Development of an Insurance Application Systems Interoperability toolbox for business organizations

Fungai Jacqueline Kiwa* Shakemore Chinofunga** Mary Shangwa*** Tinevimbo Chikondowa****

**Department of ICT and Electronics, School of Engineering Sciences & Technology Private Bag 7724, Off-Chirundu Road,+263 67 29467, Zimbabwe*

***Department of ICT, School of Engineering Sciences & Technology Private Bag 7724, Off-Chirundu Road,+263 67 29467, Zimbabwe*

****Department of ICT and Electronics, School of Engineering Sciences & Technology Private Bag 7724, Off-Chirundu Road,+263 67 29467, Zimbabwe*

***** Department of ICT, School of Engineering Sciences & Technology Private Bag 7724, Off-Chirundu Road,+263 67 29467, Zimbabwe*

Abstract

The Internet of Things (IoT) is one of the foremost rapidly creating areas within the innovation division. It houses numerous conventions, benchmarks of communications, and middleware stages. The fact that the IoT landscape is heterogeneous in nature is additionally one of its most basic challenges. There is a great need to make sure that there is a framework that guides the interoperability of services across business people. In any case, this has come about in numerous endeavors focusing on particular vertical IoT sectors, such as keen framework. To progress such endeavors, an interoperability system that is versatile in common is required. Interoperability is one of the key prerequisites within the Internet of Things, considering the diverse platforms, communication benchmarks, and specifications accessible nowadays. Characteristic asset limitations within the majority of the IoT devices make it exceptionally difficult to utilize existing arrangements thus demanding new approaches. This proposition makes a difference construct, the unique model for the IoT interoperability framework, and presents the method of adjusting a lightweight interoperability middleware for that reason, and assesses memory and control overheads. The middleware takes after a service-oriented design and classifies devices, agreeing to accessible assets to assign distinctive parts, such as Application, Benefit, and Director Nodes. These parts live totally different levels in a nonexclusive IoT design, where the Director Hubs are found within the middle Fog layer, the middleware characterizes and handles a set of occasions that are utilized to communicate with the client application.

Keywords: *Application Programming Interfaces (APIs), Internet of Things (IoT), Interoperability, Service Oriented Architecture (SOA), System Architecture*

I. Introduction

Service-Oriented Architecture (SOA) is rising as the premier integration and architecture framework in today's complex and heterogeneous environment. To enable different software which runs on different operating systems to communicate, share resources and information, there is a need for a system that integrates all of them. Service-Oriented Architecture (SOA), provides the infrastructure for building software applications that use services available on a network. Service-oriented architecture is an implementation of the service model of computing. In this architectural style, business processes are implemented as software services, accessed through a set of strictly defined application programming interfaces (APIs), and bound into applications through dynamic service orchestration. Using SOA can be positive for organizations as it allows resources from several different organizations to be dynamically pooled into virtual organizations to solve complex computing problems thus interoperability is promoted since the use of standardized communication protocol allows platforms to easily

transmit data between client and services regardless of the languages they are built on. This research focuses on developing an Interoperability toolbox that will enable resources or software applications to communicate with each other and heterogeneous resources from different organizations.

ii. Background

Smart Insurance Agency Marondera (SIAM), is an organization that deals with home insurance, car insurance, and life insurance. SIAM has knowledgeable staffs that are always ready to go out of their way to help individuals or companies to find the best coverage at the best price of their needs. Currently, SIAM uses an insurance agency management software system that runs on windows to get and share information about different insurance plans with prospective clients or existing customers. The system they have personalizes communication with its clients since different businesses or individuals use different operating systems like Android and Linux. Since Smart Insurance Agency Marondera acts as a middleman to connect insurance buyers and sellers to facilitate transactions, personalizing information is crucial. Lack of personalized communication with clients results in diminishing customer loyalty and reduced premium growth of SIAM. When Minerva insurance company introduced a home insurance policy, SIAM failed to provide awareness in time to its clients using its Windows system since 60 % of their clients were using Android platforms by that time.

