one which requires a great amount of thorough planning and preparation to develop and adjust to IPv6 business. Organizations should consider the proposed framework shown in Figure 1 as a guide when deciding on IPv6 migration.

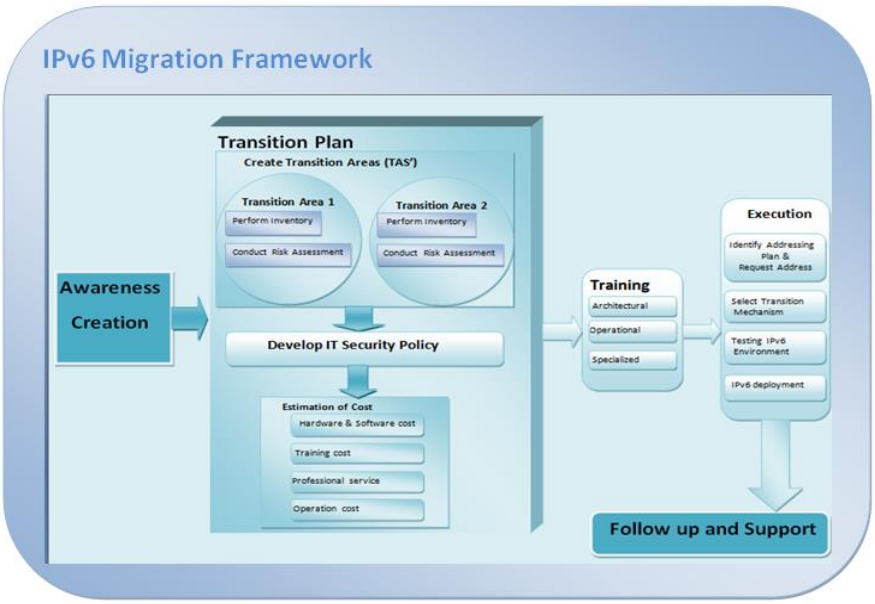


Figure 1: IPv6 Migration framework for Organizations

5. Conclusions, Recommendations, and Future work

Transitioning from IPv4 to IPv6 is a sensitive issue worldwide, and devising an IPv6 migration plan is important for organizations to maintain customer productivity. Organizations that delay their IPv6 migration plan put themselves at risk, the risk of being unable to communicate with customers. Throughout the study, IPv6 features, well known IPv6 transition mechanisms, and review of IPv6 transition framework were presented. IPv6 awareness and readiness is also assessed among IT professionals of selected IPv6 stakeholder organizations. The survey result indicated that we still have unresolved issues on IP address management and still struggling to properly deploy IPv4 Networks. It is also noted that there is continuing extensive usage of NAT in the country. However, one can take this as an opportunity to start the IPv6 migration and not to invest much on the current IPv4 infrastructure.

In Addition, it is very important that each country has enough IPv4 addresses to manage the transition to IPv6 while running dual-stack configurations. Since Africa has enough IPv4 addresses for at least the coming two years, now is the ideal time for countries like Ethiopia to plan their migration to IPv6. Based on survey result, IPv6 migration strategy/framework was developed for institutions in the context of Ethiopia.

This study serves as an initial step in disseminating awareness on IPv6 and triggers some initiatives from policy makers and service providers. It also provides a framework which helps on guiding the migration for organizations towards IPv6.

For a successful IPv6 migration, the government should play a role in coordinating and supporting the dissemination of IPv6 standards, protocols, and conformance. Otherwise competitiveness and IT leadership will suffer.

With respect to developing IPv6 expertise, the other most important recommendation of this study is for academia. Academia should design a short course of its own that covers all aspects of IPv6 and setting up pure IPv6 backbones for testing and experimentation. Software and hardware vendors should also put greater efforts into marketing IPv6 and conduct IPv6-related training for their customers and users.

Since it is critical that organizations have information as well as support from their ISP about IPv6 services, ethio telecom should develop a plan to support organizations to test IPv6 migration strategy. Future studies also need to be carried out in developing frameworks on IPv6 migration for different industries and assessment of the different IPv6 transition mechanisms for a specific case.

References

[1] J. Batallaa, A. Binczewskib, and W. Burakowskic, "Why is IPv6 Deployment Important for the Internet Evolution?", pp. 5-15.

[2] J. Palet, "The Choice: IPv4 Exhaustion or Transition to IPv6," April, 2009.

[3] Sadissa, "IPv4 IP Addresses exhaustion: What's next for Internet in Africa? http://sadissa.wordpress.com/2011/02/04/ ipv4-ip-addresses-exhaustion-whats-next-for-internet-in-africa/, accessed on 01-Jul-2011.

[4] C. Telecommunication, D. Officer, and E. T. Agency, "The Regulatory and Institutional Dimension by Ethiopian Telecommunication Agency", March, pp. 17-19, 2010.

[5] Saisree Subramanian, "IPv6 Transition Strategies", White Paper, November 2003.

[6] The Federal Democratic Republic of Ethiopia Ministry of Communication and Information Ethiopia's Vision, pp. 0-28, 2012.

Quantitative Usability Measurement for Commercial Off-the-Shelf Products: the case of Core-Banking System

Lemlem Berhanu
Commercial Bank of Ethiopia, Addis Ababa, Ethiopia
lemruth@yahoo.com

Rahel Bekele
School of Information Science, Addis Ababa University, Ethiopia
rahtesf@yahoo.com

Abstract

This paper assesses how usability is perceived and considered as evaluation criterion at the time of purchasing COTS products. Additionally, how far a COTS product is usable for its users and the extent of user satisfaction is also evaluated. Further literature study, survey, and its findings are discussed. Finally, COTS Usability Questionnaire is developed to quantitatively measure usability for COTS products.

Keywords: Usability; Quantitative Usability Measurement; Core-Banking System; COTS Product; COTS Usability Questionnaire

1. Introduction

Banks and financial service organizations have long recognized the advantages of deploying technology to improve value, speed and, flexibility of their product offering to customers [1]. Software developed in-house or Commercial Off-the-Shelf (COTS) products are used to support their products and services.

Among COTS products, Core-Banking System (CBS) is used by commercial banks. CBS is a vital element that helps a bank to differentiate itself and allows offering many new technology driven channels to customers such as Automated Tellers Machines and Internet Banking [1]. In order to systematically evaluate, rank, and select a COTS product that best meets the bank requirements as presented by Ncube and Dean [2], different criteria like cost, usability, functionality, size, portability, supplier capability, etc. are used. The focus of this paper is on usability as evaluation criterion of COTS products at the time of purchasing.

2. Statement of the Problem

Usability is increasingly recognized as an important factor for interactive software systems like CBS [3]. Moreover, Ovaska [4] states that lack of usability evaluation of software packages thoroughly leads to acquisition of software with poor usability. Poor usability increases the time needed to get desired results, increases errors, and causes misunderstanding, even misjudged decisions [4].

The National Bank of Ethiopia is on the way of implementing National Payment System (NPS) which supports straight through processing between commercial banks which requires implementation or upgrading of CBS [5]. Furthermore, the cost of purchasing a CBS is a huge investment not only for commercial banks but even for the country as well [6].

Since usability is a quality that many products possess, but many lack due to historical, cultural, organizational, monetary, and other reasons [7], focusing on usability of COTS products is found to be important. Therefore, the aim of this paper is to quantitatively measure usability of COTS products.

3. Survey Results

A survey was undertaken in six selected commercial banks using questionnaire in order to identify the current practice of usability in procurement of COTS products. Moreover, usability testing was also conducted on selected CBS to assess the extent of user satisfaction and how easy a COTS product is for actual users. According to the study analysis

- Usability is defined as user friendliness, ease of use and quality of software by majority of the respondents, whereas others

misconceived it as an attribute of the interface.

- Usability is not considered as one of the evaluation criteria like functional requirements, non-functional requirements, vendor profile, product demonstration, reference check, and negotiations.
- Usability evaluation methods that are used are not appropriate to measure usability of COTS products.
- Usability testing proves the existence of poor efficiency, "difficult" tasks and poor level of satisfaction for COTS products. These confirm that the COTS product is not fairly usable for its real users.
- Challenges related with usability at time of COTS product procurement:
  - A requirement that was not well defined in the Request for Proposal (RFP).
  - Difficult to observe usability based on vendor's response from an RFP document. Due to this, product demonstration and reference check are used to cross check usability of the candidate CBS.
  - A problem of getting usability experts.

4. The Proposed Solution

A COTS Usability Questionnaire is developed to quantitatively measure usability of COTS products. The questionnaire contains forty items with a five-point Likert scale with anchors ranging from "Strongly Disagree" to "Strongly Agree" as shown in Table 1. The questionnaire items include factors such as learnability, flexibility, memorability, efficiency, effectiveness, and satisfaction.

Table 1: COTS Usability Questionnaire

No.	*Question*	*Strongly Disagree*	*Disagree*	*Undecided*	*Agree*	*Strongly Agree*
1.	My previous experiences of using other systems help me learn the system very easily.	O	O	O	O	O
2.	It is tiring to click multiple buttons to confirm an action.	O	O	O	O	O
3.	The system helped me to perform my assigned tasks accurately and timely.	O	O	O	O	O
4.	The system is pleasant to use.	O	O	O	O	O
5.	The error messages clearly tell me how to fix problems.	O	O	O	O	O
6.	Messages which appear on the screen are sometimes confusing.	O	O	O	O	O
7.	Using the system, tasks can be performed in a straight forward manner.	O	O	O	O	O
8.	Working with the system is satisfying.	O	O	O	O	O
9.	It takes too long to learn commands/menus to operate the system.	O	O	O	O	O
10.	After a period of not using the system, I can easily use it.	O	O	O	O	O
11.	Instructions and prompts are helpful and clear.	O	O	O	O	O
12.	I feel in command of the system when I use it.	O	O	O	O	O
13.	Learning to operate this system initially is full of problems.	O	O	O	O	O
14.	The system provides me options to recover from mistakes quickly and easily.	O	O	O	O	O
15.	I sometimes do not know what to do next with the system.	O	O	O	O	O
16.	I would not like to use this system.	O	O	O	O	O
17.	The commands relevant to a set of tasks are easy to learn.	O	O	O	O	O

No.	*Question*	*Strongly Disagree*	*Disagree*	*Undecided*	*Agree*	*Strongly Agree*
18.	The system makes me recognize how to perform a task instead of remembering.	O	O	O	O	O
19.	The help information given by the system is not very helpful.	O	O	O	O	O
20.	Using the system is frustrating.	O	O	O	O	O
21.	I have to look for assistance when I use this system.	O	O	O	O	O
22.	The flow of the system reflects how the user thinks of the work flow.	O	O	O	O	O
23.	The system gives me control over my activities.	O	O	O	O	O
24.	I would recommend this system in other organizations.	O	O	O	O	O
25.	I feel that a lot of training will be needed for beginners to make use of the system.	O	O	O	O	O
26.	I can easily customize/adapt the system.	O	O	O	O	O
27.	The reports generated are not often reliable and accurate.	O	O	O	O	O
28.	I found the system useful in my work.	O	O	O	O	O
29.	I have easily learnt the steps needed to accomplish what I want to do.	O	O	O	O	O
30.	I can easily and successfully find the help options.	O	O	O	O	O
31.	It is relatively easy to move from one part of a task to another.	O	O	O	O	O
32.	I am willing to teach others on how the system works.	O	O	O	O	O
33.	It is easy to make the system do exactly what you want.	O	O	O	O	O
34.	I can always make safe exits from the system whenever I want too.	O	O	O	O	O
35.	The response time of the system is the cause of my frustration.	O	O	O	O	O
36.	The system does everything I would expect it to do	O	O	O	O	O
37.	The logical grouping and ordering of menu options helped me learn the system.	O	O	O	O	O
38.	I feel safer if I use only a few familiar commands or operations.	O	O	O	O	O
39.	Instructions for commands or choice are clear.	O	O	O	O	O
40.	I like the order of presentation of interfaces.	O	O	O	O	O

Learnability means the system should be easy to learn so that users can start getting some work done. Flexibility and memorability indicate the system is easy to remember and a user is able to use the system after some period of not using it, without additional learning. Moreover, users are able to customize aspects of the system. Efficiency and effectiveness mean once the user has learned the system, a high level of productivity is achieved by accurately completing specified tasks with expected time. Satisfaction shows how users are subjectively satisfied by using the system.

The most important and fundamental characteristics of any measurement procedure, reliability and validity, were assessed by a survey collected from 200 COTS product users. The analysis showed Cronbach's Alpha of about 0.9 that is good internal consistency. All factors regarding learnability, efficiency and effectiveness as well as satisfaction have a reliability result which shows good internal consistency. But factors grouped as flexibility and memorability resulted in a reliability index less than the minimum suggested value (i.e., 0.7). It also has good construct validity.

In order to quantitatively measure usability of COTS products, first representing the responses given to each item score contribution with a range from 0 (i.e., "Strongly Disagree") to 4 ("Strongly

Agree") is required. For negatively worded items, the contribution ranges vice versa that is 4 as "Strongly Disagree" to 0 as "Strongly Agree". Evaluated usability is a product of total average value, which is the sum of all respondents' total score divided by total number of participants by item value (i.e., 0.625).

As pointed by Bangor *et al*. [8], using a Standard Letter Grade Scale, products that scored in the 90s were exceptional, products that scored in the 80s were good, and products that scored in the 70s were acceptable. Anything below a 70 had usability issues that are causes for concern. Based on this, interpreting quantified usability of COTS products makes it easy to understand the implication by many people and it facilitates communication of results.

5. Relate Work

Though usability is becoming a more and more important software criterion, available usability measurement methods are either difficult to apply, or overly dependent upon evaluators' expertise.

Lin *et al*. [9] identified eight human factor considerations (i.e., compatibility, consistency, flexibility, learnability, minimal action, minimal memory load, perceptual limitation, and user guidance) which are relevant to software usability along with the three stages of human information (Perceptual, Cognitive and Action) processing theory which formed the framework from which Purdue Usability Testing Questionnaire (PUTQ) was derived. PUTQ is intended mainly to measure users' satisfaction toward a Graphical User Interface (GUI). Moreover, PUTQ applicability for COTS products is limited since it doesn't take into account some usability factors such as effectiveness and efficiency.

Furthermore, Lewis [10] used a rank-based system when assessing competing products. The technique works with data collected during scenario-based usability studies where participants are asked to perform realistic tasks with products. This approach creates a rank score comprised of both users' objective performance measures and subjective assessment. Application of the method allows a single composite rank-average to be assigned to a product to represent its relative usability by allowing easy comparison of products. But the resulting metric only represents a relative comparison between like-products with similar tasks. It does not result in an absolute measure of usability that can be compared across products of different task-sets.

The rank-based method is meant for software development and did not consider COTS products from the procurement perspective. Furthermore, it doesn't give absolute result for usability of the software under testing, which makes it difficult to adopt the method for COTS products usability testing. This is due to the fact that COTS products with poor usability might get the first rank when compared to other candidate COTS products. This misleads the evaluation committee members to make a wrong decision to purchase the software.

Several usability measurement models have been proposed to measure and report usability of a general software product. Sauro and Kindlund [11] proposed a quantitative model which simplifies usability dimensions into a single, standardized and Summated Usability Metric (SUM). In SUM, task completion, error counts, task times, and satisfaction scores metrics are used in order to represent effectiveness, efficiency, and satisfaction usability dimensions. A summative evaluation is used to quantitatively assess the "before and after" impact of design changes which provides an idea of how usable a task or product is without having to reference historical data.

On the other hand, Seffah *et al*. [3] come up with Quality in Use Integrated Measurement (QUIM), a consolidated hierarchical model for usability measurement which decomposes usability into factors, criteria, and metrics. The main application for QUIM is to provide a consistent framework and repository for usability factors, criteria, and metrics for educational and research purposes. The goal of developing QUIM is to reduce software development risks in a less expensive approach.

Most of the tools that exist for assessing software usability are usually proprietary and may only be available for a fee. In addition, these tools were intended to be used at software development lifecycle rather than for procurers of software at final software product level.

6. Conclusion and Future Work

In this paper, COTS Usability Questionnaire is presented to measure usability of COTS products. It is useful for commercial banks and other COTS purchasing organizations to make appropriate evaluation in terms of COTS product's usability at the time of purchase. The research provides information on how usability is perceived and gives insight for decision makers, how far the current CBS is usable or not for its intended users. Furthermore, it pinpoints the banking community to consider usability as one of the requirements and criteria at the time of purchasing COTS products.

Further task is required to amend the questionnaire so that it can yield better reliability index for the usability factors flexibility and memorability. Moreover, the proposed solution quantifies usability of COTS products by giving equal weight to usability factors. It is recommended to research about weighting the usability factors and automation of the COTS Usability Questionnaire.

Even though the majority of commonly used questionnaires were initially developed in the English language, as pointed in [12], different types of questionnaires have been widely translated into other languages. Future work in this study is recommended to translate the COTS Usability Questionnaire into local languages like Amharic.

References

[1] Infrasoft Technologies, "End-to-End Core Banking Solution for Competitive Advantage," White Paper, 2009, http:// cache-www.intel.com/cd/00/00/19/30/ 193047_193047.pdf, last accessed on December 10, 2011.

[2] C. Ncube and J. Dean, "The Limitations of Current Decision-Making Techniques in the Procurement of COTS Software Components," *in Proceedings of the 1st International Conference*, 2002, pp. 176-178.

[3] A. Seffah, M. Donyaee, R. B. Kline, and H. K. Padda, "Usability Measurement: A Roadmap for a Consolidated Model," *Software Quality Journal*, Vol. 14, No. 2, 2006, pp. 159–178.

[4] S. Ovaska, "Usability as a Goal for the Design of Computer Systems," *Scandinavian Journal of Information Systems*, Vol. 3, 1991, pp. 47-62.

[5] J. Woltjer, L. Elmasry, and V. Babin, "Modernization of the National Payment Systems in Ethiopia, Part 3, Vision and Strategic Framework," Version 6, 2009.

[6] IBM, "Commercial Bank of Ethiopia Signs Strategic Deal with IBM to Modernize Core Banking Systems and Support Launch of 500 New Branches," 2011, http://www-03.ibm.com/press/us/en/pressrelease/34913.wss, last accessed on December 01, 2011.

[7] J. Rubin and D. Chisnell, *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*, 2nd ed., Wiley Publishing, 2008.

[8] A. Bangor, P. Kortum, and J. Miller, "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale," *Journal of Usability Studies*, Vol. 4, Issue 3, 2009, pp. 114-123.

[9] H. X. Lin., Y.Y. Choong, and G. Salvendy, "A Proposed Index of Usability: A Method for Comparing the Relative Usability of Different Software Systems," *Behaviour and Information Technology*, Vol.16, 1997, pp. 267-277.

[10] J. R. Lewis, "A Rank-Based Method for the Usability Comparison of Competing Products," in *Proceedings of the Human Factors Society 35th Annual Meeting*, San Francisco, Human Factors Society, 1991.

[11] J. Sauro and E. Kindlund, "A Method to Standardize Usability Metrics into a Single Score", in *Proceedings of the Conference in Human Factors in Computing Systems (CHI 2005)*, Portland, 2005.

[12] C. W. Sun, "Questionnaire Translation and Psychometric Properties Evaluation," SEGi University College, Vol. 2, No. 2, 2009, pp. 45-51.

A Methodology for Implementation of Function Dimension in Zachman Framework

Meron Bekele
KFT (Kifiya Financial Technologies Plc), Addis Ababa, Ethiopia
merrybekele@gmail.com

Tibebe Beshah
School of Information Science, Addis Ababa University, Ethiopia
tibebe.beshah@gmail.com

Abstract

This paper aims to propose an implementation methodology for Zachman Framework for Function dimension from software development process perspective. The paper presents the limitation of Zachman Framework with regards to its implementation methodology. Zachman Framework is an enterprise architecture with a logical construct to define and control the interfaces and integration of all components of a system. The Zachman framework enables systematic capture of system specific information from the various dimensions with respect to each view.

The scope of this paper is limited to the "Function" dimension or column. An implementation methodology for Zachman framework has been proposed for each cell that lies within the scope of the study.

A case study was conducted on a company by selecting a specific project. From the selected project, a process has been identified so as to model it based on the proposed methodology. After modeling each cell, an interview has been conducted to validate the proposed methodology. The participants in the interview were stakeholders at each dimension.

The advantage of the proposed methodology is that it is helpful for software development process because it will enable defining the process at each view and also helps as a validating mechanism for the software developers to check whether they have captured each process as per the different stakeholder requirements.

Keywords: Enterprise Architecture; Zachman Framework; Dimension, Implementation Methodology; Function Column

1. Introduction

Many large size enterprises operate in a very dynamic and complex environment and these large enterprises want to ensure that they have the best enterprise architecture framework (EAF) that has the capability of managing the information technology (IT) structure and also the knowledgebase of the environment. Sometimes, the term "Enterprise Architecture (EA)" refers to the group of people responsible for modeling and then documenting the architecture. The term also denotes the process of doing this work [1]. According to the enterprise architecture community, Enterprise Architecture is a framework or "blueprint" for how organizations achieve the current and future business objectives. It examines the key business, information, application, and technology strategies and their impact on business functions. Each of these strategies is a separate architectural discipline and enterprise architecture is the glue that binds each of these disciplines into an integrated framework [1].

2. Motivation Scenario

The prime motive for this research is the absence of an implementation methodology for architectural frameworks of Zachman enterprise architecture. The methodology will be of great help in software development as the end product is software. Moreover, the need for architectural guideline to make maximum use of the potential of information technology is worth mentioning as an additional motivational factor for this specific research.