Lack of cross-platform interoperability has created a gap between different systems which run on different platforms. SIAM fails to advise motorists on weather wherever their clients transport cars from one place to another. This company cannot access weather forecast information using their insurance management system to give motorists advice. This has a great impact on both SIAM and the insurance companies since insurance companies are responsible for all the risks associated with everything that happens during the transportation of cars. Thus, the sales for insurance companies are reduced, and the agent is not paid according to the agreement, which is a loss to SIAM Company, and revenue is reduced.

SIAM needs cross-platform interoperability in its day-to-day interaction with insurance companies and clients. This is because developing some of the required software from scratch to suit their needs is very expensive. A huge investment is required in terms of technology, development, and human resource for SIAM to build the software. Therefore, we will solve their problems by developing an interoperability toolbox that works with the SIAM windows system to communicate differently. Instead of viewing computers as individual devices, we want to integrate these physically separate but electronically connected resources into a single logically unified computational environment (Wegner 1996). Software interoperability technologies seek to smooth the integration of both legacy and new applications, even on heterogeneous hardware and software platforms. Improved integration benefits users within a company and facilitates sharing knowledge with other disciplines and organizations, enhancing the workflow processes. Demands for interoperability are increasing since many products are available to provide the same functionality, which has raised some heterogeneous issues. Once this Toolbox is built, SIAM will be able to use services available on the network such as SMS Pro to send notifications to clients when a new policy is released by any insurance company, Weather Forecast, to advise their clients about the weather transporting their cars. All these services will be integrated into our Toolbox using Service Oriented Architecture (SOA). SOA will enable SIAM's insurance agency management system to use online services without considering the type of platform used to build these services, all services or resources are loosely coupled. Cross-platform interoperability by SOA allows the company to use pre-build service and don't have to rebuild the same functionality from scratch. This will also increase the productivity of the company. Now the company can only focus on its software without worrying about external component integration. The company will not have to change or update their system to suit the services they want if the service changes anything because the Toolbox maintains the services. Therefore, SOA allows the services to communicate with other applications through a common language which means it is independent of the platform on which the application is running.

iii. Problem Statement

Currently, at SIAM different applications use different operating systems, but these applications cannot share data and information since they implement different interfaces. There is no communication at the backend of these software applications. One application is stuck with using only resources it has direct control over. This lack of integration of services is causing communication breakdowns, which are key causes of system inconsistency. No application is available that allows different applications from different operating systems to communicate. Availability of such an application could be useful to the company for communication and accessibility reasons to share resources. Such an application will allow SIAM's windows management system to request data from services that are available on a network, thus increasing ICT systems efficiency and flexibly adapting faster to

changes. Quality of information is progressed as more sources can be brought together and minimizes the time required to process information, in this way expanding organizational efficiency.

iv. Objectives and Aims

1. To analyze Interoperability in hybrid platform apps.
2. To investigate how different application can interact using Service Oriented Architecture (SOA).
3. To build an interoperability toolbox for SIAM which uses services that are available on the network.

V. Justification

The main purpose of this project is to build a consumer black box (web service) that removes restrictions on sharing of resources between the SIAM insurance management system and other services that are available on the network such as SMS Pro, Weather Forecast, and Insurance Information. Since all administrations are dependent, they cannot be effectively altered and overhauled and they can influence other services at some point in organizations. If the restriction on sharing resources is removed it means services can be reused to form numerous applications. Typically encouraged by the reality that Service Oriented Architecture (SOA) administrations are held in a service store and connected on-demand, making each service a generalized asset accessible to all subject to governance limitations. Reusing services permits organizations to spare time and lower costs related to advancement. In addition, users of the SIAM windows insurance management system can get many requests for the same service from different locations, thus to have access to services one does not need a particular network or specific system. (Bieber S, 2006). Therefore, the use of SOA improves proprietary communication and language protocols that make services unable to communicate with other platforms and different services to communicate employing a free coupling framework to either pass information or arrange an activity. In expansion, employing a standard communication convention permits organizations to diminish the level of interaction between services and clients. Bringing down the level of interaction permits applications to be scaled without including additional weight. The community/stakeholders are not able to share data and resources since no interface allows them to communicate and request services they want to use. Therefore, if SOA is used it means proprietary communication and language protocols that make them unable to communicate with other platforms will be improved.