3. Proposed Methodology

The aim of the study is to propose a methodology on how to implement Zachman Framework from system development perspective. The proposed

methodology has been developed based on the different literature reviews and analyses made. Thus, the research combines literature review and case study. As the scope of the study is only on process dimension, the proposed methodology is focused on each cell that corresponds to the process dimension. The cells that have been proposed with a methodology are Scope-Process, Business-Process, System Model-Process, and System-Process. The proposed methodology incorporates the modeling technique to be used, the modeling tool which helps model based on the technique selected, and the steps that have to be followed in going from one cell to the other. One of the rules in Zachman Framework discusses about the level of detail when we go down from one cell to the next cell below it.

When we go from one cell to another cell, a transformation is taking place through the application of a different set of constraints. Therefore, what is making a lower row cell different from a higher row cell in the same column is not the level of detail. The cells in different rows of the same column are different because they are actually models of different things [2]. In any of the cell, we could have a high level of detail model or a medium level of detail model. Level of detail does not necessarily increase from cell to cell down a column. It is fortuitous that only the process models are hierarchical because hierarchical decompositions result in unidentifiable redundancies in the detail which makes normalization difficult, if not impossible [2]. This rule has a direct relationship with our study because it is stating that level of detail is hierarchical in process dimension. Based on this rule, in order to model any cell except the first one, we have to get an input from the cell above it and it has to be noted that any change made at a cell has an impact for the cells below it.

Table 1: Proposed Modeling techniques

Views-Perspective	*Focus*	*Modeling Technique*
Scope-Process (Planner view)	Business Function	List, Use case
Business-Process (Owner view)	Process Model	Business Process Modeling Notation (BPMN), Activity Diagram
System Model-Process (Designer view)	Application Architecture	Component Diagram , Context diagram
System-Process (Builder view)	System Design	Class Diagram, Subsystem

As show in Table 1, there are two modeling techniques that have been proposed for each of the cells and also a modeling tool called Altova UMode. This modeling tool is the starting point for successful software development. It visual design application models in UML and it generates Java, C#, or Visual Basic .NET code and project documentation. Or, reverse engineer existing programs into UML 2 diagrams, then fine tune the designs and complete the round trip by regenerating code. Altova UModel is the UML tool that makes visual software design practical for any project. It is the simple, cost-effective way to draw on UML [6].

4. Related Work

A method for consistent modeling of Zachman Framework cell was proposed in [3]. The authors presented three advantages of Zachman Framework (ZF), one is the fact that it uses a well defined perspective, the second one is using of comprehensive abstract, and the third is normality and extensive usage in practice. Among the challenges sated was the absence of an integrated language to model cells in the framework.

ZF doesn't recommend any tool or model for a particular cell. There is no technique to address the cells in the Zachman Framework matrix which itself resembles an obstacle. To resolve this obstacle, the authors have proposed a Model Driven Architecture (MDA) in order to model all cells in ZF. According to the paper, applying UML seems to be the best solution up to now but the argument that was presented in the study was that UML is not mature enough to support all aspects of enterprise architecture. According to the study, MDA separates certain key of a system into Computational Independent Model (CIM), Platform Independent Model (PIM), and platform specific model (PSM). Figure 1 [3] shows the proposed solution for all cells.

Zachman Framework	Data	Function	Network	People	Time	Motivation
Planner	Out of problem space					
Owner	Business profile	Business Use Case	Organization Unit stereotype	Business profile/use case	Timing diagram	BSBR BMM
Designer	CWM	BPDM	EDOC	BPDM	Timing & State diagrams	PRR
Builder	CWM	UML 2.x	Deployment diagram	Interaction Overview diag.	Timing diagram	OCL
Contractor	CWM	CORBA, EJB, .Net, WSM	CORBA, EJB, .Net, WSM	Interaction Overview diag.	Timing diagram	OCL

Figure 1: Consistence model for Zachman Framework cells

The proposed approach has been implemented on a case study and it was evaluated by a measurement criterion that was set by the authors. The result presented states that the feedback they get were promising compared to previously proposed methods, indicating that it is well suited to diminish the modeling problems one is facing using ZF. Thompson [4] conducted a research on scaling Zachman Framework for small scale non-Enterprise applications. The software development methodology brought forth in Zachman's System Architecture Framework can be used to design a wide range of information systems. While the Zachman Framework is very robust and typically used for developing large scale Enterprise Applications, the study demonstrates that the framework can easily be scaled to fit a small scale non-Enterprise Application.

Fazil *et al.* [5] discussed about how digital library brought a high impact phenomenon as it contributes a lot in terms of new ways in dealing with knowledge. They stated that libraries built after 1995 have a different look and feel that defines "intellectual information centers" where people and technology interact. According to the paper, most scholars believed that the way knowledge has been created and disseminated fundamentally were altered by a series of innovation begins by Michael S. Hart, with his Gutternberg's invention as the first producer of free electronic books in 1971. It was presented that apart from being the oldest digital library developed, Gutternberg also works as a catalyst in rapid development of multiple kinds of digital libraries nowadays. The study proposed a framework to determine the content of a semantic theses digital library based on Zachman Framework data dimension. The study used the six artifacts in Zachman Framework which are Data, Process, Place, People, Time, and Motivation and applied it to represent six layers of semantic digital library architecture. Twenty-four online theses collections of public and private institutions of higher learning in Malaysia were selected for data collection and further evaluation.

A case study approach and a collection of master's and doctoral dissertations in Perpustakaan Tun Seri Lanang, Universiti Kebangsaan, Malaysia was chosen as the case sample. The collection scope is limited to theses from the Faculty of Technology and Information Science from year 2000 to 2009. A variety of data collection techniques were applied including a pilot study by distributing questionnaire among 20 postgraduate students of University Kebangsaan, Malaysia, interviews with librarians, focus group interviews, and on site observations to observe the specific environment of the theses digital library. As presented in the paper, feasibility study to identify the approaches, methods, and techniques in implementing semantic digital library as well as designing the models and architectures of semantic digital library were represented in a usable semantic theses digital library prototype. Then user testing, evaluation, and refinement of digital library prototype were executed until a complete literature review of semantic theses digital library projects compiled and a formal framework to determine content of a semantic theses digital library based on Zachman Framework accomplished.

5. Conclusion

This paper proposed a methodology on how to implement Zachman Framework from Function column dimension. It is reported on most of the researches that Zachman Framework is a well known and accepted enterprise framework but it lacks implementation methodology. The prime motivation for this research was the lack of implementation despite its acceptance in the industry. In order to come up with a methodology, literature review has been conducted. Based on the different literature reviews, we have proposed the methodology and it was validated by a case study. Based on the case study, Use case was selected for Planner-Process cell, BPMN was chosen for Owner-Process cell, Component diagram has been selected by majority for the Designer-Process cell, and finally Class diagram has been well accepted by programmers as a modeling technique for the builder-Process cell. The proposed methodology is valuable, especially for software developers because at every cell, each stakeholder is presented with a modeling technique that s/he is able to understand easily. On the other hand, it will help the process to be captured well and finally when developers move to development, they will have a requirement that is very clear and also that has been validated by each stakeholder. The research has advantage in puling enterprise architecture concept to the software development practice by addressing each stakeholder with the respective cell. It also adds a value by testing Zachman Framework for software development process. One of the challenges in software development is capturing process in the manner that stakeholders planned it. Hence, the proposed methodology will help in capturing all the process at each step.

References

[1] Castelo Branco, "A Method to Define an Enterprise Architecture using the Zachman Framework", in ACM Symposium on Applied Computing, Nicosia, Cyprus, March, 2004.

[2] John A. Zachman, "The Zachman Framework for Enterprise Architecture: Primer for Enterprise Engineering and Manufacturing", Zachman International, 2003.

[3] S. Shervin Ostadzadeh, Fereidoon Shams Aliee, and S. Arash Ostadzadeh, "A Method for Consistent Modeling of Zachman Framework Cells", 2007.

[4] Carla L. Thompson, "Scaling the Zachman Framework", Regis University, School for Professional Studies, 2006.

[5] Qurratu AiniAdibah Ahmad Fazil, Zuraidah Abdullah, Shahrul Azman Mohd, "Applying Zachman Framework to Determine the Content of Semantic Theses Digital Library", Faculty of Information Science and Technology, Universiti Kebangsaan, Malaysia, 2010.

[6] Altova, "UML Tool for Software Modeling and Application Development", http://www.altova.com/umodel.html, March 2012.

Automated Heuristic Evaluation for Usability of Multiple Web Interface

Mihirtab Zewdu
Kolfe Keranyo Sub City Land Information Centre, City Government of Addis Ababa, Ethiopia
mihirtz@gmail.com

Dagmawi Lemma
Department of Computer Science, Addis Ababa University, Ethiopia
dagmawil@yahoo.com

Abstract

Heuristic evaluation is the most common usability evaluation method of web interface, that involves usability specialists' judgment on agreed upon usability principles (heuristics). The current specification method is limited on independent expert judgments. Though, there are works to evaluate web interfaces from the usability perspective, these works underlying developer heuristics and lacks consistency in using the agreed represented standard heuristics.

This research addressed the above problems by representing agreed heuristic specification of principles via ergonomics guideline and automating evaluation that would made possible by AHEMW algorithm. As a result, this automation is one step towards a procedural syntax based heuristic evaluation of multiple web interfaces that brings consistent evaluation by maintaining users' preferences.

Keywords: heuristic Evaluation; Usability Evaluation; Web Interface

1. Introduction

The unaddressed issues/problems that need to be solved over this study are; How to represent the heuristics for automation to usability evaluation of web pages? Could it be possible to find evaluation techniques that consistently perform automated evaluation across multiple web pages? Usability in the ISO standard for software qualities (ISO 1991b) is defined as "a set of attributes of software which bear on the effort needed for use and on the individual assessment of such use" [2]. Also Eason defines usability as "the degree to which users are able to use the system with the skills, knowledge, stereotypes and experience they can bring to bear".

Heuristic evaluation is defined as a usability evaluation technique in which the most appropriate solution is chosen using rules and principles [3, 4].

Recently used evaluation techniques are used to obtain information from limited usability attributes, which plan accessibility guidelines to allow interface evaluation only through pre-defined and limited guideline evaluation definition method. The interface evaluation syntax reduces dependency on implementation specifics and makes more reusable [5]. The ultimate goal is to design and implement automated heuristic evaluation techniques for consistent evaluation of publicly accessible hypertext structures.

The study is worthy because of user preference based web software evaluation method developed, simple and natural expression of web users creating satisfaction and creating consistency in evaluation of multiple pages. However, the study is limited on context awareness and prediction and types of guidelines (i.e., guidelines have to be representable as per this work).

2. Related Work

2.1 Automation of Heuristic Evaluation

The literature has showed different usability evaluation methods such as analytical modeling and information simulation model. Most of these usability evaluation tools were recording action methods a user makes while exercising in a given web interface. This produces voluminous log report that makes it difficult to map the recorded log to usability actions, which has a need for further analysis on the huge amount of data that reduces evaluators' efficiency.

Literature [7, 9] shows that most quantitative methods of evaluating web sites focus on statistical analysis of usage patterns. The analysis is mainly based on server logs, traffic-based analysis, and time-

based analysis that provide data the evaluator needs to interpret. However, it has been delimited on the lack of agreement over guidelines and suggests as there is no one path to good design. The studies open a way toward the expression of guideline definition in platform independent definition by creating empirically justified and reproducible interface design recommendations.

A study in [7] tried to demonstrate knowledge-based web automated evaluation tool with reconfigurable guidelines optimization (KWASMI). It covers guidelines based evaluation for a single web page with Guideline Definition Language (GDL) representation [13]. The guideline definition is done using GDL. It lacks to have a heuristic basis for the criteria's and platform independence in the definition.

2.2 Design of Automated Evaluation

It is important to have the ability to facilitate the evaluation of web pages by creating well-developed (platform independent) guideline among institutions that wish to implement them. Guideline based evaluation of a content specific web based evaluation from the contents of HTML source sent to the web browsers has decisive impact for usability judgment.

2.3 Guideline Representation

Design advisor [10, 11] explained by Faraday contains an ergonomic algorithm that predicts the visual path between elements on the web page depending on their type, size, and color. It has been used to assist the visual analysis of web pages. Following eye tracking analysis results of design advisor provides log report results of evaluated web pages. Which evaluators biased towards aesthetically pleasing interfaces and structurally feasible designs regardless of efficiency and effectiveness.

Web Static Analyzing Tool (Web SAT) processes HTML tag grouping in an individual page and does not suggest possible mitigation and improvement strategies [12]. This tool does not support other usability guidelines such as IEEE Standard except the HTML composition of web pages.

2.4 Multiple Web page Evaluation

The problem with multiple pages is unaddressed to automated evaluation of metadata description, due to the nature of web navigation and crawling web page is nondeterministic and technologically it was unreachable.

Even if single page has made great improvements to the multiple views of user preferences it has limitation of handling logged and graphed information [8].

3. The Proposed Solution

Lack of usability experts and increased demand of experts as a possible solution consists of retaining the knowledge and experience of these experts and expressing it in syntax form to be reviewed and applied. So, the manual heuristic makes an overhead to the evaluation because of the need for organization, limited domain expertise, and need to have automation.

An evaluation is needed to perform evaluation of the product selection process and produce a usable user interface at stage n+1 by making conformity of ergonomic and functional requirements to the adherence of heuristics.

In the manual heuristic evaluation, results of the assessment and experts' judgment is recorded as written reports from each independent evaluator or by having the evaluators verbalized comments from an observation as they go through the interfaces. The currently often used approach to heuristic interface evaluation is manual evaluation that provides independent interface usability judgment, which has bias more to the aesthetics of the web interfaces. In general, the representation helps to obtain direct access to relevant evaluation points in the usability evaluation, as a result proceed through the different steps of the evaluation.

3.1 Guideline Representation

A knowledge representation is needed for guidelines of a web that could be used as if they explicitly state the attributes of the usability guidelines of web content. The representation can be new or modified guideline criteria enabling evaluators to express the usability body of

knowledge done in a separate definition to come across the heuristic definition.

The evaluator need to configure according to the web problem of the system evaluation presented. The overall guideline structure grouped into five problem domain design evaluation decision structure are, high level design, conceptual content, content objects, navigation & search, and content presentation process.

Identifying the guidance on structuring WWW user interface problem attributes: The syntax followed to represent heuristics is:

[HTML Tag/Evaluated content substring][Existence/Pattern] [Needed/Not needed/Optional] [Messages to the process] [Scope of criteria]

The syntax defines standards to establish to a grammatical practices, identify changes to improve web services incorporating opinions and needs of users [6]. Once defined in representation, heuristic knowledge can be evaluated by the distinguishable parts of web content.

Table 1: Syntax definition for the AHEMW representation

Syntax element	*Use interpretation*	*Syntactic nature that entails to the evaluators`*
[HTML Tag/ Evaluated content substring] alternative name: Evaluated value	Specific notation to which the evaluation acts. It perform layout or functional evaluation	The subject of the evaluation,
[Existence or Pattern]	The item the functions i.e. possible values: either existence or pattern,	Act as a verb of required evaluated value,
[Needed or Not needed or optional] for Existence or [pattern {tag_substring}] for pattern	Is a value where the specified web rendering content is either 'needed' or 'not needed' is specified. 'sub-sting' for the pattern needed.	Act as a value object, configured to the evaluator context.
[Success and failure messages to the problem]	Show message for success and problem message for corrective notification.	
[Scope of criteria in the evaluation]	Scoped the evaluation guideline is either for only home page or for any of the child pages.	

The categorization into five problem domain of user interface design in Table 1 of heuristics helps to follow through

G: Place title information consistently, where G is set of criterion of 'C1 'and 'C2'

By applying the grammatical syntax the principle criterions defined to the evaluation pattern.

C1: Title substring (Tag) should exist and C2: Title should be presented consistently

Definition using syntax for C1: [title], [existence], [needed], [Title Exists], [Title Doesn't exist], [any]

Definition using syntax for C2: [title], [pattern {title_substring*}], [needed], [Title of a page is consistent], [Title of a page is inconsistent], [any]

As defined in the above the represented guideline first, title is checked if it exists in the title tag, then the 'title_substring' pattern of checked in the <title> tag, whether the string pattern is followed or not. The criteria used is set for a heuristic evaluator guideline specification.

* The title substring to be evaluated is stated. For example: Yahoo has a pattern Yahoo mail, Yahoo news.. following "Yahoo- service name" pattern.

3.2 Automation of Heuristic Evaluation

When a heuristic evaluation of the web was conducted based on usability standard, the objective was to assess the web pages compliance with usability standards and design guidelines. The usability issues were classified according to the usability problem domain design evaluation decision structure.

Design of Automated Evaluation

Several groups are attempting to create computer interpretable guideline representations that can provide user specific decisions [15]. It is important to have the ability to facilitate the evaluation of web pages by creating well-developed (platform independent) guideline among institutions that wish to implement them. Guideline based evaluation of a content specific web based evaluation from the contents HTML source sent to the web browsers has decisive impact for usability judgment.

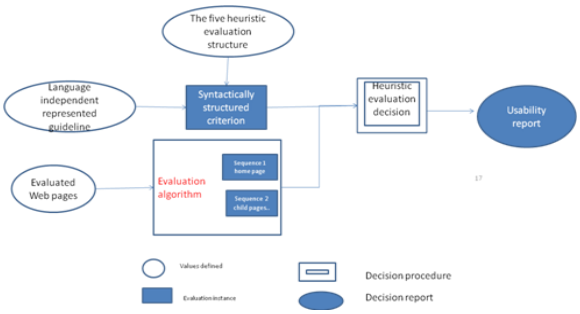


Figure 1: Generic MWIE model of heuristic evaluation.

As illustrated in Figure 1, the MWIE model usability guideline representation is specified in language independent specification, where the evaluators come across the different views.

The knowledge of notation is promoted to user interface evaluation patterns by providing a specified evaluation interaction design notation and documenting recurring solutions [14]. After that, evaluation report is developed that expose multiple pages usability violation consistently.

3.3 Algorithm of Automated Evaluation

The representation in a separate evaluation configuration focuses to work and execute the heuristic definition that makes separation of guideline easily modifiable and web programming language independent web interface content

Web pages consist of a set of display tags, links and additional data associated child links. All web servers mainly dispatch HTML tags to browsers. Home page is the main page through which users typically enter a website and whose URL is typically published or linked as the main web address. The Uniform Resource Locator (URL) substring is retrieved from the home page and child pages content. The URL is queued for web pages content that uses URL of home page stored in temporary variable stack for recursive analysis.

Algorithms pick the right criteria for the respective situation, in a sequential approach for usability problems [16].

<-- = takes method URL= address in <a href “”> tag

Test result = result of syntactic string comparison DOM to HTML

```
Start
    Httprequest(URL of homepage) // provided by the evaluator in the AHEMW graphic
      interface
    http request (Test stub [i++]) <-- Array[page[i]]
    Evaluate Homepage <-- TestStub(page[1]) //Homepage
    Httprequest Success (Home page) {
        [HTML Tag/ Evaluated content substring][Existence/Pattern] [Needed/Not
          needed/Optional] [Messages to the process] [Scope of criteria]
        Test result[11] <-- string compare(Homepage)
        [HTML Tag/ Evaluated content substring][Existence/Pattern] [Needed/Not
          needed/Optional] [Messages to the process] [Scope of criteria]
        Test result[12] <-- string compare(Homepage)
```

```
        Usability evaluation Result <--Test result(page [i])
        Array(i =0 i< numberTestStub i++)
        {
            http request (Test stub [i++]) <-- Array[page[i]]
            Evaluate page [i]<-- TestStub (page[i])
            For (j=0 guideline[j] <-- guideline +1) Httprequest Failure (URL) {
                Do {
                    http request (Test stub [i, URL) attempt <-- maxAttempts
                    Attempts Success : JUMP X
                    Failure : Report as "broken link"
                }while (Attempt<3)
                    TestStub[i+1] <-- For each (<a href = "descendent child URL address" >)
                        : JUMP to Point X
            }
        }
        Usability evaluation Result <--Test result(page [i])
End
```

Algorithm 1: AHEMW Algorithms

The two main activities done in the algorithm are: firstly, the evaluating algorithm queue home page by defining set of TestStub make set of evaluation with a needed syntax parameters.

Start

Httprequest(URL of homepage) // provided by the evaluator in the AHEMW graphic interface

Evaluate Homepage <-- TestStub[1] (Homepage)

The usability algorithms are deterministic as they build the contents inspection of a web interface by applying usability rules defined in the representation. The action is made by syntactic string comparison to guideline represented and retrieved home page DOM of HTML to a guideline defined pattern and existence, which is stripped from the structural HTML markup. The algorithm generates report for each test case as long as there are evaluations covered as to the required loop.

4. Prototype

The AHEMW algorithm is simulated to show how the representation and heuristics contribute for the automation of usability evaluation. In addition, the representation concepts have been enabled to show in a natural human understandable, pre-defined and web content based usability corrective recommendation to be made. The prototype is developed in JavaScript JavaScript with ExtJS library was used for reading (retrieval) file, and retrieving DOM structure of web.

The AHEMW is divided into three sub systems (Figure 2): graphical interface, guideline representation, and evaluation decision subsystem, because it helps to embrace new approaches and perspectives in order to cope with the complexity of modern evaluation.

Graphical interface subsystem: The evaluation needs URL information for evaluation service request. The evaluation graphical interface is provided by URL.

Guideline representation subsystem: Make ISO 9241-151 heuristics of human-system interaction part 151 guidance on WWW standard into easy specification in a separate text file so that any evaluator can be behind an interface without impacting the rest of the subsystem.

Evaluation decision subsystem: Provides analysis to the landing and any of child pages. The attributes result is presented to the corresponding identified problem.

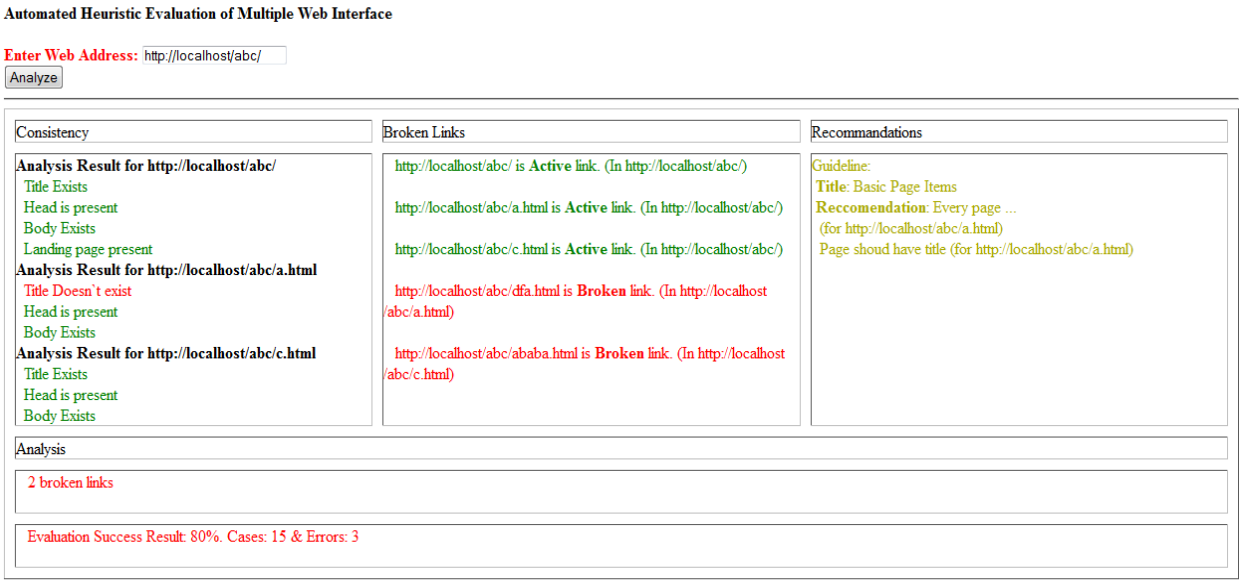


Figure 2: Prototype for running usability evaluation

5. Conclusion

The analysis report is made by an empirical validation and taking a case of in HiLCoE School of Computer Science and Technology site. The representation facilitates communication amongst evaluators practicing across the globe. Also it helps evaluation usability specialists to understand and share best practices.

The automation with an algorithm enables evaluators better understand reactions to their web evaluation and help them defend against guideline critics facilitating different solution assumptions for the problem of web pages. The AHEMW enabled to meet upcoming evaluation challenges by creating platform for evaluation language independence and studying representation that address evaluation of multiple pages consistently.

References

[1] A. Holzinger, "Usability Engineering Methods for Software Developers", Communications of the ACM, Vol. 48, No. 1, pp. 71–74, Jan. 2005.

[2] G. Gaffney, "What is Usability Testing?," in Information & Design, 1999.

[3] B. A. Myers, R. Wolf, K. Potosnak, F. A. South, and C. Graham, "Heuristics in Real User Interfaces", INTERCHI'93 Panel, pp. 1–4, 1993.

[4] A. Seffah, M. Donyaee, R. B. Kline, and H. K. Padda, "Usability Measurement and Metrics: A Consolidated Model", Software Quality Journal, Vol. 14, No. 2, pp. 159–178, Jun. 2006.

[5] B. Venners, "Design Principles from Design Patterns: A Conversation with Erich Gamma, Part III," Scribd, http://www.scribd.com/doc/79668962/Design-Principles-From-Design-Patterns.

[6] M. Y. Ivory, "An Empirical Foundation for Automated Web Interface Evaluation", University of California at Berkeley, 2001.

[7] R. Grandgagna and U. De Louvain, "KWARESMI – Knowledge-based Web Automated Evaluation Tool with Reconfigurable Guidelines Optimization," Proceeding of 2nd Int. Conf. on Universal Access in Human-Computer Interaction UAHCT', No. 4, pp. 1504 – 1508, 2003.

[8] M. Y. Ivory and M. A. Hearst, "The State of the Art in Automated Usability Evaluation of User Interfaces", ACM Computing Surveys, Vol. 33, No. 4, pp. 1–26, 2000.

[9] M. Y. Ivory and U. C. Berkeley, "Empirically Validated Web Page Design Metrics," CHI 2001, Vol. 3, No. 1, pp. 53–60, 2001.

[10] K.-P. Robert W. Proctor, "Web Guidelines vs GUI Guidelines", In Handbook of Human Factors in Web Design, pp. 691–692, 2005.

[11] C. Mariage, "State of the art of Web Usability Guidelines," … of Human Factors in Web …, pp. 1–25, 2005.

[12] J. Scholtz, S. Laskowski, and L. Downey, "Developing Usability Tools and Techniques for Designing and Testing Web Sites," in … on Human Factors & the Web, Vol. I, pp. 1–10, 1993.

[13] F. B. Tan and L. L. Tung, "Exploring Website Evaluation Criteria using the Repertory Grid Technique: A Web Designers' Perspective", in SIGHCI 2003 Proceedings, pp. 65–69, 2003.

[14] D. Nuno and J. Nunes, "Object Modeling for User-Centered Development and User Interface Design: The Wisdom Approach," No. April, 2001.

[15] P. Chawan and S. Ninoriya, "Towards the Quality Measurement of Web Content Management System," International Journal of Advanced Engineering & …, No. 148, pp. 148–151, 2011.

[16] E. K. Burke, B. L. Maccarthy, S. Petrovic, and R. Qu, "Knowledge Discovery in a Hyper-Heuristic for Course Timetabling Using Case-Based Reasoning," 2002.

A Web Based School Networking System with ICEfaces MVC in the Ethiopian Context

Pineal Terefe
pinny_thf@yahoo.com

Fikreyohannes Lemma
PhD Candidate, Addis Ababa University, Ethiopia
fikreyohannes@yahoo.com

Abstract

This work came into existence to address the necessity that educational institutions found in Ethiopia have communication and resource sharing needs. Besides, it also addresses one more issue, which is the fact that the public does not have a centralized place where they can get information about the whereabouts of the educational institutions and the services that they provide. After clearly identifying and prioritizing these problems, the paper illustrates the method, architectural design techniques/technologies, and other approaches required to developing web-based software. Ultimately, it brings solutions to the aforementioned issues with the help of a software prototype. In doing so, the software's overall architectural design will be illustrated which will be incorporated by the end product, in this case the prototype. The methodology that was incorporated for the design is a combination of architecture centric and user-centered, which means that the needs of users of the system will be considered from the start of the designing process but most of the design will focus on the structural components of the system, their interaction, and significant design decisions made about the organization of the system. MVC (Mode-View-Control) architecture is chosen for the overall software design and implementation which provides modifiability, flexibility, and a clear separation of design concerns among other things. The framework used for the implementation of the prototype is ICEfaces. The detail on how this framework realizes MVC is clearly stated in this paper. Apart from the design of the software, the deployment architecture is also presented which is 3-tier architecture. Finally, using snapshots from the developed prototype, each of the system's features will be demonstrated and a conclusion will be drawn based on the results of the prototype.

Keywords: Model View Controller; School Networking System; Web Application; ICEFACES MVC Implementation; Software Architecture

1. Introduction

At this day and age, Internet is playing huge role in bringing the world closer together with adequate information. The numerous possibilities that are offered by its existence paved a way for software developers to use it and make something that helps different aspects of our life. People can communicate with each other by sending e-mails, via chat or using other ways like online discussion forums. People can also search for electronic materials, ideas and/or share electronic resources by uploading and downloading them. Information is a key to success and one way of availing it to the people is via applications that run on the Internet. The Internet by itself is nothing but a huge network and what makes it useful is the applications and software that run on it which are designed to solve problems that a particular group of people are facing. These problems can be different based on their nature, severity, applicability, and many other reasons.

This being said, educational institutions of our country starting from elementary schools to universities, all use the Internet one way or the other. Specifically speaking, they need to communicate with each other to share ideas. They need to share electronic resources like audio, video, e-books, tutorials, research papers, and the like. On the other hand, the public also needs to know about the whereabouts and other detailed information of these educational institutions. Parents, being part of the existing educational system, also need to get connected to their children's school to get frequent information about the status of their children and other essential pieces of information about the school

community. All these issues need a solution. There can be different possible ways of tackling this; for instance, by investigating for any existing work done on similar issues and customizing it to achieve all of these abilities. The other possible option is designing a new software whose goal is bringing a solution to the issues mentioned here. Both of these ways need thorough assessment of what is out there and what different methods and technologies there are to be used to attain the desired outcome. This paper summarizes the overall processes that were followed to address all of those needs that our educational institutions have regarding communication, resource sharing, providing information to the public, and creating a way for parents of students to interact with the school of their child.

2. Background

Educational institutions are one segment of the vast varieties of service providers in our country. By educational institution, it is meant to say any of the higher or lower level schools including universities, colleges, high schools, and elementary schools. These institutions share some common features/functionalities that can be generalized without affecting the way how things are actually being done at a particular institution. Obviously, each of these institutions needs to communicate with each other and/or let their members communicate with one another; they need to share electronic materials amongst each other, they need the public to know about their existence and/or their services, they need to manage student records, and they also need to stay connected with the public for mutual benefit (accepting donations or for other social reasons). Taking these common needs into account, a more generalized solution can be proposed architecturally which makes the application design more effective and operationally feasible. The major issue here is to create a structure where each of these institutions exists in a system where all the aforementioned issues are addressed.

This being said, currently there are a number of technologies and design patterns that software developers can incorporate in their software designs. Specially, for web based software, there are new and advanced web technologies that are emerging these days. On the other hand, architectural patterns are the other issues that a software developer should identify and integrate into the designing process. Choosing the right architectural pattern demands knowing the nature of the software to be developed and picking the right pattern among many possible architectural solutions. This can be done by stating the pros and cons of each possible architectural style and choosing the one that best fits the design considering the architecturally significant requirements set for the project and the design methodology chosen for the overall project. Just like the designing process, the deployment is also a crucial aspect of software development. It involves identifying the software and hardware components and how they are going to be mapped to different physical processes.

Among the many development technologies, ICEfaces is the one that was selected to be incorporated for this work. ICEfaces is a JSF extension that adds rich UI and AJAX features to JSF applications. It also supports AJAX which eliminates the full-page-refresh problem that most of the existing frameworks have. AJAX plus rich UI are often referred to as *Web 2.0* or *RIA* [1]. Thus, ICEfaces is a framework for creating Web 2.0 applications (or RIAs). The other thing is facelets support. Facelets is a view technology for JSF that replaces the standard view handler which is JSP. It has amazing benefits like: Fast Templating/Decorators for Components and Pages, their APIs aren't dependent on a Web Container; they provide the ability to specify UIComponent trees in separate files (UICompositions), XML configuration files aren't necessary, and so much more. ICEfaces supports MVC architectural pattern. MVC is a design pattern which provides flexibility and improves the performance of the application by separating the application into three layers: Model, View, and Controller.

The design methodology used in this work is mainly architecture centered design method (ACDM). As it is explained in [4], ACDM provides techniques and a structure for designing the architecture and then using the architectural design to guide the programmatic aspects of a project.

3. The Proposed Solution

The output of the research is a web based school networking software. The ultimate goal of the development is to bring a solution to all of the issues that are stated in the previous section which are issues of communication and resource sharing between different types of educational institutions and involving the public and parents of students in the communication and resource sharing aspects as well. The different types of users that the software encompasses are: School Administrators (Representatives), Teachers, Students, Parents, and Public users (non-registered users). Except the public users all the rest will be registered into the system and they will have their own accounts to use the system.

3.1 Architecturally Significant Requirements

1. The system should be usable and should address usability attributes like ease of use, learnability, and consistency.
2. The system shall respond to search actions in not more than 5-7 seconds depending on the specificity of the search keys provided by the user.
3. The architecture of the system must support modifiability.
4. The system must incorporate role based security.
5. The system should use cryptography to secure user account.

3.2 Architectural View

The business process view is depicted by UML use-case diagram to show major business processes of the system as shown in Figure 1.

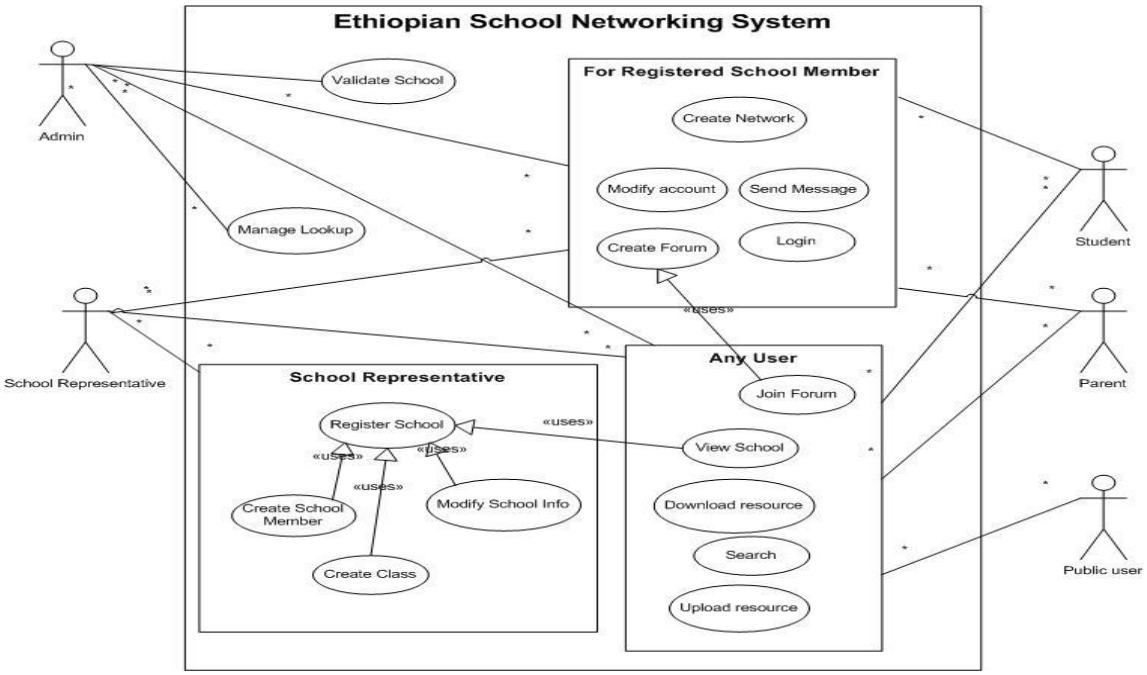


Figure 1: System use-case model

3.3 Deployment

Three-tier-architecture is chosen for the deployment. This architecture has three distinct tiers: presentation, application, and data tiers which represent the client side browsers, applications servers with Apache Tomcat (glassfish) servers installed on it, and a database server with MySQL 5.0 installed on it respectively as shown in Figure 2.

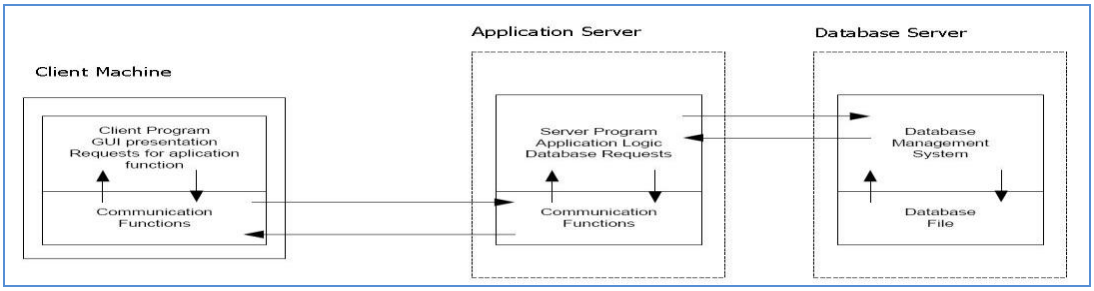


Figure 2: Three tier architecture

3.4 Chosen Architecture

The architecture chosen for the design of the software is MVC (Model View Controller). This pattern proposes a clear separation of concerns by separating the application into three which are model, view, and controller. The model layer of MVC is responsible for managing data. JSF implements the model layer with its managed Bean facility. The second layer of JSF's MVC framework is the View Layer which defines the user visible aspects (User Interface) of an application. JSF MVC architecture uses EL (Expression Language) to bind the view to the model. The third and the last layer of JSF's MVC architecture is the controller. It handles user interactions and navigation between views. JSF implements the controller layer with navigation-handler which by default supplies action plus outcome style navigation rules.

4. Prototype

The prototype is developed using ICEfaces MVC framework. The view handler used is facelets which adds advanced features to JSP which is the default view handler of ICEfaces. Templating feature of facelets makes the design modifiable, readable, and makes the process of software design easy and manageable. To give a highlight on how the software operates, the administrators/representatives of a particular educational institution will submit their personal information and the information of the school that s/he represents. This request arrives at the web-admin and s/he will examine the validity of the submitted information and will either accept or deny it. If the information is accepted, then the corresponding school representative's account will be activated which will allow him/her to login to the system and do all the tasks the system allows them to do. These tasks include registering class (which identifies a batch of students uniquely (e.g., Veterinary_1st_year), students, teachers, and parents. Once each of these different user types are registered and have their account, they will be able to login and access the system and perform tasks that are visible to them based on their role. The features of the software are defined in the previous section; hereafter we will briefly present how each feature is implemented by the prototype.

The software will have the following distinct features:

1. *Educational institution management:* This feature handles the following tasks:
   - Requesting school account by submitting school information
   - Approving submitted school
   - Updating school information
   - Blocking/Denying school
   - Viewing school information
2. *User management:* tasks included in this feature are:
   - Registering teacher, student, parent
   - Communicating parents of a student with:
     - School representative
     - Teachers
     - Other parents
3. *Class management:* the tasks of this system feature are:
   - Registering new class
   - Modifying existing class
   - Assigning teacher(s)/instructors to a class
   - Assigning/removing or changing students to/from a class
4. *Upload download management:* tasks handled here are:
   - Uploading resources (e-book, audio file, video)
   - Downloading resources

5. *Message management:* tasks handled here are:
   - Sending message to friends
   - Checking for messages (Inbox)
6. *Profile management:* the tasks included in this feature are
   - Updating personal information
   - Changing profile picture
   - Changing login password
7. *Network management:* this feature includes the following major tasks:
   - Sending friendship request to other users (school representative, teacher, student, parent) separately.
   - Approving or denying sent friendship requests.
   - Viewing friends to either send message or disconnect friendship.
8. *Forum management:* this feature includes the following:
   - Creating new forum
   - Posting to a forum
9. *Search management:* tasks included in this feature are:
   - Searching for a resource, person, forum, school.
10. *Locale management:* this feature is responsible for:
   - Handling locale of the system by changing language (font)

5. Related Work

Based on the problems that were identified to bring solutions to and the time allocated for the design and implementation of the entire work, a few related works were examined.

The first work is software called Moodle. Based on the definition collected from [2], Moodle is a web-based system that helps educators to create interactive online courses without having to know the ins and outs of web design and development. According to [2], Moodle is a versatile tool that allows students and teachers to work collaboratively online and it covers issues like: upload video, audio and links relevant to a lesson, engage students in a discussion forum, guide a real-time discussion or "chat", create, conduct and grade quizzes, prompt students with survey questions, post classroom blogs, create and maintain a classroom glossary related to a lesson, assign, collect, review and grade assignments, work collaboratively on a classroom Wiki or group-edited document.

Most of the things that Moodle does relate to the problems that this work tries to address, but again, Moodle does not provide information to the public about any of the educational institutions registered in it (regarding location, departments/faculties, courses, teachers and the like) and/or parents do not have a role to play in either knowing about the status of their children or by communicating with the rest of the school. These two features are proposed to be parts of this work.

The second related work that was reviewed is web-based software called TemariNet. Based on [3], temarinet.com is an educational social Learning Website created to promote a Healthy and Motivated Responsible Student Community. It mainly focuses on providing a medium where teachers and students communicate (colleges and universities) and learn from one another by communicating via forums and blogs. Based on the review, like Moodle, temarinet.com only focuses on the students and teachers of a particular college or university but not the public or parents of the students.

Temarinet.com somehow covers the two major problems that this work tries to bring solutions to which are communicating members of educational institutions and allowing resource sharing. But, it gives services to colleges and universities only. Plus, it does not let the public get information about the colleges or universities or the services that are using it. Parents are not included here as well; just like the previous one. On the other hand, temarinet.com lets users from anywhere to sign up and the administrator has to approve them individually so that they can get access to the system which we believe is too much burden on the administrator(s). This task could have been handled by the school itself using the representative of the school. But again, since temarinet.com does not register school information, the one way to handle sign ups is probably by the administrators.

6. Conclusion

Using the implemented prototype, the public can get information about registered educational institutions, students/teachers, the rest of the school members can interact with members of other schools to communicate and share resources, and parents can interact with the school of their children. On the other hand, this work contributes to the science of software engineering by showing how to design software using MVC architectural style. It also illustrated the overall software designing process following the appropriate steps and procedures. In addition to these, it incorporates ICEfaces technology for the implementation of the prototype which has advanced features as mentioned in the previous sections. For these reasons, this work contributes to the society and the science of software engineering.

References

[1] http://www.smart-soft.com/ice/gen/ aa_Day1/bb_WhatIsIcefaces/index.html, last accessed on March 12, 2012.