VI. Remote Procedure Call (RPC)

RPC is a computer communication protocol that allows the program running on one computer to call subprograms on another computer without coding for remote interaction (Randy J.2000). An RPC is a synchronous operation requiring the requesting program to be suspended until the results of the remote procedure are returned. Therefore, the client is blocked while the server is processing the call and only resumed execution after the server is finished. RPC enables the usage of the applications in the distributed environment, not only in the local environment. The RPC specification itself doesn't provide any degree of well-defined interoperability between clients and servers from different implementations. The client and server use different execution environments for their respective routines, and the use of resources, e.g., files, are also more complex.

VII. Common Object Request Broker Architecture (CORBA)

CORBA is a standard developed in 1991 by the Object Management Group (OMG) to provide interoperability among distributed objects. CORBA is the world's leading middleware solution, enabling information exchange independent of hardware platforms, programming languages, and operating systems. The gap in this technology allows passing generic objects that can have methods, and there is no real CORBA standard to bind an ORB and its clients to a port, meaning there are only specific options. CORBA has standardized interfaces and a

standardized protocol too. Besides, during the standardization, it has been forgotten to determine the port of the communication. Thereby network administrators had to open a new port for every CORBA communication partner in the firewall. (Helmat P, 2004).

VIII. Distributed Component Object Model (DCOM)

Distributed Component Object Model (DCOM) is one of the emerging standards for distributed objects. DCOM provides distributed component communications. The DCOM aims to intercept all the local inter-process communication messages and substitute them with a network protocol based on the Distributed Computing Environment (DCE). DCOM provides the object glue that allows business applications to work across the Web. Apart from what was done on DCOM, the technology has a challenge that it failed to solve, such as platform dependence. DCOM does not support multiple inheritances, and rather, it supports multiple interfaces.

IX. Research Gap/Main Problem

Lack of cross-platform interoperability is a major problem in computing. Most software's are platform-dependent, meaning they run only on windows, android, etc. Therefore, the communication between the different systems is hindered and could not be achieved effectively using DCOM, RPC and CORBA.

X. Methodology

This chapter discloses the techniques and strategies used by the authors during the research project. The methodology includes the methods, procedures and techniques used to collect and analyze information and the instrument used to collect data.

Data Collection Methods

The methods that were used for data collection in this research are interviews and questionnaires.

a. Questionnaires

A questionnaire was chosen as a data collection instrument. A questionnaire is a printed self-report form designed to elicit information that can be obtained through the written responses of the subjects. (Burns & Grove 1993). The data was collected over two weeks. This method was chosen because questionnaires are easy to use in investigating patterns and frequency. The questionnaire that was used helped explore user needs, expectations, perspectives, priorities, and preferences of the system to be developed. There was less opportunity for bias as they were presented invariably.

b. Interviews

An interview is a method of asking quantitative or qualitative questions orally from key participants. Quantitative questions are closed-ended and have specific answers to choose among that can be categorized and numerically analyzed. Qualitative questions are open-ended, the respondent responds to his or her own words. Informal conversational interviews were used to collect data. The researchers chose to use interviews for interviewing Smart Insurance Agent Marondera and other individuals from CBZ bank. This is because interviews permit face-to-face contact with respondents. They also provide an opportunity to explore topics in-depth and allow the interviewer to explain or help clarify points highlighted, increasing the likelihood of useful responses. We used interviews to help us build an interoperability framework which will lead us to creating a tool box. Although interviews were chosen as a data collection method, they carry some disadvantages, including that they can be very time-consuming, costly, and different interviewers may understand and transcribe interviews in different ways.

xi. Data Analysis Method

After the data was collected, it was organized and analyzed. For the analysis of closed-ended questions, a computer program called Statistical Package for Social Sciences (SPSS) was used. Data were analyzed by using descriptive statistics. Frequency tables were drawn, and from these, the data was presented in pie diagrams and bar graphs. The open-ended questions were analyzed through quantitative content analysis by the researcher to quantify

emerging characteristics and concepts. Concept analysis is the process of analyzing verbal or written communications in an organized way to measure variables quantitatively (Polit & Hungler 1995).

xii. System Architecture

The diagram below shows the system architecture.