[2] http://orvsd.org/tutorial/what-moodle: last accessed on January 20, 2012.

[3] http://en.wikipedia.org/wiki/Service-oriented_architecture: last accessed on March 23, 2012.

[4] Anthony J. Lathanze, "Architecting Software Intensive Systems", CRC Press, London, 2009.

SOA Business Process Interoperability with TOGAF - The Case of Microfinance Customers

Rahma Hussein
rahma.huss@gmail.com

Fantahun Abera
PhD Candidate, Addis Ababa University, Ethiopia
fanetsa8@yahoo.com

Abstract

When customers seeking MSE (micro and small enterprise) related services, it is required to fill multiple forms at each participating institution and expect validation at each stage. The validations of the forms, that is, information provided by the customers and by the respective institutions depend on each other. There is lack of a communication system between the participating institutions and lack of dependable organizational interoperability, as the institutions perform according to their own plans, procedures and timings without consideration of the completion of service provision to the customer. To have a better insight on the design of the architectural framework, this paper discusses in detail about the business process, organizational structures, and business rules of the participated organizations in the interoperability framework. In the framework design process, it is also needed to select and use appropriate standards, tools, guidelines, and architectural patterns that should be used in the process of designing the multilevel form validation architectural framework. It is also discussed in detail about how to apply SOA in TOGAF architectural framework

Keywords: Multilevel form validation; Business Process Interoperability; Architectural Framework; SOA

1. Introduction

Understanding business processes across multiple organizations requires understanding the connections that exist between the organizations, the degree of commonality in business processes, the governance rules applied on the process, the flow of information across those processes, and the technology used to create those connections. A consistent approach to defining and managing processes, both internally and with external partners, is critical. It should also be kept in mind that whether a process is performed internally, or in an interoperable framework, appropriate evidence (records) of the transactions and activities will need to be created and stored for future reference [1]. It aims to improve the ability of organizations to respond to new requirements quickly and effectively by providing a common language and the ability to develop a common understanding of business processes and business requirements. This is particularly important as the need for cross-organizations collaboration increases with the drive towards 'connected government'.

In general, this paper is intended to grasp the service provision by each organization in the validation of customer forms filled in request of MSE group formation, business license, and loan at microfinance institutions. Accordingly, identifying the problems related to defining the interoperability between the organizations in the process, the governance rules applied on each organization and standardizing the services and design the architectural framework is considered. The intension is to design the business processes interoperability for solving interoperability problems and to simplify the process of the MSE group formation and loan appraisal, as multiple organizations participate in the process of validating, approving and updating customers' data.

The rest of this paper discusses on what are the problems, in relation to the research question and the objectives of the research. The significance of the research and the methodology of the research will lead us to review literature. The literature review will revolve around the supporting documents in relation to the research questions and objectives. The document also mainly focused on the design of the architectural framework and discusses the governance and the case studies in related to the implementation and testing of the work.

2. Background

Microfinance is often defined as a financial institution that is organized to give financial services for the poor and low income people [2], sometimes called "banking for the poor "or "rural banking" because microfinance is an amazingly simple and proven idea that empowers poor people around the world to pull themselves out of poverty by giving small loans for the purpose of starting, establishing, sustaining, or expanding small businesses. Microfinance institutions share similar vision and missions, as the strategy of the government of Ethiopia, which is clearly associated with serving people who are under poverty and not getting banking facility or who are living in un-banked and under-banked areas. The government also focused on MSE as one of the main strategies to reduce poverty and bring about sustainable development in the country in creating jobs for the urban unemployed. To this effect, MSEs need some capital to do their business to be successful on the strategy of poverty eradication and the government created different workflows to process this MSE strategy by incorporating microfinance [3].

For this work, the case study is taken at Addis Ababa and Oromia regions by focusing on the MSE loan of the MFIs. Customers seeking MSE related services are required to fill multiple forms at each participating institution and expect validation at each stage. Basically, the participating institutions comprise close-by woreda administrative offices, cooperatives, associations, MSE offices, MFIs, and training institutions. The validations of the forms, that is, information provided by customers and by the respective institutions depend on each other. This dependence creates serious overhead on the customers as the institutions business processes perform independently and there is no end-to-end process management for the series of form validations among the institutions. There is no known standard mechanism of information exchange between the institutions, and the chore of carrying the required information from one institution to another is the sole responsibility of the customer, creating customer dissatisfaction and reduced interest in seeking MSE services.

To solve problems related to customer service, operations and process management literatures recommend the creation of single window systems whereby the collaborating institutions perform back office activities and are visible to the customer through one contact point. According to the information obtained through data collection and observation made in Addis Ababa, woreda 03, this same strategy is also exercised by the Addis Ababa region. This approach creates overheads on the participating institutions as it doesn't take into account the specific organizational missions they are organized for. So another solution is making interoperable the organizations without changing their working structure and business process. To make interoperable, it is crucial to understand business processes across multiple organizations and requires understanding the connections that exist between the organizations, the degree of commonality in business processes, the governance rules applied on the process, the flow of information across those processes, and the technology used to create those connections. The identification of common processes and services within the government has the potential to achieve economies of scale, reduce duplication, and improve seamless service delivery across the government. In designing the business process interoperability, it is also crucial to select the best architectural framework that is compatible with the existing system. For this work, the TOGAF Enterprise Architecture framework [4] is selected for developing the architectural design document, as it is the one adopted for the Ethiopian National Enterprise Architecture [5]. Using SOA with TOGAF is also applied on the application framework of the architecture on the designing of service as a means of communication between the organizations.

3. The Proposed Solution

The proposed solution is the deliverable container for the core architectural artifacts created during the research. The architectural design started by defining the architectural vision. It clarifies the purpose and demonstrates how it will be achieved by the proposed architecture development. Key elements of the Architecture Vision such as the enterprise mission,

vision, strategy, and goals have been documented. This has its own lifecycle within the enterprise. The next step is defining and discussing with the business architecture (Phase B of ADM) [4].

The Business Architecture defines the business strategy, governance, organization, and business services and processes. The objective of business architecture is to develop the Target Business Architecture that describes how the enterprise needs to operate to achieve the business goals, and respond to the strategic drivers set out in the Architecture Vision, in a way that addresses the request for Architecture Work and stakeholder concern. According to the case study, there is no existing architecture description in the organizations that should be used as the basis for the baseline description. Where no such descriptions exist, information will have to be gathered in whatever format comes to hand and should start by studying the baseline architecture as a high level and document it as an input for the target architecture. So, a knowledge of the Business Architecture is a prerequisite for architecture work in any other domain (Data, Application, Technology), and is therefore the first architecture activity that needs to be undertaken. A variety of modeling tools and techniques are employed like BPMN [6], SOAML [7], and others to model different architectural views that are produced in each step of the ADM. Some of them are AS-IS and TOBE BPMN, Services, Data and sequence diagrams that show us the cherography of the interfaces in the service information exchange matrices.

The next step is service identification for the purpose of defining the target application architecture on the domain/sub-domain. All the domain teams (excluding the data and infrastructure domains) need to produce target application architecture at the conceptual and logical levels for their respective domains. The starting point for the artifacts that are developed in this phase is the set of key business requirements identified in Phase A (Business architecture). For the kind of enterprise SOA that are discussed here, the following artifacts should be used because they contribute to the definition of SOA building blocks in Phase C (Application Architecture). The other information that is used as an input for the design of the application architecture is the national enterprise architectural framework of Ethiopia Service Delivery channels [8] which should support open standards in order to allow access for a wider range of client access technologies and devices to interact with government services and exchange information with these services. On the service delivery solution component, authentication is used to authenticate citizens before getting the services. So in this architectural design, the MSE groups should authenticate as a citizen before getting the MSE services and the designed multilevel form validation middleware should have to get the citizen information from the citizen information system. If the citizen is not registered, the middleware should request the citizen to insert all the necessary information and the middleware updates this information to the citizen information system. So, on this service design to integrate the citizen information system and the multilevel form validation, it is necessary to build a service. Services are modeled using SOAML. The first reason is that the TOBE diagram and BPMN diagram shows us functionalities and their process interactions between the organizations will be transformed to the service using IBM Rational web sphere. The other reason is SOAML provides the capability to create and leverage an architecture that helps people, organizations, and systems to collaborate via services and shows how those services connect to other parts of the architecture such as processes, information, and business rules.

Service identification is one of the first steps in the Service Oriented Development life cycle. The process of identifying services can be approached from a number of perspectives and in combination with a particular project type. There are different approaches used to identify services form BPMN models. Some of the approaches and tools mentioned in different literatures are top-down and bottom-up, and exposing existing assets. Top-Down – Business Process Driven (Value-Chain Analysis) is the most preferable approach and best practice to start with for the enterprise transformation to SOA. The first step to identify the services from BPMN using SOAML

[9] is service identification. It is done by extracting candidate services from the TOBE BPMN diagrams. As can be seen from Figure 1, the service identification will start by identifying specific capabilities that may be exposed as services. Capabilities identify or specify a cohesive set of functions or resources that a service provided by one or more participants might offer. Capabilities can be used by themselves or in conjunction with participants to represent general functionality or abilities that a participant must have. The next step is to identify the data flows between these services or tasks. These data objects will be associated with message types or data entities in SOAML. It focuses on the tasks that describe the interaction points between the business entities. These interaction points will be associated with service contracts in SOAML. Finally, the choreography [10] defines a specification of what is transmitted and when it is transmitted between parties to enact a service exchange. It defines what happens between the provider and consumer participants without defining their internal processes. Their internal processes have to be compatible with their service contracts. Modeling the behavior of service contract in order to get a better understanding of the interaction between the roles is done using sequence diagram.

Based on the candidate services, data and choreographic design, the application view of the target application architecture at the logical level is designed.

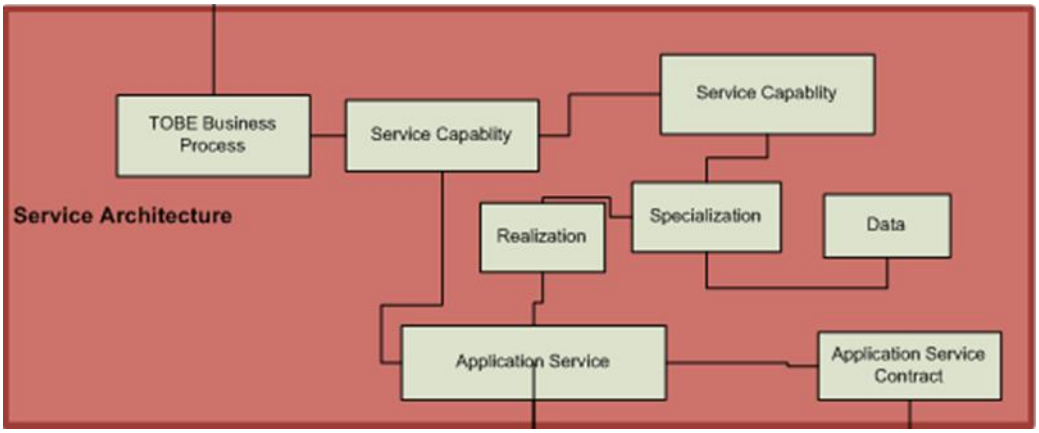


Figure 1: Service Identification

Finally the architecture framework is a foundational structure, or set of structures, which can be used for developing a broad range of different architectures. It describes a method for designing a target state of the enterprise in terms of a set of building blocks, and for showing how the building blocks fit together. It contains a set of tools, standards and deliverable channels, and deliverables in each phase of the development steps and building blocks. Using TOGAF [4] results in enterprise architecture that is consistent, reflects the needs of stakeholders, employs best practice, and gives due consideration both to current requirements and the perceived future needs of the business.

In general, the architectural framework incorporates the following components:

a) Deliverable channels,
b) Customer/Citizen applications,
c) Service layer/Middleware service for multilevel form validation,
d) Organizational services/functions,
e) Applications or portals that can be used by the organizations, and
f) Standards and guidelines, goals, objectives participated organizations, roles, actors, functions, business services, and processes.

4. Experimentation/Prototype

To test the framework and to design and modify the governance that should be applied, it was necessary to discuss with different organizations that participate in the business process interoperability. One of the selected organizations that participated in the case study was MSE office. The key issues discussed on the framework testing were to establish which organizations were responsible for which process and which staff/actor in the organizations

was responsible for which tasks and activities. The other is the prototype design using the web service on J2EE environment to show the designed framework

5. Related Work

Different organizational interoperability are designed in different countries based on their business process, service delivery provision, and governance rules. The Australian Government Interoperability Framework (AGIF) [1] is a key element in the implementation of the e-government strategy, especially in relation to building connected service delivery and achieving value for money.

The Australian Government Business Process Interoperability Framework (BPIF) is one of several frameworks which combines to form a foundation for connected government. The Business Process Interoperability Framework (BPIF), in association with the Australian Government Architecture (AGA) provides agencies with principles, policies, tools, standards, and guidelines for working together. The framework provides the means for individual and collaborating agencies to map business processes with a view to identifying areas of commonality and opportunities for integration or collaboration within an agency and with other agencies [11]. Understanding business processes within an agency or across multiple agencies encompasses understanding the connections that exist between agencies, the degree of commonality in business processes, the flow of information across those processes, and the technology required to facilitate those connections. Business process interoperability enables collaborating agencies to share processes for the achievement of a common goal or for delivering similar services.

The BPIF also provides a roadmap which operates as a high-level methodology for undertaking business process interoperability in government. A number of industry standards and proprietary solutions exist to support agencies to map, model, and manage processes. Undertaking business process interoperability requires a common set of standards, methodologies, and frameworks. Some emerging standards that are used to model business processes include Business Process Execution Language (BPEL), Unified Modeling Language (UML) and Business Process Modeling Notation (BPMN) [12].

A defining characteristic of business process interoperability is that two or more autonomous organizations jointly execute a process with the purpose of creating a certain output that can serve as an input to other organizations. Some specific challenges exist at the boundaries of organizations.

- Business processes of different organizations are often perceived as a “black box” since process activities and their interdependencies with internal processes are unknown to the internal staff. In order to allow for basic coordination among business partners and clarify interdependencies, the future process architecture needs to reflect external process integration [13].
- In business process interoperability, responsibilities for the different process activities are shared among two or more organizations. Process coordination at the boundaries of organizations is often performed independently due to inadequate clarification of responsibilities. Process models have to provide means to split responsibilities among different organizations and allocate tasks to specific actors.
- Business process design may contain specific organizational know-how that represents competitive advantage. Furthermore, processes contain confidential information, giving rise to a need for organizations to hide internal details of their processes. The future process architecture needs to provide concepts for providing selective transparency of internal processes.
- Boundaries between legal entities have to be treated differently than boundaries between internal organizational units. It is usually governed by contracts and legal frameworks specifying service levels and prices. Process design has to formally describe the interfaces at the organizational boundaries [11].

6. Conclusion and Recommendations

For this work, some of the appropriate TOGAF enterprise architectural framework artifacts and catalogs are applied. The design is done by the point of view of the enterprise architect. In general, this work is a starting point for our country in the field of software engineering. Because the National EA framework of Ethiopia is planned to be implemented using TOGAF so this will show us TOGAF can be applied for our country enterprise architecture. The research also shows the implementation detail of the service architecture of the TOGAF reference model. As a recommendation in the future, it is necessary to study in detail on the governance that should be applied and implement and test fully the security of data exchange between the participated organizations and the real time multilevel form validation option. The other is the multilevel form validation middleware prototype should handle any change made on the services.

References

[1] "The Australian Government Business Process Interoperability Framework Enabling Seamless Service Delivery," July 2007.

[2] "What is microfinance?" http://www. microfinancegateway.org/p/site/m/template.rc/1.26.12263/.

[3] "Socio Economic Determinants of Small Manufacturing Enterprises' Growth," July 2004.

[4] "Architectural Principles" *TOGAF 9.1*. http://pubs.opengroup.org/architecture/togaf9-doc/arch/.

[5] Final e-Government_strategy_Implementation_ Report Ethiopia_114.

[6] B. M. Owen, J. Raj, and P. Software, "BPMN and Business Process Management Introduction to the New Business Process Modeling Standard," *Management*.

[7] A. M. D. Solutions and C. Casanave, "Service Oriented Architecture Using the OMG SOAML Standard," *Management*, pp. 1-21, 2009.

[8] "National_EA_Assessment_Report_V1_ Draft.".

[9] S. Integration, J. Bpmn, S. F. T. F. Meeting, O. M. G. Tc, S. Antonio, and T. Am, "BPMN 2.0 and SOAML Integration," *Integration The VLSI Journal*.

[10] A. Karande, M. Karande, and B B Meshram, "Choreography and Orchestration using Business Process Execution Language for SOA with Web Services," *International Journal of Computer Science Issues IJCSI´11*, Vol. 8, No. 2, pp. 224-232, 2011.

[11] U. Greiner, C. Legner, S. Lippe, and K. Wende, "Business Interoperability Profiles: Relating Business Interoperability Issues To Technical Interoperability Solutions," *Business*, pp. 1-12.

[12] O. M. G. D. Number and A. S. Files, "Business Process Model and Notation (BPMN)" *Business*, January 2011.

[13] S. Gallen, "The Challenges of Inter-Organizational Business Process Design – A Research Agenda" pp. 106-118.

Reengineering Open Source CMS using Service-Orientation: The Case of Joomla

Tagel Gutema
tagelgutema@gmail.com

Dagmawi Lemma
Department of Computer Science, Addis Ababa University, Ethiopia
dagmawil@yahoo.com

Abstract

To fill the gap of the business requirements on an already existing open source product, reengineering can be performed to make it flexible and used by other systems. This can be achieved by using appropriate technologies and methodologies. Service-Orientation is an emerging technology that is currently being used as facilitation for reusability. Although service-orientation is applied for different types of systems, its application on open source CMS is questionable with respect to simplicity, time, and experience. This paper provides guidelines and prototypes in using service-orientation for this type of specific products.

The paper reengineers an open source CMS (Content Management System) after a thorough understanding of the open source system, in this case Joomla. The paper tries to apply service-orientation on Joomla which in turn enables to expose its component services for use. The methodology that is employed to reengineer the case is a SOSR (Service-Oriented Reengineering). SOSR involves basically on creating the 4+1 view models revealing the existing system to be reengineered in order to create a visual model. Once the existing system is understood and visualized, SOFSE (Service-Oriented Forward Software Engineering) is applied for service-orientation. To realize the service-orientation, the paper uses common technologies that facilitate SOA.

Keywords: Content Management System; Business Process Execution Language; Service-Oriented Architecture; Reverse Software Engineering; Service-Oriented Forward Software Engineering; 4+1 view; Service-Oriented Software Reengineering

1. Introduction

It is crucial to define the term reengineering before introducing the substance of this paper.

Software reengineering is defined in [1] as the study, examination, and modification of the internal system or functionality of an already existing system or product so as to reconstitute it in a new form or design with new features, often to take advantage of newly emerged technologies, but without significant or major change to the native functionally and purpose of the system. Similarly, it is explained in [2] that it is to first maintain the existing functionality, then prepare the existing system for the changes and finally add fresh technology.

Reengineering can be applied on an already existing open source product or software. Open source products could be one of the preeminent opportunities to use them as we find them freely and specifically we can consider open source Content Management Systems (CMS). Open source CMSs can be customized and used for several business needs and requirements. However, a single open source CMS may not satisfy the whole needs of an organization as it has a limited number of functionalities. Hence, looking for another open source module or component might be a solution to fill the gap of the business requirement. It is less likely to make different CMS components homogenous, as each may have been built to meet specific requirement which may diverse from others. Besides, each of these systems has specific objectives and are also implemented in different technologies.

To use open source product modules or components, one way is to apply service-orientation and expose them for reusability and this experimentation will be done in this paper. This can be achieved by using appropriate technologies and methodologies.

2. Statement of the Problem

Organizations always formulate a cost benefit analysis before making any investment on Systems. The investment analysis result may fall on developing a new system or reusing the already existing product that may be used with the old one or buying a new product. Considering the second option, i.e., reusing already existing products, is time saving. But, due to the nature of the heterogeneity of systems, making use of an existing system with less modification may not be an easy task. This could be an obstacle to meet the desired target within a short period of time.

In the old days applications were developed to address a specific business need and to add additional business requirements in the future systems to be developed should be flexible enough so that they can be able to meet frequently changing and new business requirements. The traditional methods of developing and deploying software systems is no longer sufficient due to the difficulties in developing a large number of software subsystems in a short period of time [3]. A new approach is mandatory to provide flexible system that could meet the frequently changing business requirements or needs of the time. In an effort to alleviate challenges in the software process, several models have been proposed and are evolving (such as scrum, extreme programming, etc.). However, addressing the dynamic nature of business requirements may not be solved by focusing on the software process; the type of software architecture applied also matters. Besides, heterogeneous components of different products may not easily work together with new requirements. Therefore, a new approach that makes system components easy and flexible is essential.

Open source CMSs are available in different forms and making use of these product components and modules directly is impossible due to their nature. Therefore, looking for a method or a solution that makes these products more flexible and accessible for reusability is essential.

3. Objectives

The general objective of this paper is to address the problems specified in the problem statement section with an SOA (Service-Oriented Architecture). "An SOA is an architecture that utilizes the core concepts of service providers and service consumers to define a system "[4]. SOA solves problems through the use of reusable components. In SOA, components are developed to solve a specific business problem. These components can be threaded, linked, or integrated into a specific configuration for the current business needs. When new business requirements arise, the system is flexible to be reconfigured [5].

The specific objectives of this paper are to:

- understand the basic principles of SOA and apply it an open source CMS pertaining to the principles of reengineering.
- enable service-orientation on an open source CMS, in this case Joomla
- introduce a guideline on how to enable service orientation on Joomla.

4. Methodology

Open source CMSs are being solutions for multiple business needs. They can be customized and used for several requirements.

SOA requires at least service reusability and autonomy to satisfy its fundamental principles. Most old systems are not implemented in the context of service orientation and hence reengineering them to comply with the SOA principles could be essential [6]. In this paper, we used the SOSR methodology and extended it for a service broker that composes a set of services as a composite service(s). The SOSR methodology has two phases - RSE and SOFSE. The former one is used to reverse Joomla so that we can visualize with the 4+1 view and RACI chart based on UML diagram and the later one is to model Joomla into the modern target system in the context of service-orientation [6].

5. Literature Review

This section describes the SOA realization in different perspectives from scientific points of view. The technologies described in this section are in detail defined in Section 5.1.

SOA is a new technology choice to integrate systems to improve business processes in an

organization with the right tools, skills, and methodology to deliver reusable software applications [7]. In the 21st century, only few organizations are adopting SOA to meet their business needs. SOA is still in its infant stage and many researches are being made towards SOA. These days the word SOA is replacing the traditional middleware as the software technology shifting to service orientation. Dependent, inflexible, and incompatible systems are not suitable for modern IT business as the architectures are expected to be agile and have to respond quickly for the ever changing business needs [5].

A framework is proposed in [8] to enable SOA on CMS systems. This shows that the application of SOA may lead to use a framework to apply SOA. This paper applies SOA on open source CMS (not to create a framework).

Applying service-orientation on a CMS requires understanding the principles of SOA. SOA uses the basic principles of web services to impact service orientation. As web services are standardized across multiple vendors, they are the main implementation techniques for SOA. SOA is a concept that can be influenced by different frameworks and vendors as the web services influence SOA in shaping several of its principles including abstraction, service loose coupling, and service composability [9].

The core compositions of web service technologies - XML, SOAP, and WSDL - build the basic Web services architecture and through further Web service protocols that enable a qualified performance web service. Web services provide reliable, flexible, loosely coupled, and extensible middleware [10].

Analysis on service oriented architecture among different SOA solution vendors like Microsoft, IBM, Oracle, and Red Hat is carried out in [9] highlighting how and to what extent SOA can be achieved with open source solutions and what kind of competence is required for it. In addition to this, a comparison of different SOA solution vendors (both proprietary and open source) is made. The case was based on Vattenfall, which is one of the first five largest energy suppliers within Europe [9]. Having operational systems in different countries brings the challenge of integrating all these distributed systems and this integration is a vital requirement for Vattenfall which uses a proprietary product to integrate across different platforms but the research indicates that the company requires a better infrastructure to be extensible and cost effective. The result of the analysis would help the company to decide on SOA in most cost effective way [9]. To apply this, it requires a thorough understanding of SOA principles.

SOA governance is an essential factor to determine success of SOA [11]. The research result in [11] stated that strong governance and control were key factors to succeed with SOA.

It is shown in [12] that SOA is prominent to strategically align IT and business especially when implemented through the use of Web services. For example, while using HTTP, it is possible to employ widely adopted industry standards such as XML, SOAP, and WSDL which help to run an application on different platforms [12].

In [13], it is described that web service composition is an inevitable aspect to solve complex problems by combining available services and in [10] similar technologies that are being used in this reengineering project with SOA infrastructure are applied. In this paper, basic services are composed and selected as per the request and decision made by the orchestrator BPEL. In [14], it is tried that since it is difficult to compose web services manually; there has to be a mechanism to dynamically generate compositions.

Based on the technologies available and understanding of SOA principles and governance, there are possibilities to successfully apply service-orientation on a CMS system. Therefore, we would focus on reengineering and service-orientation on an open source CMS using the SOA principles.

6. The Proposed Solution

Before discussing the proposed solution, it is important to understand Joomla and how it works.

6.1 Joomla

Joomla is one of the open source CMSs which is released under version 2 of the GPL license.

According to [15], Joomla is one of the most popular CMSs which is proved by a number of awards and too many massive committees. Besides, it has large number of freely and commercially available extensions that help them to do more functionalities more than just simple content management.

There are only two entry points in Joomla: Frontend and Backend request. The Frontend request will only be instantiated through the root index.php and the backend administrator/index.php entry point. By doing so, Joomla minimizes security vulnerabilities as well. Basically, there are four operations - receive request and load libraries, initialize/build application, determine application route (determines selection of appropriate component and execute it), rendering the selected document, and respond appropriately.

It is important to make systems more flexible to service-orientation in this ever-changing environment. This means any software should be designed in such a way to facilitate easy reengineering. Fortunately, most open source CMS systems, including Joomla, are flexible enough to facilitate reengineering. Nevertheless, it is difficult to find a detailed design document made for open source CMS. Therefore, bringing the available source code into visual view requires a thorough study and time. Thanks to the Joomla community coders, most of the classes and methods are documented which, at least, help for code readers. Getting the proper location of component code execution initiation and return point is an essential part of the code where we need to make a focus. Determining the appropriate location where the client should be put, i.e., it could be on the front of the Joomla or in the place where it divides the Joomla into service provider and service consumer that means the service orientation will divide the system into different parts. This does not mean the "site" and the "administrator" as Joomla divides.

Irrespective of any architecture, it is possible to expose each component into the service consumer. The decision where the client code (in this case the SOAP client) should be put requires if the service provider is independent and properly designed to be consumed. Although, how to put the SOAP client and where to put it is decided by the designer. In this paper, it is put where the dispatcher starts to look for the requested components because we want to expose the components for service-orientation.

Joomla prevents direct access to the system. That means it only allows a single point of entry. This is done for the sake of security, especially to prevent a directory traversal attack. But, in this paper the orchestrator is accessing outside the system and Joomla, in its original form, does not allow the service provider. In this paper, it is located in the server where Joomla is located. Hence, this requires breaching the security. In this case, for example, the RACI chart will show the responsible and accountable person in the project. Based on the accountability, the responsible person shall consider the vulnerability to lock it or to put it in the appropriate place or to make appropriate solution for it. The RSE enables to visualize the Joomla system and the SOFRSE helps view the target system with its modern requirements. This solution gives how Joomla components can be exposed to any service consumer. As there are a number of open source CMS products, it is possible to use other Joomla components' output for any particular consumption if needed. The service consumption may not be only specific to open source CMS, but for any application if needed.

In the solution design part, we will create the SC, SB, and the SP based on the components as they will be considered as services when the system is reengineered. Therefore, reengineering into service-orientation is done on three of the components due to time limitation as the ultimate goal is to show as a prototype.

In this paper, due to time limitation, we consider three components: Content is used to display articles, content categories, and content sections, NewsFeeds are any series of news articles, and Contact is list of contacts information. The implementation of these components shows that it is possible to service-orient all the rest of the entire Joomla components whether from the frontend or backend as the way of access and the entry point of components is the same for both parts. The decision is made by the dispatcher based on user request. The access mechanism for the

frontend and backend is through index.php and administrator/index.php, respectively. The dispatcher dispatches to the appropriate component based on the input path irrespective of the front and the backend [15]. Therefore, selecting either from the front or the backend does not affect the service-orientation principle.

In Joomla, the frontend operations are usually page requests, viewing contents, search values, and the input is a page to be requested and similarly the backend is a page request and input values of the form if any [15]. Selecting a single component from the backend requires other associated components like user registration and creation and due to time limitation, we could not select from that part. However, as described above, this does not affect the representation of the selected components. Therefore, randomly selecting any of the components represent the entire components except the content component should be incorporated as it is a must to exist in Joomla to get contents.

7. Prototype

To implement the prototype, SOA implementation technologies (XML, BPEL, SOAP, WSDL, etc.) and other SOA implementing tools are used.

Figure 1 shows a screenshot after Joomla has been service oriented. It shows when Joomla is first instantiated. A result from the content component is also shown.

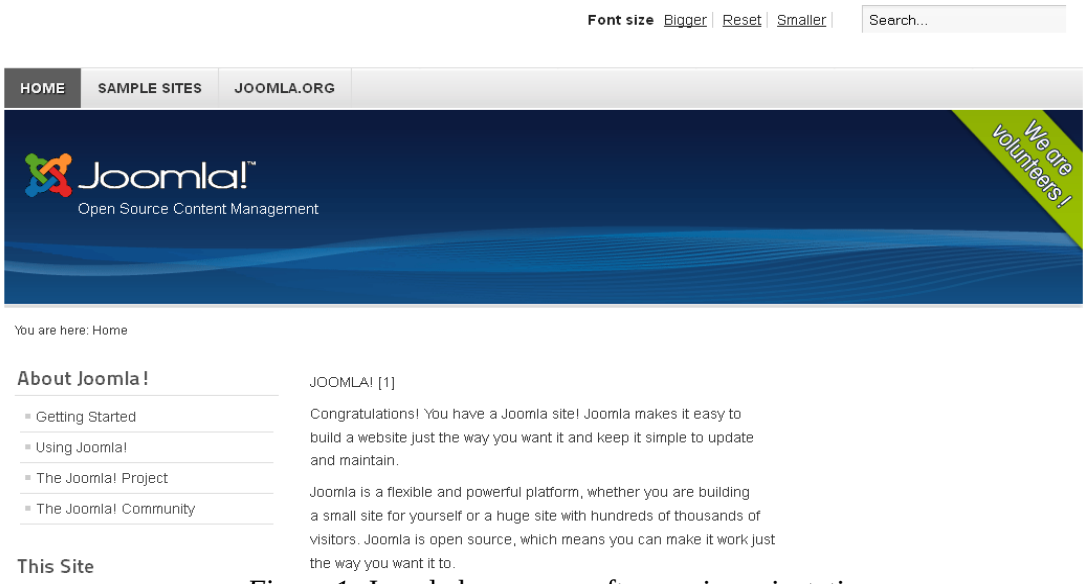


Figure 1: Joomla home page after service-orientation

8. Conclusion

It is shown that making a legacy system ready for integration by incorporating service orientation is to mean ready to be used by other applications for reusability. Any client system that requests the server gets the desired output. The SOA technologies enable independent or heterogeneous systems to communicate across a network and through the Internet. This paper tried to apply the concept of service orientation to an open source CMS (Joomla) and this has been done successfully on some components of Joomla.

SOA is usually for different large systems to come to a common purpose for an organization to meet its business needs. However, it is also possible to apply the concept of SOA on an application having too many components by applying interface description and passing message among various services through broker(s) with some modification of both the server and client side code in a way making appropriate for service orientation. The depth of modification may vary form system to system. If the system is very old, it may require too many modifications as it usually becomes tightly coupled and difficult to divide into components or modules. When services are exposed to another application, an organization can benefit from the concept of service orientation since it brings various open source components composed with services orientation

technologies and available to the business need by incorporating into other legacy or modern systems within a relatively short period of time instead of developing a new component or system from scratch.

References

[1] "http://www.answers.com/topic/ reengineering#ixzz21KtigFCX

[2] "http://www.hiddenbrains.com/articles/tag/re-engineering-services-business-reengineering-software-reengineering-service-hire-software-developer"

[3] C. L. Liu, N. T. Hua, and A. B. Tucker, "PRACTITIONER' S".

[4] M. B. Juric and M. Krizevnik, "Composite Applications with Oracle SOA Suite 11g".

[5] M. B. Juric, R. Loganathan, P. Sarang, and F. Jennings, "SOA Approach to Integration".

[6] S. Chung, J. Byung, C. An, and S. Davalos, "Service-Oriented Software Reengineering: SoSR," 2007.

[7] J. Melton and W. Group, "Praise for FastSOA".

[8] "The Undersigned Faculty Committee Approves the," 2006.

[9] "Analyzing Service Oriented Architecture (SOA) in Open Source Products". 2010.

[10] Y. Wang, "Web Services for a Software Development Platform," 2010.

[11] J. Franzén, "Shared Experiences from Five Organizations moving towards a," 2008.

[12] J. Lundberg, "Service Oriented Architecture & Web Services", 2008.

[13] E. Krankoc, "Web Service Composition under resource allocation constraints, 2007.

[14] E. K. Kuban, "Abductive Planning Approach for Automated Web Service Composition using only user specified inputs and outputs", 2009.

[15] J. K. Chuck Lanham, Mastering Joomla 1.5, Extension and framework development, second edition, The professional guide to programming Joomla, 2010.

Usable User Identification Technique - The Case of European Union Remote Web Access

Temechu Girma
Delegation European Union to the African Union, Addis Ababa, Ethiopia
temechug@gmail.com

Dagmawi Lemma
Department of Computer Science, Addis Ababa University, Ethiopia
dagmawil@yahoo.com

Abstract

Remote Web Access (RWA) has become more common and widely used in different organizations [1]. RWA provides a central web location for remote users to access corporate resources. It also enables to access an email application from a remote location. The European Union (EU) is an organization that uses RWA. However, the user identification technique associated with RWA is one of the key challenges that need to be addressed. To protect the organization from an attack (such as cyber or any other), the EU decided to add more identity checking steps on RWA to identify the user when they try to login remotely. However, survey results show that RWA has a usability issue which is compromised with the security in RWA. Qualitative and quantitative data were collected through a questionnaire, focus group discussion, interviews and document analysis. Usability testing software was used as a tool to assess usability of RWA and user satisfaction.

The study showed that adding more identity checking steps in RWA has an inverse relation with usability and increases the perceived security and trust in RWA. Moreover, the overall usability of the UIT in terms of effectiveness, efficiency, and user satisfaction is very low.

Besides, the more the user identification steps increase, the user frustration of RWA also increased. To minimize this frustration, RWA users start to write a note with a piece of paper to remember their secret such as username and password for ECAS application, domain, and delegations. All these actions make them exposed to intruders (attackers). Moreover, they are not only exposed for stealing their data but also, as a whole, the organization UIT will be vulnerable to attack. Therefore, the new user identification technique of RWA makes them unhappy.

To alleviate the usability problem, an enhanced usable user identification technique (UIT) for RWA is proposed considering the relevant security features regarding UIT and usability of secured system which is looking into consideration of all assessment results.

Keywords: Usability; Security; User Identification Technique; Remote Web Access

1. Introduction

Usability and security are part of design goals that indicate success of the design and the overall quality of software application [1]. According to Nielsen [2], usability is a quality attribute that assesses how easy and satisfactory the User Interface (UI) would be. Also, usability focuses on how users perform their tasks efficiently with minimal or no error. On the other hand, security is a system attribute that reflects the ability of a system to protect itself from internal/external attacks, which can be accidental or deliberate [3]. Security usability is concerned with the study of how security information should be handled in UI. Moreover, it deals on how security mechanisms and authentication systems themselves should be easy to be used [4]. This paper focuses on usability of user identification technique (UIT) on Remote Web Access (RWA).

RWA is a web-based application which is accessed through a web browser. It provides access for remote users to facilities such as email, reading/modifying their data, and enables them to remotely control a machine as if they are sitting in front of it [5].

The European Union is one of the organizations that use RWA. A serious cyber attack on European Union's mailing system has been reported in March

2010 [6]. Because of the attack, there was a "risk" of "unauthorized information disclosure", a way of saying that outsiders may have been able to gain access to even more confidential and sensitive information. It has been indicated that both the Commission and the European Union External Service (EEAS) are "the subject of an on-going widespread cyber attack" [7]. According to the official report "A detailed investigation and assessment of the threat is under way." Meanwhile the RWA of email has been closed for all the EEAS users.

Six months later, RWA of email re-launched UIT with very strict and secured steps that make use of Short Message Service (SMS) authentication. However, these authentication methods have laid down unforeseen influences on usability of the email application when it is accessed by users remotely.

In order to confirm the influence of the new UIT, a preliminary assessment has been conducted. The result showed that the user identification and authentication methods have a significant influence on the usability of RWA.

In order to answer the research question and address the objectives of the research, primary and secondary data collection methodologies were used.

Interview, focus group discussion, questionnaire, and usability testing were used to collect primary data from selected EEAS staff that use RWA.

These were considered to

- find information on the current practice of usability and security (user identification steps) in EEAS,
- know the extent of user satisfaction with RWA, and
- identify the major usability related challenges in user identification steps to access email via RWA.

Usability testing has been conducted to assess how easy is the UI for users of RWA and to measure user satisfaction. Moreover, secondary data from documents such as Information Technology (IT) Security Policies, Procedures, Guidelines and Reports of IT of EEAS were reviewed as a source of information.

2. Usability and Security

2.1 Themes on Usability

It is important to realize that usability is not a single, one-dimensional property of a user interface. There are five usability attributes [8]:

- *Learnability*: The system must be so simple and easy to learn that users can learn it quickly in order to accomplish their work on the system without difficulty.
- *Efficiency*: The system should be efficient to use and save time for users when they work with it. The most common measures of efficiency are time to perform a particular task, number of key presses or interaction to achieve the task, number of screens visited to complete a specific workflow scenario, number of back buttons uses, and time to execute a particular set of instructions.
- *Memorability*: The system should be easy to remember by the users so that when they use the system after some time, they should not need to learn everything again for doing their work.
- *Effectiveness*: The system should have a low error rate, so that users make few errors during the use of the system. The common measures of effectiveness include number (rate) of error, path taken to complete the task, severity of error, and requests for help.
- *Satisfaction*: The system should be pleasant to use, so that users are subjectively satisfied when using it; they like it.

According to ISO standard, usability is "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use" [9].

2.2 Themes on Security

Information security is the protection of information from a wide range of threats in order to ensure business continuity, minimize business risk, and maximize return on investments and business opportunities.

Security systems are designed to let authorized people in (the permission problem), and to keep unauthorized people out (the prevention problem) [10]. This involves three distinct steps: Identification, Authentication, and Authorization.

Identification occurs when a user (or any subject) claims or professes an identity. This can be accomplished with a username, a process ID, a smart card, or anything else that can uniquely identify a subject. Security systems use this identity when determining if a subject can access an object.

User identification is required for the following reasons [11]:

- Authorization: deciding whether a user is authorized to use the system.
- Policy enforcement: applying the right policy to the user (e.g., Security, Logging and HTTPS policies).
- Auditing: tracing user activity through logs, i.e., recording (logging) of transactions with details for future viewing and analysis of activities performed by the user.

3. Findings and Discussions

To find out the current practice of usability and security (user identification steps), to know the extent of user satisfaction with RWA, and to identify the major usability related challenges in user identification steps to access email via RWA, interview, focus group discussion, questionnaire and usability testing were used to collect primary data from selected EEAS staff.

Moreover, usability testing was taken as a method to assess how easy the UI for users of RWA is. Additionally, secondary data from documents like IT Security Policies, Procedures, Guidelines, and Reports of IT of EEAS were reviewed as a source of information.

Based on the above fact finding methodologies, the following results were found.

3.1 Questionnaire Survey

The respondents have the main role in performing the usability test and usability survey. They are classified based on their experience.

a. General Usability Perception

For 49% of the respondents RWA is not easy to use but for 31% of them, RWA is easy to use. The remaining 20% of them did not perceive that RWA is easy to use or not.

34% of the respondents agreed that RWA has attractive presentation but 56% of respondents do not agree with the attractive presentation of RWA. The remaining 10% of the respondent neither agree nor disagree.

Around 34% of the respondents agreed that RWA is interesting and engaging, but for 47% of respondents, RWA was not interesting and engaging. For the remaining respondents, RWA is neither interesting nor engaging

The general usability perception is less than average. Figure 1 shows the general usability perception of RWA in terms of its attractiveness, easiness, and satisfaction.

b. The Content of the Page

For more than average of respondents, there is a problem regarding "easy to understand and follow" the content of the page and "clarity and simplicity" of Language while to understand and follow the content of the page of UIT in RWA.

c. Security Perception

Most of the respondents strongly agree (62%) that they receive warning messages when the communication is insecure. Moreover, the remaining 38% also have the same (agree) opinion.

About 93% of the respondents believe that if RWA site addresses contain 'https', they are secured. The remaining 7% of respondents do not decide.

Around 69% of the respondents believe that if they are connecting to RWA with acceptable time limit, they will perceive that they are secure, while the rest of 31% also agree with same condition.

65% of the respondents have a strong trust on RWA and 18% of them also agree that they have a trust on RWA. But around 17% of them do not decide on this matter.

From the above survey result, it can be concluded that as security perception on RWA has marginally positive, the trust for UIT in RWA is also good.

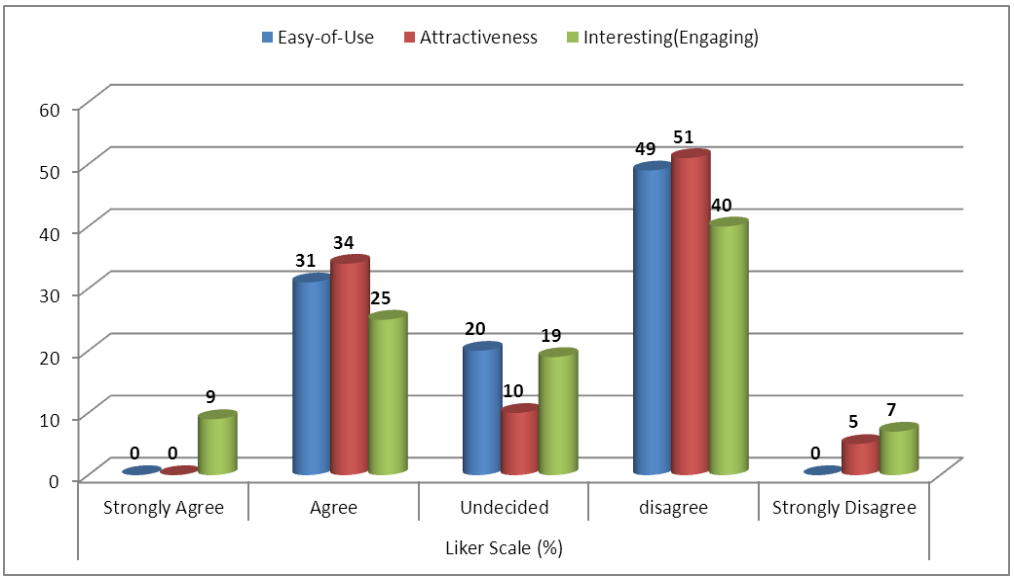


Figure 1: General Usability Perception

3.2 Usability Test Result

During the test, users have been observed while performing their tasks.

a. Task Completion

From the total participants, 66.66% of them successfully complete their tasks while 33.34% of them did not finish their tasks. Figure 2 summarizes the completion of tasks per participant.