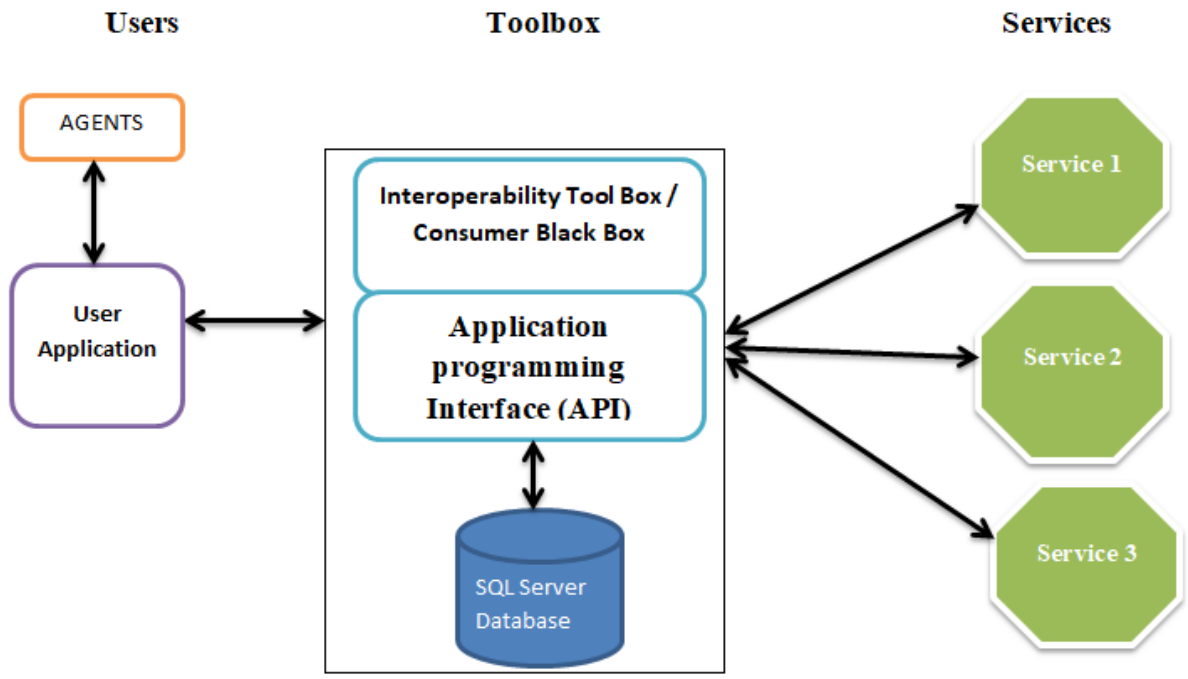


Figure 1: Gateway Api-Azure Architecture Center | Microsoft Docs

a. Software Development Tools

To develop a Consumer Black Box, the following tools will be used:

- SQL Server Management Studio 2018
- C-sharp using Visual Studio 2019
- ASP.net Web API
- Extensible Markup Language (XML)
- Java Script Object Notation (JSON)

b. Software Design Methodology

Design science methodology is an investigation and design of artifacts in context. It aids a build-and-evaluate loop (when applied in software development) which is typically iterated several times before the final design artifact of the system is generated. The researchers, therefore, chose design science methodology because of its ability to conform to their context and intended development procedure precisely. The methodology process includes awareness of the problem, suggestion, development, evaluation, and conclusion.

xiii. Awareness of the problem

The problem awareness stage offered a solid and important foundation for the overall research process as it aids us, the researchers, specify the framework at large and verify its practical relevance to the business. This framework will guide us into the development of an interoperability toolbox. Identifying the problems involved with the lack of communication between different systems to communicate using the different operating systems, we interviewed some of the SIAM management System users and other members of CBZ bank. All these people highlighted the problems they faced in the lack of communication between systems.