One novice user and one expert user did not successfully complete their task. Usability is a generally relative measure of whether a software product enables a particular set of users to achieve specified goals in a specified context of use [12].

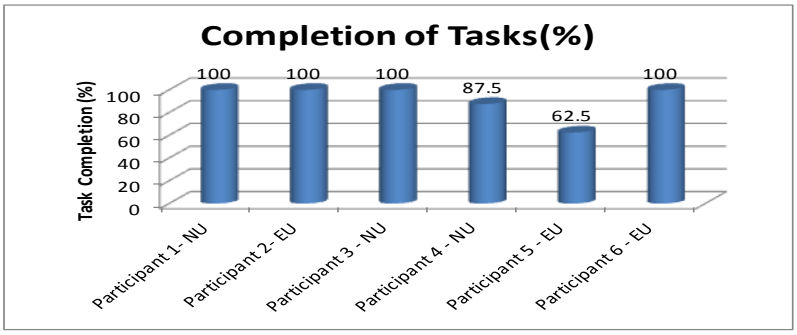


Figure 2: Completion of Tasks per participant

b. User Satisfaction Analysis for Proposed System

According to Aaron *et al.* [12], SUS is an effective tool for assessing satisfaction of a product. Thus, Figure 3 shows the mean score of SUS result per participant.

SUS has 10 questions which measures user satisfaction and the analysis is based on the participants' input. Figure 3 gives an idea about the SUS score ratings corresponding to participants and average SUS mean score for RWA. For novice users 1, 3, and 4, their average rate to perform the tasks (Identification Steps) is 43, 35, and 47.5 out of 100, respectively. Therefore, the satisfaction level for both participants is poor.

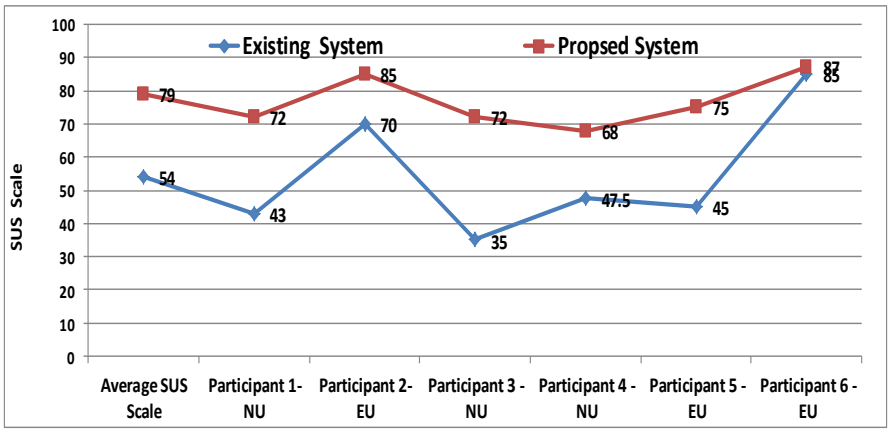


Figure 3: User Satisfaction results

For expert users 2, 5, and 6, their average satisfaction rate to perform the tasks (Identification Steps) is 70, 45, and 85 out of 100, respectively. For participants 2 and 6, satisfaction rate is marginally high (Good) and Excellent which is acceptable, respectively. The overall user satisfaction survey result for the existing RWA is 54, which shows low marginal.

From the above usability testing results, UIT in RWA has efficiency, effectiveness, and satisfaction problems.

4. Enhanced UIT for RWA

Based on the rationale behind the finding and discussion, the enhanced UIT for RWA considers usability features such as effectiveness, efficiency, user satisfaction, learnability, and memorability. Moreover, from security point of view, attention and memorability are also considered.

Figure 4 shows the high level design for the enhanced UIT for RWA.

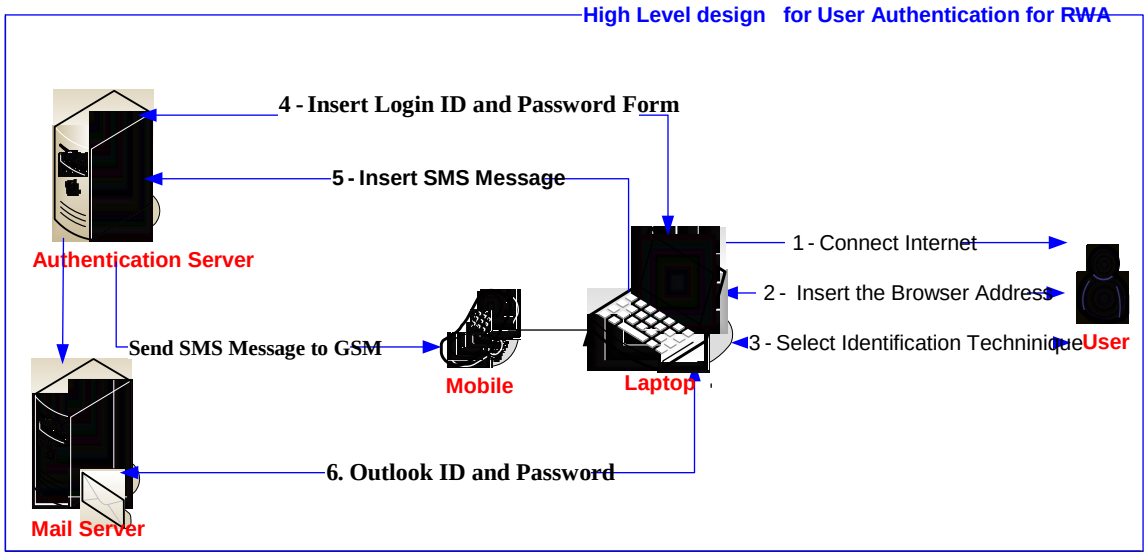


Figure 4: High Level design for UIT for RWA

5. Conclusion

Most of the time users are using RWA when they are offsite. Thus, the overall user satisfaction survey result for the existing UIT is low. Moreover, RWA is not effective with the current user identification steps. The rationale behind this is that users fail to achieve their goals and complete their tasks. The goal must be achieved within the acceptable time and effort but according to this research, the existing UIT is not effective. Therefore, it is possible to say the existing UIT has affected the usability of RWA.

The major finding of this research shows that most RWA users UIT is not easy to use and understand to follow. According to the survey result, for most users, UIT is not interesting and engaging; rather it is complex. Because of this reason, the UIT frustrates users. To minimize this frustration, RWA users start to write a note with a piece of paper to remember their secret such as username and password for ECAS and application, domain, and delegations. Also users discuss about their secret with others to login and access their email easily. All these actions expose the system to intruders. Moreover, they are not only exposed for stealing data but also as a whole the organization's UIT is vulnerable to attacks. Therefore, the new user identification technique of RWA makes them unhappy.

Despite the above conclusion, this study has limitations and future research in terms of accessibility will need to be done. Moreover, if this study is extended to other organizations which are using RWA, the outcome would be more precise.

References

[1] Wikipedia, "Microsoft Remote web Workplace," http://en.wikipedia.org/wiki/Microsoft_Remote_Web_Workplace, last access on March 20, 2012.

[2] J. Nielson, "Introduction to Usability", http://www.useit.com/alertbox/20030825.html, last accessed on February 02, 2012.

[3] L. Sommerville, Software Engineering, 9th ed., Addison-Wesley, 2010, pp. 745.

[4] C. Braz, and J. Robert, "Security and Usability: The Case of the User Authentication Methods", in Proceedings of the 18th International Conference of the Association Francophone D'interaction Homme-Machine, IHM 2006, Montreal, Canada, Vol. 133, 2006, pp. 199-203.

[5] Wikipedia, "European Union," http://en.wikipedia.org/wiki/European_Union, last accessed on March 02, 2012.

[6] CNN World, "European Union under cyber-attack as major summit begins," http://articles.cnn.com/2011-03-24/world/eu.cyberattack_1_cyber-attack-eu-summits-eu-administration?_s=PM:WORLD, last accessed on January 02, 2012

[7] European Commission Information System Security Policy, "C(2006) 3602", Brussels, 2011, pp. 18

[8] J. Nielsen, "Usability Engineering", Academic Press, Boston, 1993, pp. 26.

[9] ISO 9241-11, "Ergonomic Requirements for Office Work with Visual Display Terminals", 1999.

[10] B. Schneider, "Sensible Authentication," ACM Queue 1, 2004.

[11] J. R. Lewis, "Sample Sizes for Usability Studies: Additional Considerations," Human Factors, Vol. 36(2), 1994, pp. 368-378

[12] B. Aaron, K. Philip, and M. James, "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale," Journal of Usability Studies, Vol. 4, Issue 3, 2009, pp. 114-123.

Quantification of Software Requirements Specification and Software Quality: Quantitative Approach

Wesen Tegegne
Commercial Bank of Ethiopia, Addis Ababa, Ethiopia
wesenteg@gmail.com

Mesfin Belachew
Ministry of Communication and IT, Addis Ababa, Ethiopia
mesfinbelachew@gmail.com

Mesfin Kifle
Department of Computer Science, Addis Ababa University, Ethiopia
kiflemestir95@gmail.com

Abstract

One of the major problems of local software development firms and organizations is absence and unavailability of developed quantitative quality factors measurement technique that could assist to rate Software Requirements Specification (SRS) and software quality. In this paper a quantitative quality measurement, Quality Quantification Technique (QQT), is proposed by adapting and adopting from available measurements to quantify quality of SRS at requirement stage of System Development Life Cycle (SDLC) and software product. It provides 14 steps.

QQT uses propositional parameter for measuring quality attributes using linguistic variables. Due to human decision-making where the decisional criteria are not clear-cut, but blurred, it attempts to quantify the quality using fuzzy logic that can be defuzzified to get the final quality in crisp value. In QQT, application of fuzzy logic technique in measuring quality helps to resolve vagueness that could exist to some extent. The technique used in quantification of software quality is adopted and adapted to establish the base for SRS quality measurement. In addition to adopting and adapting the available software metrics fuzzy value, we tried to incorporate self-understanding and experience of inferring quality attributes. In order to evaluate the proposed QQT, a case study is carried out to measure SRS and software quality.

Keywords: SRS Quality, Software Quality, Fuzzy Logic, Defuzzified, QQT

1. Introduction

Looking at software engineering from a historical perspective, the 1990s and beyond could be viewed as the quality and efficiency era. In this era, demand for quality is further intensified by the ever increasing dependence of society on software. Quality has been brought to the center of the software development process. Software development has to be more efficient and the quality level of the delivered products has to be high to meet requirements and to be successful. These factors will continue to affect software engineering for many years to come [1].

The need for quality measurement is increased by the rising reliance of society on software. This is where the demand for measuring quality becomes apparent, be it qualitative or quantitative. In qualitative measurement, quality is measured in a more subjective manner whereas in quantitative measurement, the focus is in establishing metrics that could appropriately measure the degree of quality. In this paper, a quantitative technique that uses fuzzy logic to measure quality of SRS and software is applied.

2. Background

SRS Quality: Writing a good requirements document is difficult in the eyes of most developers. It delays getting on with the coding and it is thus considered a waste of time [2]. If we can better understand how to recognize and measure quality in SRS, we will be better equipped to detect errors in the SRS. SRS errors need to be detected during the requirements phase, or the cost to repair them will grow significantly [3]. As Davis *et al*. [3] quoted from Daly, Bohem *et al*. and Fagan they provide conclusive evidence that the later in the lifecycle an error is detected and repaired, the more it will cost. The ratio between detecting and repairing an error during requirements versus maintenance phases is 200:1

Software Quality: According to Herb [4], even though many successful software products and systems exist in the world today, an overall lack of attention to quality has also led to many problematic systems that do not work right as well as to many software projects that are late, over budget, or canceled "… so many systems are delivered that are not used, or not liked … we all believe we have done a good job and are mortified if our work is not well received [5]". Identifying quality requirements that can be elicited, formalized, and further evaluated in each phase of full software product lifecycle becomes a crucial task in the process of building a high quality software product [6].

According to Challa *et al*. [7], fuzzy logic provides an easier way to infer definite conclusions from highly imprecise, vague, and ambiguous information when compared with classical logic and relies on the experience of the operator rather than the technical understanding of the subject.

Applying fuzzy logic for the quantification of SRS and software quality is the basic concept behind fuzzy logic. According to Changli [8], fuzzy logic is inspired by the human processes for quality measurement where the decisional criteria are not clear-cut, but blurred, and it is difficult to find objective measurement to make the decisions more precise and clear. In this quantitative quality measurement, fuzzy logic is used. According Zadeh [9], fuzzy logic is a precise logic of impression and approximate reasoning.

3. Related Work

For some quality factors of SRS, Davis *et al*. [3] tried to give objective measurement, recommended weights of importance and reference values for attributes. They suggested that some of SRS qualities appear to be essential for all applications and gave weights of 1. Others seem less important in general and gave lower weights. The actual weights for all the attributes must be assigned by each project to be meaningful. They proposed SRS quality equation, $Q = \Sigma W_i Q_i / \Sigma W$ where W is weight of quality factors and i is set of SRS quality attributes such as {complete, correct, etc.}. Some of the limitations are most quality attributes have no objective measurement and weight of importance and reference values for attributes are not suggested.

Bawane *et al*. [10] proposed a quantitative model to establish the quality requirements expected by various stakeholders and to incorporate these requirements in the product under development. The quantitative quality model takes a set of quality requirements as input for the development of a software application. Some of the limitations are that it is model specific and does not apply methods to solve vagueness and imprecision which exist during subjective judgments of relative priorities of quality characteristics by stakeholders.

The authors in [11] identified software quality factors for the telecommunication industry in Malaysia. The quality characteristics and sub characteristics of a product were isolated from literature review to reflect what is currently thought to be important. Then they are converted into meaningful questions in a questionnaire and sent to stakeholders (users, developers, and managers) who expressed their quality opinions. Then, they gathered the questionnaires and consolidated the opinions from the correspondents. Some of the limitations are that it is model specific and there exist subjective judgments in determining relative priorities of quality characteristics by stakeholders.

Challa *et al*. [7] attempted to quantify software quality factors with the help of quality factors stated in ISO 9126 model. The software quality model is sub divided into perspectives, characteristics, sub characteristics, and metrics. Three perspectives (users, developers, and managers) are used to express their quality opinions. The project manager's perspective has been separately added to the model. Each perspective is sub divided into various characteristics. The sub characteristics included in the project manager's perspective are: cycle time, cost, and schedule pressure.

Every characteristic is further sub divided into sub characteristics and every sub characteristic is further sub divided into metrics. At these levels, parameter is associated with a corresponding rating and weight. Fuzzy weighted average of the metrics is taken to evaluate the rating of the sub characteristic. Then fuzzy weighted average of the characteristics is taken

to get the rating of the perspective. Fuzzy weighted average of different perspectives is taken to get the final software quality in terms of a fuzzy set. Centroid formula is then employed on this triangular fuzzy set to calculate the final software quality [10]. Some of the limitations are that it does not provide a technique to include other stakeholders for quality measurement and it is not generic (based on ISO 9126 model).

Our proposed solution to quantify quality is a remedy to the limitations in [3, 7, 10]. In our proposed solution, different stakeholders' views are included to express their quality opinions. Quantification techniques include SRS quality (not only for software quality), it is not to be quality model specific (the model is a general one). In SRS quality measurements, fuzzy logic is applied to solve vagueness and imprecision that could exist during subjective judgments. A general formula is given for quantification of net quality and the quantification process is general (not specific to software quality only). Project managers' characters (cost, time, and schedule pressure) is added separately to measure quality of SRS. We also suggest measurement techniques for attributes that have no objective measurement.

4. The Proposed Solution

All the steps required to follow in quality quantification process are listed in Table 1 and can be used as a checklist. QQT has three main stages: quality analysis, quality design, and quality measurement. In turn, each of these stages has further sub steps. The main objective of quality analysis is to identify and include the view points of different stakeholders and their quality requirements based on their main interest or the definition of quality they favor. In managing a software project, stakeholders could have different views or expectation in measuring quality attributes.

Quality design is the stage that is going to prepare metric questions for stakeholder quality characteristics and preparation of format to be filled out by stakeholder information. QQT will use fuzzy logic. The weighted average technique of triangular fuzzy sets is used. Once all the stakeholders' quality requirements for SRS or software are identified and gathered for each group, each attribute will be quantified by individual metric. Linguistic variables for the ratings and weights of the fuzzy sets for the quality metrics defined as VL (Very Low), L (Low), M (Medium), H (High), and VH (Very High).

Table 1: The Fourteen QQT steps

1	Identify stakeholders	Quality Analysis
2	Classify stakeholders	
3	Identify stakeholders' quality requirement	
4	Specify metrics for each quality characteristics	Quality Design
5	Define fuzzy value for quality attribute	
6	Determine ratings of metrics for quality attribute	
7	Determine weight of metrics for quality attribute	
8	Define the weights of the fuzzy sets for quality metrics	Quality Measurement
9	Define the ratings of the fuzzy sets for quality attribute	
10	Compute average fuzzy ratings of quality metrics	
11	Compute weighted average of quality	
12	Compute average weight of quality	
13	Compute net quality	
14	Defuzzification	

In order to measure the quality factors of SRS and software, triangular fuzzy set is used for simplicity. For every metric, there will be a corresponding rating and weight. The rating and the weight of any metric is fuzzified into triangular fuzzy sets. The required inputs of the ratings of the metrics and weights at

different levels will be obtained from stakeholders via questionnaire or in an interactive discussion with stakeholders. The weighted average technique is used to calculate the final net quality. The pictorial representation of weighted average of quality process in shown in Figure 1. In the process of calculation, levels could be omitted if a quality factor does not have sub characteristics. Moreover, the levels can be extended if the classification hierarchy is increased.

Let classification be the name given to the levels in quantification process. These are quality (level 0), perspectives (level 1), character (begins at level 2), sub characteristic (ends at level N-1), and metrics (level N) classification names. The fuzzy rating of the Level (N-M)+1 classification is obtained by a weighted average of Level N- M classification affecting it. It can be written as a formula.

Rating of Level N-M classification = $r_1 \times w_1 + r_2 \times w_2 + \ldots r_n \times w_n = \sum r_i \times w_i$

where r is fuzzy average rate of Level (N-M)+1 classification, w is fuzzy average weight of the Level (N-M)+1 classification, and i belongs to the set of Level (N-M) +1 classification affecting Level (N-M) classification, M = 1, …, N.

The process of converting a real time problem into fuzzy set is called fuzzification. The last step in QQT is defuzzification which is the process of converting the fuzzy sets obtained during fuzzification process into crisp or real time data. The fuzzy ratings of the quality with respect to different perspectives and the net quality are defuzzified. The centroid method is adopted to defuzzify the triangular fuzzy sets [7].

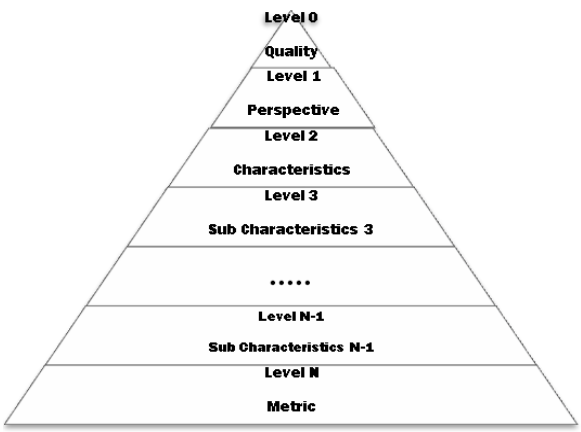


Figure 1: Weighted average calculation process

Centroid method $Z^* = \frac{\int \mu(z) z dz}{\int \mu(z) dz}$

Here z* is the defuzzified crisp value, z is the value on the X axis, and μ(z) is the membership function.

The inference table for SRS and software quality is shown in Table 2 which is adopted from [7]. The quantified quality level lies in the range of 0 to 1. Calculated quality value that is closer to zero indicates the poorest quality level and a value closer to one shows excellent quality.

Average quality level might be acceptable for non-critical systems if it could not compromise the major functionalities of the software. However, it is highly recommend having at least good quality level for non-critical systems. If software quality is below average (poor or very poor), it is unacceptable.

Table 2: Inference on Quality

Overall Quality Calculated	*Inference on Quality*
More than 0.65	Very Good
Between 0.5 and 0.65	Good
Between 0.35 and 0.5	Average
Between 0.25 and 0.35	Poor
Less than 0.25	Very Poor

Software quality always depends on the context in which software operates. Good quality level might be dangerous depending on the environment in which the software operates. A software application that is critical to the operations of the organization will require very good quality level. A higher level of quality even closer to one might be important.

SRS is the success of all software projects and it is the basis for all subsequent stages of SDLC. Very good quality level for SRS could lead to excellent software. If the quality level of SRS is close to one, then the cost in detecting and repairing an error in the later life cycle of software development will decrease significantly. Hence, SRS quality is acceptable if its quality level is at least very good and a level of quality is preferable if it is closer to one.

5. Case Study

Company A has prepared SRS and this is to evaluate SRS quality on the basis of the selected quality attribute from Davis *et al*. [3], Wiegers [6], and IEEE [12]. Besides, project manager's perspective quality factors (cycle time, cost, and schedule pressure) are added to the measurement. Fourteen QQT steps are used as check list for measurement. Stakeholders are classified based on the definition of the quality they favor and their main interest. They are builders, measurers, and managers. It is agreed and negotiated with compromise that weight of importance to builders perspectives are very high, managers perspectives are high, and measurers perspectives are average in the SRS quality.

Table 3: Fuzzy triangular number for the weights

Importance of Criteria	*Fuzzy Weights*
Very Low	(0.0, 0.0, 0.25)
Low	(0.0, 0.25, 0.5)
Medium	(0.25, 0.5, 0.75)
High	(0.50, 0.75, 1.0)
Very High	(0.75, 1.0, 1.0)

Table 4: Fuzzy triangular number for the ratings

Ratings of Criteria	*Fuzzy Ratings*
Very Low	(0.0, 0.1, 0.3)
Low	(0.1, 0.3, 0.5)
Medium	(0.3, 0.5, 0.7)
High	(0.5, 0.7, 0.9)
Very High	(0.7, 0.9, 1.0)

The characteristics are identified from literature review and discussion among stakeholders to reflect what is currently thought important to SRS. Stakeholders filled out different questionnaire. Three builders, two measurers, and two managers filled out questionnaire of input for SRS measurement. The fuzzy value of weights and ratings are shown in Tables 3 and 4, respectively.

Now the fuzzy rating of the overall SRS quality (net quality) is calculated using

r Net Quality = r Builder's×w Builder's+ r Measure's× w Measure's+ r Project Manager's× w Project Manager's. r Net Quality = (.33,.79,1)

Centroid method $Z^* = \frac{\int \mu(z) z dz}{\int \mu(z) dz}$

So the crisp value of the SRS net quality is calculated as 0.70.

Table 5: Fuzzy ratings for net SRS quality and quality with respect to different perspective

Net SRS Quality	*Quality*	*Net Weighting*	*Net Rating*	*Quality Attribute*	*Average Fuzzy Weight*	*Average Fuzzy Rating*
(0.33,0.79,1)	Builder	(0.75, 1.0, 1.0)	(.44, .79, 1)	Achievable	(0.75,1,1)	(0.63, 0.83,0.97)
				Annotated	(0.33,0.58,0.83)	(0.7,0.9,1)
				Complete	(0.25,0.5,0.75)	(0.7,0.9,1)
				Consistent	(0.75,1,1)	(0.7,0.9,1)
				Correct	(0.67,0.92,1)	(0.7,0.9,1)
				Modifiable	(0.33,0.58,0.83)	(0.23,0.43,0.63)
				Necessary	(0.42,0.67,0.92)	(0.5,0.7,0.9)
				Precise	(0.58,0.83,0.92)	(0.3,0.5,0.7)
				Reusable	(0.17,0.33,0.58)	(0.43,0.63,0.8)
				Traceable	(0.75,1,1)	(0.3,0.5,0.7)
				Unambiguous	(0.5,0.75,0.83)	(0.1,0.3,0.5)
				Understandable	(0.67,0.92,1)	(0.5,0.7,0.9)
	Measure	(0.25, 0.5, 0.75)	(.44, .79, 1)	Achievable	(0.63,0.88,1)	(0.5,0.7,0.85)
				Annotated	(0.13,0.38,0.38)	(1,0.3,0.25)

Net SRS Quality	*Quality*	*Net Weighting*	*Net Rating*	*Quality Attribute*	*Average Fuzzy Weight*	*Average Fuzzy Rating*
				Complete	(0.25,0.5,0.75)	(0.7,0.9,1)
				Consistent	(0.25,0.5,0.75)	(0.5,0.7,0.85)
				Correct	(0.63,0.88,1)	(0.6,0.8,0.95)
				Modifiable	(0,0.25,0.25)	(0.65,0.4,0.35)
				Necessary	(0.38,0.63,0.88)	(0.4,0.6,0.8)
				Precise	(0.5,0.75,0.88)	(0.4,0.6,0.8)
				Reusable	(0,0.13,0.38)	(0.65,0.4,0.6)
				Traceable	(0.25,0.5,0.75)	(0.6,0.8,0.95)
				Unambiguous	(0.38,0.63,0.75)	(0.7,0.9,1)
				Understandable	(0.63,0.88,1)	(0.7,0.9,1)
	Manager	(0.50, 0.75, 1.0)	(.35, .68, 1)	Cycle Time	(0.38,0.63,0.88)	(0.5,0.7,0.9)
				Cost	(0.13,0.38,0.25)	(0.3,0.5,0.7)
				Schedule Pressure	(0.5,0.75,1)	(0.7,0.9,1)

Based on inference quality Table 5, and its SRS quality interpretation, since the net SRS calculated value is 0.70, we can say that the SRS has very good quality and it is acceptable. Due to space limitation, this paper does not elaborate each of the 14 steps of QQT and the case study carried out to measure software quality. For detailed information, please refer to [13].

Table 6: Crisp value of perspectives' SRS quality

Name of Perspective	*Perspective quality*	*Crisp value*
Builders	(.44, .79, 1)	0.74
Measurers	(.44, .79, 1)	0.74
Managers	(.35, .68, 1)	0.67
*Total Quality (*net quality)	(.33, .79, 1)	0.70

6. Conclusion

In this paper, a quantitative SRS and Software quality measurement technique, QQT, is proposed. Software development firms and organizations or stakeholders can utilize QQT to measure quality status of the SRS at requirement stage of SDLC and the end result of software quality quantitatively. QQT is also useful for developers, users, and project managers to measure quality from each perspective and net quality. The proposed technique is easy and understandable. It uses propositional parameter for measuring quality attributes using the linguistic variables very low, low, medium, high, and very high.

These linguistic variables are fuzzy because the boundary of variables is not clear-cut. It is difficult to find objective to make the decisions more precise and clear. QQT uses fuzzy logic due to human decision making where the decisional criteria are not clear-cut, but blurred. Fuzzy logic uses to analyze approximate data to precise solutions. In QQT, the quality parameters are quantified in terms of fuzzy set and finally converted to crisp or numeric value.

The fuzzy value or criteria to metrics are not without dispute. The approach uses experience and opinion of a person involved in the process of measurement. Different persons can set fuzzy values for fuzzy metrics differently. The other limitation is also that quality classification among perspectives by different persons may not be the same.

References

[1] Stephen Kan H., "Metrics and Models in Software Quality Engineering", Addison Wesley, 2nd edition, 2002.

[2] Tom Glib and Lindesy Brodie, "What's Fundamentally Wrong? Improving Our Approach Towards Capturing Value in Requirements Specification", http://www.

requirementsnetwork.com/system/files/Whats Fundamentally Wrong.pdf, last accessed on August 06, 2011.

[3] Alan Davis, Scott Overmyer, Kathleen Jordan, Joseph Caruso, Fatma Dandashi, Anhtum Dinh, Ledeboer Kincaid, Gary, Glen, Patricia Reynolds, Pradip Sitaram, Anh Ta, and Mary Theofano, "Identifying and Measuring Quality in a Software Requirements Specification", First International Software Metrics Symposium, Proceedings, 1993.

[4] Krasner, Herb. "Using the Cost of Quality Approach for Software", Crosstalk Journal of Defense Software Engineering, 1998.

[5] Isabel Evans, "Achieving Software Quality through Teamwork", Norwood, 2004.

[6] Wiegers, Karl E. "Software Requirements", Microsoft Press, 2003.

[7] Jagat Sesh Challa, Arindam Paul, Yogesh Dada, Venkatesh Nerella, Praveen Ranjan Srivastava, and Ajit Pratap Singh, "Integrated Software Quality Evaluation: A Fuzzy Multi-Criteria Approach", Journal of Information Processing Systems, Vol. 7, No. 3, 2011.

[8] Sun Changli, "Software Documents Quality Measurement - A Fuzzy Approach", Information Theory and Information Security (ICITIS), IEEE International Conference, 2010.

[9] Loti A. Zadeh, "Is there a need for Fuzzy Logic?", Fuzzy Information Processing Society, NAFIPS.

[10] Bawane, Neelam and Srikrishna C. V, "A Novel Method for Quantitative Assessment of Software Quality", International Journal of Computer Science and Security, Volume 3: Issue 6, 2009.

[11] Nor Fazlina, Iryani Abdul Hamid, and Mohamad Khatim Hasan, "Identifying Software Quality Factors for Telecommunication Industry in Malaysia", International Conference on Electrical Engineering and Informatics, 17-19 July 2011.

[12] IEEE Standard Board, IEEE Recommended Practice for Software Requirements Specifications, IEEE, 1998.

[13] Wesen Tegegne, "Software Requirements Specification and Software Quality Measurement: Quantitative Approach", Unpublished Master's Thesis, HiLCoE School of Computer Science and Technology, 2012.

Semantic Knowledge Oriented Electronic Health Record Integration System

Winta Tesfaye
Tulane International Ethiopia, Addis Ababa, Ethiopia
wintes@gmail.com

Fekade Getahun
Department of Computer Science, Addis Ababa University, Ethiopia
fekadegetahun@gmail.com

Abstract

Electronic Health Record system aims to enhance the health care service. However, as the amount of information being stored grows, it not only makes the patient data duplication prone but also affects the quality of the health care service. This is so as a patient's medical history is distributed across several facilities and is not synced.

Duplicate records can happen because of many reasons - misspellings, typos, and transpositions. The high demand of data sharing especially in the domain of the medical world is highly related with providing a better health care and that possibly leads to less loss of life. This calls for data integration. In a country like Ethiopia, a nationwide unique patient identifier does not exist. Rather the concept of master patient index restricted to a heath facility is used. Hence, a patient identification number generating system is one of the first steps towards a unified view of a patient's medical history. In this paper, we have proposed an algorithm that helps us produce a unique patient identification number. Going further, when trying to merge or integrate heterogeneous and distributed information, there is a need to compute the similarity between patient profile and medical history. We identify similar patient through Edit Distance based similarity method and corresponding medical history matching using vector based similarity method - cosine with the help of dedicated knowledge base. In the last stage, we merge the clinical data of the patient as a result of these combined matchers.

Keywords: Semantic; Knowledge Oriented; Integration; Electronic Health Record System; Schema Matching; Patient Identification

1. Introduction

In recent years, there have been a significant amount of research that have focused on integration of data that are disparate and fragmented across different sources. It is the rapid growth in the amount of information that is being stored in such data sources that made the concept of data integration techniques a pressing issue and lead to this study.

The high demand of data sharing, especially in the domain of the medical world, is highly related with providing a better health care and that possibly leads to less loss of life. Healthcare organizations demand to unite and share the very sensitive patient data so as to increase the quality of care which in turn reduces treatment costs. Due to the abundance of great volumes of health records, there is a need to have a unified view of a patient's entire medical history. This need and the increasing awareness of added value through integration of medical information and to improve patient care is what has called for improvement in ways of integration.

2. Background

The first known medical record was developed in the fifth century B.C. with two basic goals [1]:

- a medical record should accurately reflect the course of disease, and
- a medical record should indicate the probable causes of disease.

These goals are still appropriate, but electronic health record systems can also provide additional functionality, such as interactive alerts to clinicians, interactive flow sheets, and tailored order sets, all of which cannot be done with paper-based systems.

The first EHRs (Electronic Health Record Systems) began to appear in the 1960s. In [2], it is reported that by 1965, at least 73 hospitals and

clinical information projects and 28 projects for storage and retrieval of medical documents and other clinically-relevant information were underway.

Motivation Scenario

Assume that Patient P1 goes to a health facility, HF1, to get health care service. In HF1, s/he gets a unique patient ID upon the first visit and registration process. The patient receives a medical diagnosis (for instance, Asthmatic) with or without a confirmed laboratory result and the medical practitioner may prescribe medicine and provide consultation geared to the patient case. Even though health record can be kept using paper or an electronic counterpart, this paper focuses on the electronic version.

After a few months, the same patient P1 goes to a different health facility, HF2. In HF2, P1 is given a different patient ID and is considered as someone who goes to a health facility for the first time or with a different prior set of primary disease diagnosis, as patient ID is restricted to one Health Facility. As a result of this, patients' data are scattered across facilities each having possibly different repository structure. Because of the lack of past patient history that accompanies each patient, in HF2, P1 was diagnosed for early stage of high blood pressure and given Aspirin wrongly.

This scenario demonstrated the need to

- identify patients independent of the health facility,
- match patients demographics and/or patient summary information based on the actual content, and
- integrate patients' records distributed in several locations.

3. The Proposed Solution

The aim of this work is to provide medical practitioners with a unified view of patient's medical history regardless of which facility the patient has received the service before. The high level architecture of our approach is shown in Figure 1. One of the main results of this work is generating nationwide patient ID. Unique patient ID is crucial to facilitate the integration process. Merging of patient information from different sources is assisted with element name mapper that identifies the elements/attributes that describe the same fact. The mapper uses our dedicated knowledge base that contains list of related concepts. Once the elements are mapped, we compute the content similarity using a text similarity module which is assisted with vector space mode. We have considered similarity matchers and their combined result to identify a certain patient across different databases. Then the clinical data of the patient shall be merged to be able to be viewed as one by a physician.

In order to briefly go through the details of the above overview, we have classified the report into three phases: the Patient Identification, Schema Matching, and Merger.

Among the different demographic information of a patient, the use of static and less frequently changing attributes is crucial as input to the ID generator. Thus, we propose the use of Birth-Place, Date-of-birth, Sex, and Blood type to uniquely identify each patient.

The algorithm that produces the output is shown in Algorithm 1.It returns a 14 digit value resulting from the concatenation of representing a Patient Uniquely as shown in Line 12.

We used edit distance to compare the similarity of the element names and content of the schemas. Edit Distance [9] is a string metric for measuring the amount of difference between two strings. The Edit distance between two strings is defined as the minimum cost sequence of operations to transform one string to the other. The basic edit script operations are: I (insert), D (delete), and R (Replace).

Example: Given two strings S1: 'Error' and S2: 'Eror', an insertion of r in S2 or removal of the double r from S1 makes the string similar, thus EditDistance(Error, Eror) = 1.

Similarly the EditDistance (great, grate) = 2

The higher the edit distance, the smaller is the similarity between the two strings. Thus, the edit distance similarity between two strings S1 and S2 is defined as

$$simedit(S_1,S_2) = \frac{1}{(1 + EditDistance(S_1,\ S_2))}$$

Example:

- $simedit(Error, Eror) = \frac{1}{(1 + EditDistance(Error, Eror))} = 0.5$
- $simedit(\text{great}, \text{grate}) = \frac{1}{(1 + EditDistance(\text{great}, \text{grate}))} = 0.33$

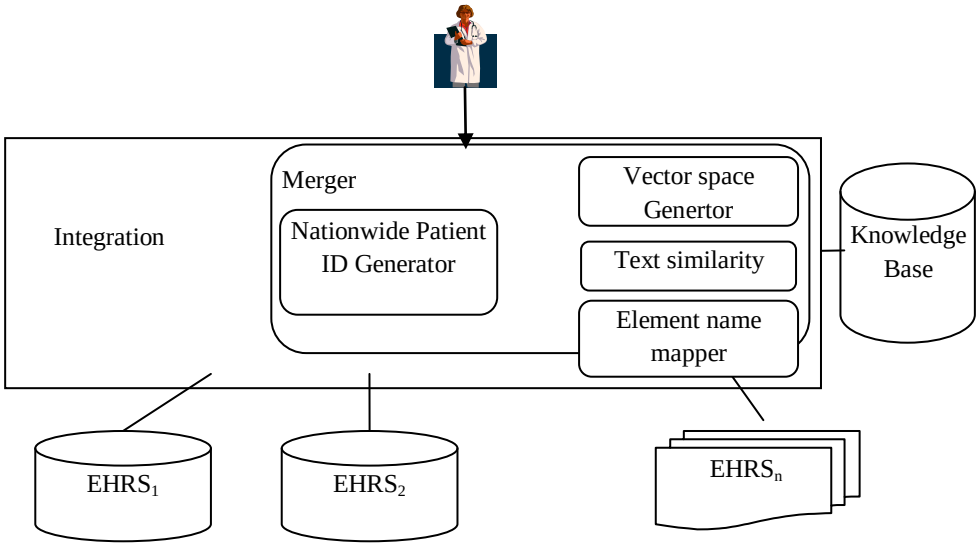


Figure 1: Overview of architecture of knowledge base integration of EHRS

Algorithm 1: Nationwide Patient ID Generator

Line #	
	Input:
1.	BirthPlace: string
2.	Dateofbirth: date
3.	Sex: char //sex as M or F
4.	B: char // blood type
	Variables:
5.	SSS: string, PPPPP, YYYMD, SSS: char [5]
6.	KB: LocalityKnowledgeBase
	Output:
7.	PatientID: Char[14] // PPPPP YYYMD B SSS
	Begin
8.	PPPPP = GetLocalityCode(BirthPlace, KB)
9.	YYYMD = GetDateBirthSex(Dateofbirth, Sex)
10.	B: stands for Blood Type
11.	SSS = GetNextSerialNo(PPPPP, DDDDD) //returns is a sequential number
12.	Return PPPPP + DDDDD + B + SSS
	End

The algorithm accepts the patient profile as input and visualizes to the user the relevant medical history. It first extracts the nationwide patient ID of the input using the function *ExtractNationWidePatientID*. Then it looks for the medical history of the patient with the given nationwide patient ID using the function *GetPatientMedicalHistorywithNationWidePatientID*. If the result is successful, merging comes down to applying conflict resolution also called merging rule presented here as function *ApplyConflictSolvingRules*. Otherwise, we look for list of patients profile very similar to the input profile. The similarity is computed using edit distance combined with vector based similarity – cosine. Notice that two profiles are similar if their similarity value is more than the threshold provided

by the user and returns the medical history of similar patients.

4. Experimentation (Prototype)

The prototype was developed based on the requirements at each phase of this work. The development tools used for the Semantic Knowledge Oriented Electronic Health Record Integration System are as follows:

Microsoft visual Studio 2005 for IDE Integrated development environment is used to parse through the database records as input and we used C# programming language. We used Microsoft SQL Server 2005 to keep our mediated schema that is used to show a unified view for the end user. Figure 2 shows the user interface used to register patient information which could be done in any of the health care facilities.

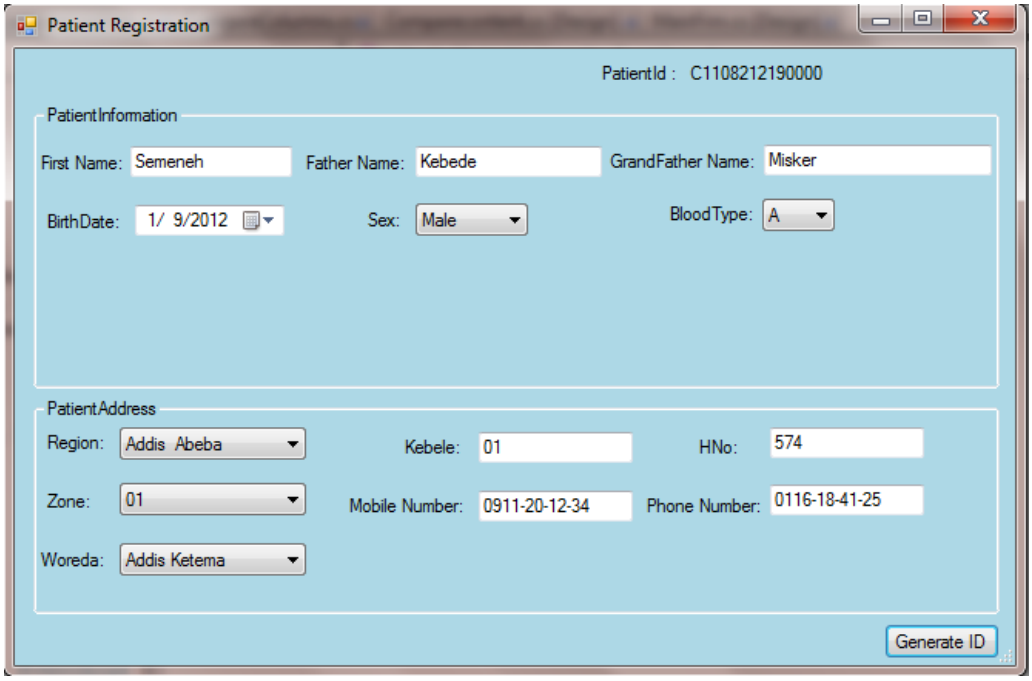


Figure 1: Sample Patient Profile with unique identification number

5. Related Work

5.1 Patient Identification number

In the United Kingdom [3], the National Health Service (NHS) proposed National Unique Patient Identifier to be used in England and Wales. They are in the middle of implementation by applying a unique patient identification number throughout the health facilities on each patient that comes for a service. Each NHS Number is made up of 10 digits shown in a 3-3-4 format, the first 9 being the identifier and the last digit being a check digit.

According to the recent RAND Corp. study, in the United States [4], it is indicated that the use of a unique patient identification number for every person in the United States would help in reducing medical errors, simplify the use of electronic medical records, increase overall efficiency, and protect patient privacy.

5.2 Schema Matching

CUPID is a hybrid match approach combining a name matcher with a structural one. Input schemas are converted to trees in which additional nodes are added to resolve the multiple relationships between a shared node and its parent nodes [5].

Similarity Flooding (SF) converts schemas (Relational, RDF, XML) into labeled graphs and uses fix-point computation to determine correspondences of 1:1 local and m:n global cardinality between corresponding nodes of the graphs [6, 7].

5.3 Similarity Computation

Given a hierarchical knowledge base KB, collection of concepts related with is kind of relationship, Wu & Palmer evaluated a conceptual similarity between pair of concepts in hierarchy-based structure using the distance between their most common ancestor [8]. The similarity measure is denoted as:

$$sim(C_1, C_1) = \frac{2 * depth\ (C)}{(len(C_1, C) + len(C, C_1) + 2 * depth(C))}$$

where:-

- C is the least common ancestor that subsumes both C_1 and C_2
- depth(C) is the depth of C from the root
- len(C_1, C_2) returns the path length between C_1 and C_2

Example: For instance, using the knowledge base shown in Figure 2, let us compute the semantic similarity between pair of concepts.

6. Conclusion

In this paper we have been able to provide approaches that provide unified patient identification number and merge/integrate heterogeneous patient data. Having an electronic health record system comes with a requirement of integration. We have seen the different experiences of developed countries such as the United States and the United Kingdom and learnt the pros and cons in order to recommend a separate identification number for patients. We have also seen the demand for integration of patient data. Studying through all the significant data matchers and merging data, we have provided a number of solutions. The contribution of this paper can be summarized as follows.

- an approach that generates a unified unique patient identification,
- a knowledge base schema matcher that results in the integration of the schema generated from the heterogeneous database schema of the electronic health record system,
- merging approach that puts together medical information of patients, and
- developed a prototype that contains the different algorithms developed in this work.

References

[1] J. H. van Bemmel and M. A. Musen, ed., Handbook of Medical Informatics, Springer, The Netherlands, 1997, pp. 99.

[2] Richard Dick, Elaine B. Steen, and Don Detmer, editors, The Computer Based Patient Record: An Essential Technology for Health Care, Institute of Medicine, National Academy Press, 1997, pp. 111.

[3] The National Patient Safety Agency, NHS Number to be used as the unique patient identifier by all NHS organizations in England and Wales, September 2008.

[4] RAND Corporation, Creating Unique Health ID Numbers Would Improve Health Care Quality, Efficiency, Study Claims, October 2008.

[5] Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm, Generic Schema Matching with Cupid, August 2001.

[6] Melnik, S., H. Garcia-Molina, and E. Rahm, "Similarity Flooding - A Versatile Graph Matching Algorithm", Proc. 18th Intl. Conf. Data Engineering (ICDE), San Jose CA, 2002.

[7] Amol Deshpande and Anthony Hunter, "Scalable Uncertainty Management" 4th International Conference, SUM 2010, Toulouse, France, September 27-29, 2010.

[8] Wu, Z. and M. S. Palmer, "Verb Semantics and Lexical Selection", In ACL., pp.133-138, 1994.

[9] Nick Koudas, Sunita Sarawagi, and Divesh Srivastava, "Record linkage: Similarity Measures and Algorithms", September 2006.

Integrated Approach for Software Project Management Methodology Selection

Wondwossen Elssa
Ministry of Communication and IT, Addis Ababa, Ethiopia
wondelsa@gmail.com

Fekade Getahun
Department of Computer Science, Addis Ababa University, Ethiopia
fekadegetahun@gmail.com

Abstract

The success rates of software projects are remaining too minimal over the years due to various technical and managerial issues. One aspect of the managerial issue includes selection and usage of appropriate software development and management models. In this work, an Integrated Framework for Software Project Development and Management Selection is proposed. The approach is comprehensive and inculcates the most common types of software development and project management models.

The framework has a knowledge base and inference engine components. The knowledge base component is organized from facts of completed software projects and extracted from the literature. The inference engine component has Mapper and Inference sub-components, responsible to extract appropriate software project management and software process models.

To demonstrate our approach, a prototype, Software Management Models Selection System (SMMoSS), is developed. The system accepts high level user requirements and project characteristics and returns list of highly relevant project management and software development models that fit to the need. The result of this work is believed to be a basis for software development works towards the realization of successful software projects.

Keywords: Software Development and Management Models; Software Development and Management Models Selection; Knowledge Base; Inference

1. Introduction

Most software development projects end-up far from the expectations of users. This can be supported by results such as in [2] that was obtained after evaluating software projects against the three traditional standard measurements of software success, namely time, cost, and quality.

Factors contributing for software failures could be both technical and managerial. Tesch *et al.* in [5] emphasize that the management aspects of a project are typically more critical to its success than the technical aspects. The software management aspect by itself may include various factors and identifying and selecting the right software development process and software management process models are few among them.

Software projects development and management models selection support system is characterized with diverse managerial and technical challenges. It is clear that providing a comprehensive and full-fledged solution for such an area is too vast and unimaginable. The coordinated effort of diversified professionals in the software engineering fields is vital. In this work, an attempt has been made to address one of the key aspects regarding the problem.

In this paper, an integrated software inference framework that could infer the appropriate model given system characteristics and overall user requirements (functional and non functional) of the new system is proposed.

2. Background

Unlike the rapid increment of software development projects that are globally conducted by different organizations [1], the success rates remain too minimal over the years [2]. The report in [2] shows incredible decrease in project success rates from 2006 to 2009; only 32% of all projects are considered successful (delivered on time, on budget, and with required features & functions); 44% were challenged (which were late, over budget, and/or

with less than the required features and functions), and 24% failed (which are cancelled prior to completion or delivered and never used). These numbers represent very low success rates of software projects. Factors contributing to the failures could be both technical and managerial. Accordingly, the solutions to overcome these failures should involve both; as only the technical approaches would not help to improve the high failure rates of software development projects.

It is reported that most of the blames for software failures are associated with human errors, such as management and lack of skilled manpower [3]. Moreover it is showed in [4] that effective project management (PM) is vital to the success of any software development project. The authors in [5] emphasized that the management aspects of a project are typically more critical to its success than the technical aspects. According to Baccarini in [7], project success is mainly influenced by project management success that focuses on the project management process, in particular on the successful accomplishment of the project with regards to cost, time, and quality. Also types of development models have relationship with cost, time, and quality and impact project success. The RAD development model, for instance, is characterized by short duration, low cost, and low quality. On the other hand, the Waterfall model is characterized by long duration, high cost, and high quality.

Understanding this and aiming to reduce the impact of software failures caused by the management issue, so far, a number of studies have been conducted.

The studies propose many new models, frameworks, or methodologies as a solution. Lessons are learnt from them and are incorporated in this paper. On the other hand, the limitations in those works are tried to be addressed.

3. Related Work

Aiming to reduce the impact of software failures caused by management, so far, a number of studies have been conducted. For Example, Rehman and Hussain [3] suggested a generic project management approach after comparing five important project management methodologies/frameworks with the PMBOK. Decisions in which a methodology/framework is chosen or used based on a single criterion can have serious impacts. In this work, features of the most common software management models are described thoroughly. However, it does not consider software development models.

According to Schwalbe [6], only all the nine project management knowledge areas (i.e., project integration, scope, time, cost, quality, human resource, communications, risk, and procurement management) and all the five process groups (i.e., initiating, planning, executing, monitoring and controlling, and closing) of PMBOK Management Model are described with regard to software projects and other management models did not get consideration.

Tesch *et al.* [5] presented IT project risk factors that pose threats to successful project implementation and describe assessment of these risk factors. Finally they conclude with suggested strategies for avoiding and or mitigating these risks. The limitation of this work is software development models do not get attention at all.

Hui in [4] outlined types, advantages and disadvantages, and applicable range of software IT projects on the basis of analyzing the relationship between the whole lifecycle. But it does not consider software development models one by one.

Recently, Hewagamage and Hewagamage in [1] proposed an integration of software project management and software development practices as redesigning framework for IT project management. However, their approach is towards only the waterfall development model and PMBOK management model. Thus, their approach is not complete enough to be used in a general context and hence doesn't support the need of MCIT-Ethiopia. Also they did not provide a tool to select the best appropriate software project management and development models.

4. The Proposed Approach

The proposed approach is an integrated software inference framework that could infer the appropriate model given system characteristics and overall user

requirements (functional and non functional) of the new system. These characteristics and overall user requirements would then be aligned with existing knowledge.

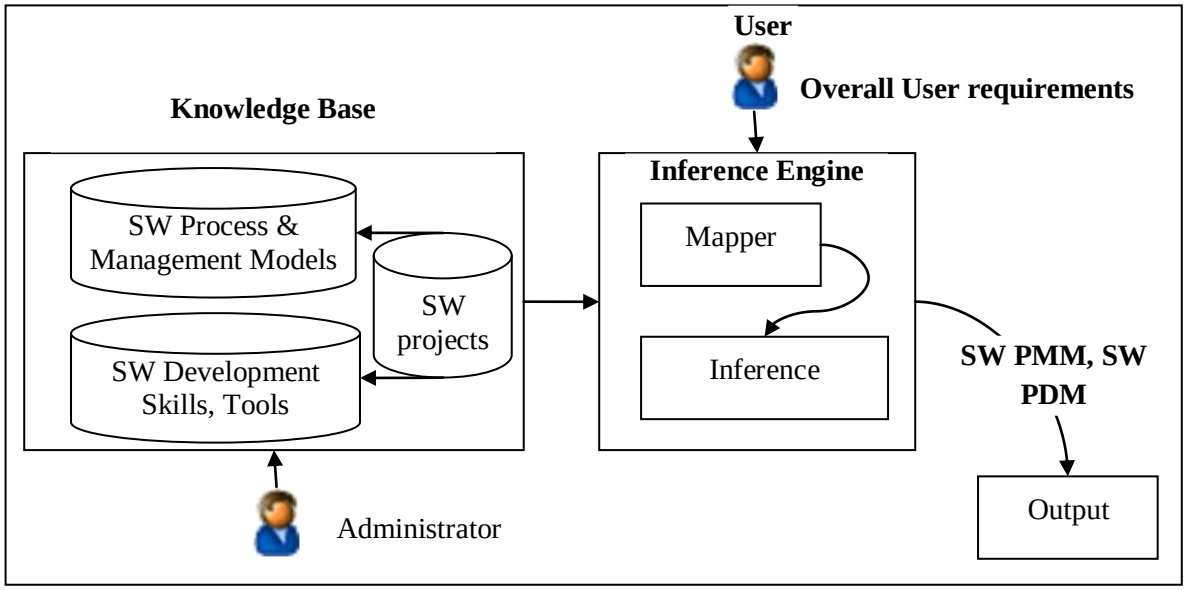


Figure 1: Integrated software inference framework

Figure 1 shows the two main components of the proposed framework and the interaction between them. The knowledge base is organized from knowledge of completed software projects and facts extracted from the literature. The knowledge base contains software characteristics and user requirements. The Inference Engine component has two sub-components (Mapper and Inference). The Mapper sub-component maps or aligns the overall user requirement and software characteristics to information stored in the knowledge base. The inference sub component is responsible to extract and propose appropriate software project management model and software process model.

The proposed solution has three basic steps and the third step has five intermediary steps:

Step 1: Software Project Knowledgebase (KB) representation

Step 2: Model Project Characteristics and User Requirements

Step 3: Inference

3.1: Mapping

3.2: Building Intermediate data structure

3.3: Counting 'Y' values

3.4: Sorting

3.5: Computing Confidence Level

Step 1: KB Representation

Software Project Knowledge Base (KB) can be expressed in terms of:

- Software project Characteristics (KBC),
- Overall User Requirements (KBR),
- Software Development Model (SDM), and
- Software Management Model (SMM).

Assuming the overall requirements of a user and the characteristics of a given software project are equally important criteria and have equal weight in selecting the right software development and software management methodologies, if the total number of characteristics is 'm_1' and the total number of requirements is 'm_2', then the characteristics and requirements of the knowledgebase KB would be:

KBC = $f(P_i, KBC_1, KBC_2, \ldots, KBC_{m1})$ and KBR = $f(P_i, KBR_1, KBR_2, \ldots, KBR_{m2})$

SDM = $f(P_i, KBSDM_1, KBSDM_2, \ldots, KBSDM_q)$ and SMM = $f(P_j, KBSMM_1, KBSMM_2, \ldots, KBSMM_k)$;

where f is a function and P_i and P_i are codified projects:

Step 2: Model Project Characteristics and User Requirements

A given project (P) is defined in terms of project characteristics (PC) and overall user requirements (PR). Let the total number of project characteristics be 'x' and the total number of overall user requirements is 'y'. Accordingly the project characteristics and overall user requirements of the project P would be:

PC = f(P, PC_1, PC_2, …, PC_x) and PR = f(P, PR_1, PR_2, …, PR_y)

Step 3: Inference

A structure to store the information of the project P will be created as shown in Table 1:

Table 1: Project descriptor of a given project P

PC_1	PC_2	PC_3	…	PC_X	PR_1	PR_2	PR_3	…	PR_Y

Step 3.1: Mapping

Match or map each project characteristics and overall user requirements of the given project P against project characteristics and overall user requirements in the Knowledge Base KB.

For each descriptor P_y of P, if a match descriptor KB_x is found in KB, write KB_x as a match and for others mark its match as NULL. In Table 2, the second row shows KB descriptors that correspond to the given project descriptors. Thus, Table 2 represents list of all projects from the KB that have at least one match descriptor of the project P.

Table 2: Sample mapping of project P and KB descriptors

PC_1	PC_2	PC_3	…	PC_X	PR_1	PR_2	PR_3	…	PR_Y
KBC_3	KBC_1			KBC_2		KBR_1	KBR_3		KBR_2

Step 3.2: Building Intermediate data structure

Modify the structure created in Step 3.1 so that it will be suitable to hold the necessary information of each project in the KB as shown in Table 3.

Table 3: Projects with matching KB desciptors

P#	PC_1	PC_2	PC_3	…	PC_X	PR_1	PR_2	PR_3	…	PR_Y	
P	KBC_d	KBC_b	KBC_k				KBR_1				No of 'Y'
1											
2											
…											
n											

Step 3.3 Counting 'Y' values

For each project in KB, count the total number of 'Y' values of the KB descriptors having matching project descriptor of P (i.e., result of Step 3.2) as shown in Table 4.

Table 4: Projects with matching KB descriptors and values

P#	PC_1	PC_2	PC_3	…	PC_X	PR_1	PR_2	PR_3	…	PR_Y	
P	KBC_d	KBC_b	KBC_k				KBR_1				No of 'Y'
1		Y					Y			Y	3
2	Y	Y			Y		Y	Y		Y	6
…											
n	Y	Y			Y		Y			Y	5

Step 3.4: Sorting

Sort the result of Step 3.3 in descending order on number of yes (last column) and take the project no, and number of match of the top K results as candidates. Notice that 'no of Y' is considered here as Z value.

Step 3.5: Computing Confidence Level

For each project identified in Step 3.4, identify the corresponding SDM and SMM from the KB. In addition, compute its confidence level using equation 1.

$$\left(\frac{Z}{(X+Y)} * 100\right)\% \quad (1)$$

where X is the number of Project characteristics and Y is the number of user requirements.

5. Prototype

SMMoSS implements *integrated approach for software project management methodology selection*. The system serves as a prototype to demonstrate the feasibility of the proposed solution. SMMoSS is a window based application developed using Visual Basic .NET 2010 Development Environment and using SQL Server 2008 R2 as back end.

In its development process, first, data such as software development models, software management models, common project characteristics, user overall requirements, and list of actively participated software development companies are identified. All these were identified or used by local software private companies. Then the data on various project characteristics with respect to software development and management models are collected and organized in a way to assist for decision. Finally the system is experimented with proper data to evaluate the feasibility of the selected models.

The functionalities are provided through Windows Explorer type interface that groups basic functionalities as data entry and knowledge inferring. Also the data entry group consists of setup sub group and developed software projects sub group. The setup sub group gives functionality to enter basic information on software development models and their characteristics, software management models and their characteristics, project characteristics, user overall requirements, and software project developing companies. The developed software projects sub group gives functionality to enter different information on completed software projects. Finally the knowledge inferring group provides functionality to accept information on user overall requirements and on characteristics of a new project and then analyzes these with the knowledge base and the science and finally yields the recommended software development and management models that best suit. Figures 2 and 3 show sample screenshots for SSMOSS Software Management & Development Models Recommendation Form.

In order to show the workability of the proposal made, two case studies are selected and experimented.

Case Study 1

An "Integrated HR and Payroll Handling Management System for Company X" from MCIT-Ethiopia is arbitrarily selected to verify the prototype. The project characteristics are: the application type is transaction-oriented, it is a medium sized project, it is a long duration system, the requirements are well clear, the requirements are stable, the project has clearly stated objectives, the assigned project manager is highly skilled and experienced (certified), and a strict requirement exists for formal approvals at designated milestones. The overall user requirements have facilities such as data storage, retrieving information, easy to use, and provision of role based security system.

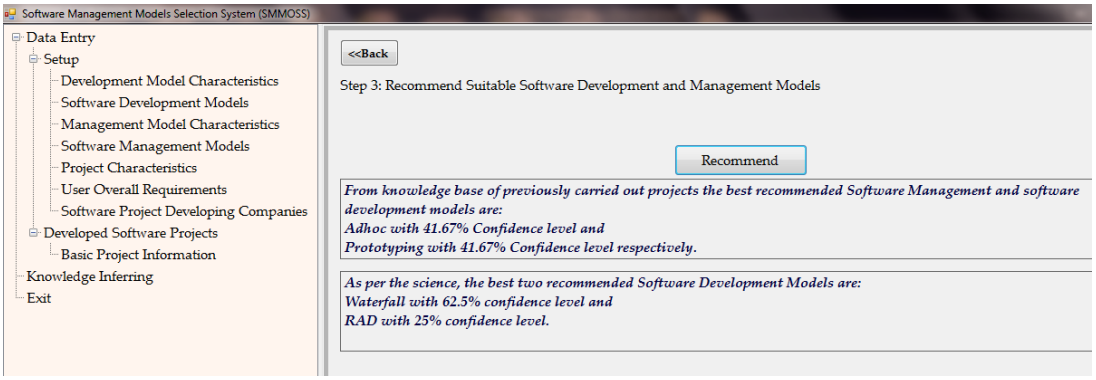


Figure 2: SSMOSS Software Management and Development Models Recommendation Form

Case Study 2

A "Static Web Page Development for Company Y" from MCIT-Ethiopia is arbitrarily selected to verify the prototype. The project characteristics are: the application type is Web informational, it is a small sized project, it is a short duration system, the requirements are not clear, the requirements are not stable, the project has no clearly stated objectives, the assigned project manager is not highly skilled and experienced (certified), and strict requirement exists for formal approvals at designated milestones. The overall user requirements are facilities such as retrieving information, local language support, local language calendar support, easy to use, and flexible to use.

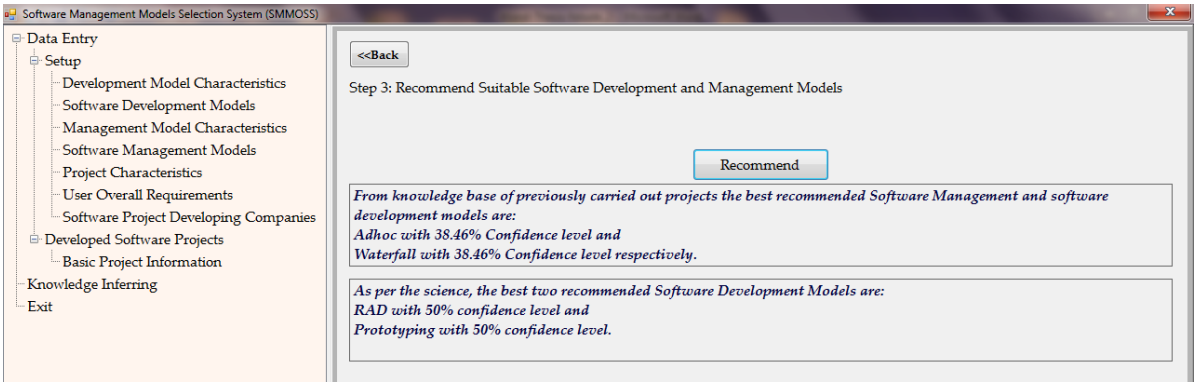


Figure 3: SSMOSS Software Management & Development Models Recommendation Form

6. Conclusion

An integrated software inference framework that could infer the appropriate model given system characteristics and overall user requirements (functional and non functional) of the new system and a Software Management Models Selection System (SMMoSS) that implements the integrated software inference framework are proposed.

References

[1] C. Hewagamage and K. P. Hewagamage, "Redesigned Framework and Approach for IT Project Management", International Journal of Software Engineering and Its Applications, Vol. 5, No.3, July 2011.

[2] Project Smart, "The Curious Case of the CHAOS Report 2009", http://www.projectsmart.co.uk/the-curious-case-of-the-chaos-report-2009.html, last accessed on October 24, 2011.

[3] A. U. Rehman and R. Hussain, "Software Project Management Methodologies/ Frameworks Dynamics - A Comparative Approach", Proceedings of Information and Emerging Technologies (ICIET) International Conference, 2007.

[4] F. Hui, "Research on the Developmental Management Model of Software Type IT Project", in proceedings of Computational Intelligence and Software Engineering (CISE) International Conference, Nanjing, China, 2010.

[5] D. Tesch, T. J. Kloppenborg, and M. N. Erolick, "IT Project Risk Factors: The Project Management Professionals Perspective", Journal of Computer Information Systems, Summer 2007.

[6] Schwalbe, K., "Information Technology Project Management", Boston, Massachusetts: Course Technology, 2002.

[7] Baccarini, D, "The Logical Framework Method for Defining Project Success", Project Management Journal, Vol. 30, No. 4, pp. 25-32, 1999.