a. Suggestion stage

At this phase, less attention was centered on the problem rather, emphasis was on the possible solution for the identified problem. We suggested developing a Consumer Black Box, which overrides the lack of communication between ICT systems.

b. Development stage

The prototype design was further developed and implemented in this phase. We used Agile Development to come up with the prototype. Each stage was reviewed with the client to produce requirements that feed the next version.

c. Evaluation stage

Design science involves the steps which include the case of going back to the knowledge contribution phase in which the end-users of the system were being asked if the developed artifact is solving the problem statement, if not, then the researchers restarted the process again that is starting to the awareness of the problem, then suggestions then the development phase.

d. Conclusion

We chose design science because it is an outcome-based information technology research methodology that offers specific guidelines for evaluation within research projects. This method gives room for design validation to evaluate the Consumer Black box (Interoperability Toolbox). It provides feedback information and a better understanding of the problem to improve both the quality of the proposed system and the design process. We used this method because it allows us to demonstrate how different systems can communicate using Service Oriented Architecture (SOA).

xiv. Proposed system framework

The proposed system ensures system interoperability by combining different services like insurance, weather forecast and many others. Below is an image of a screen capture were insurance values are captured, no empty values ae accepted into the system.

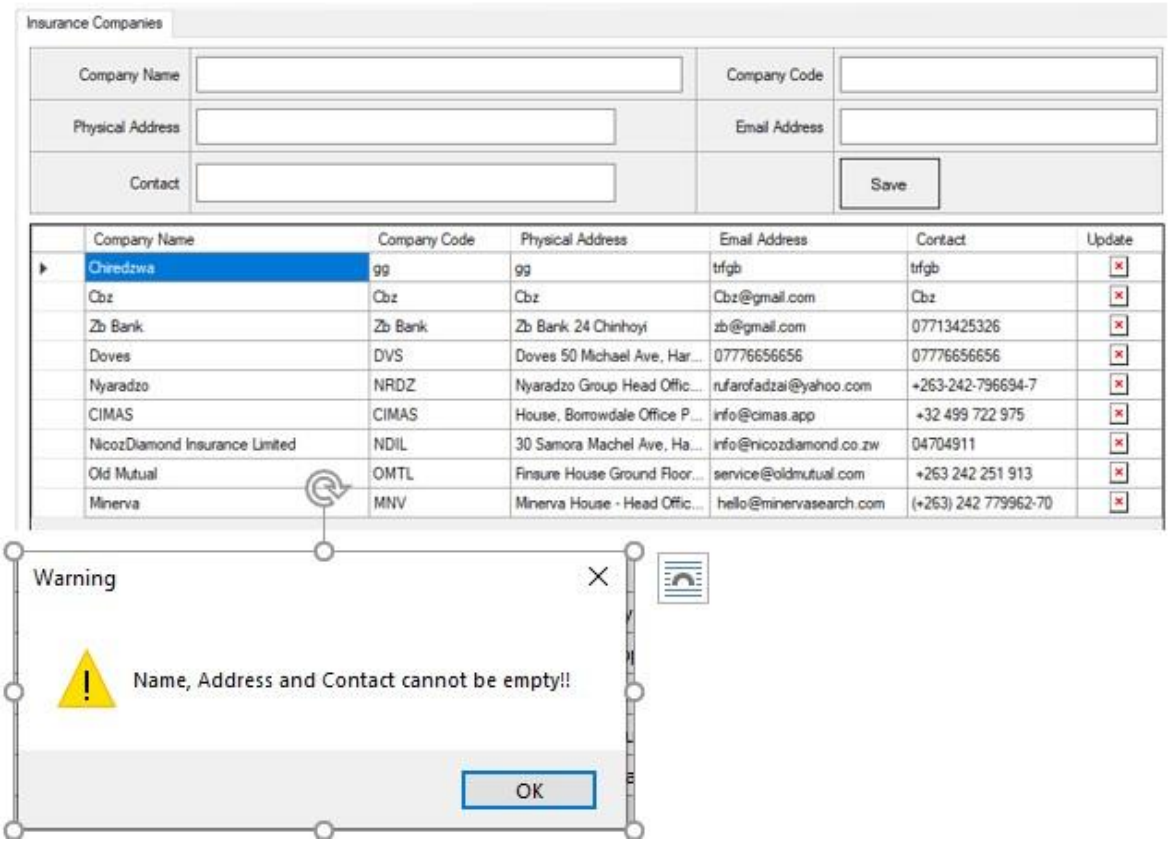


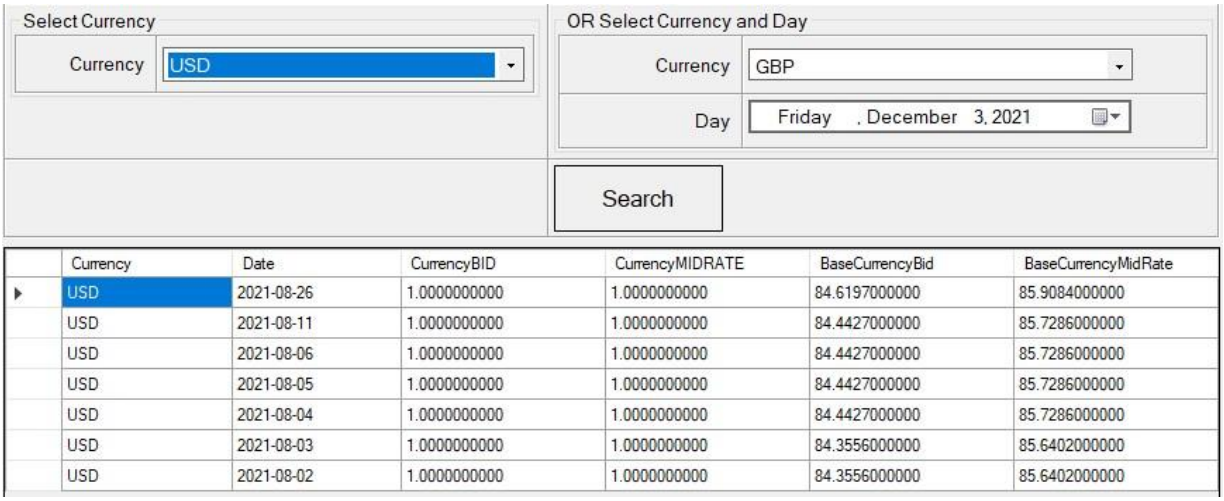
Figure 2: Adding Insurance Company values

When requesting types of insurance associated with an insurance company.

Figure 3: Invalid result for type of insurance and Insurance Company

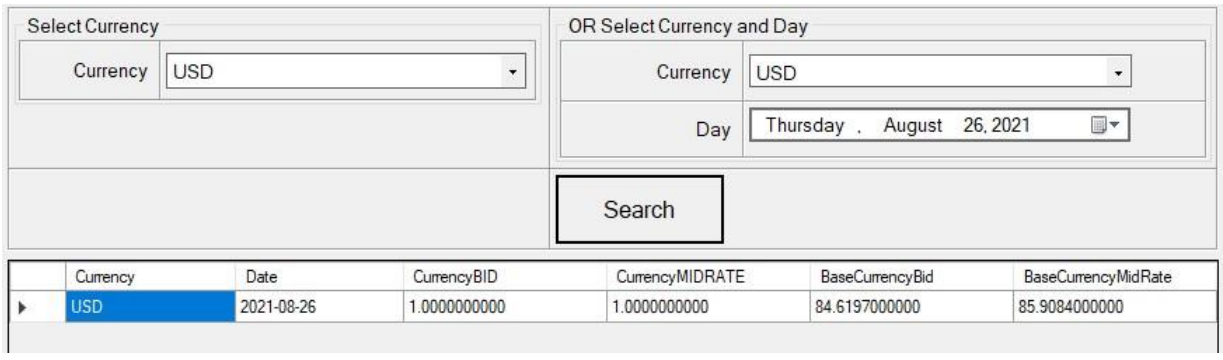
Searching of Exchange rates

The agent can search for exchange rates per day and currency or using currency only



Currency	Date	CurrencyBID	CurrencyMIDRATE	BaseCurrencyBid	BaseCurrencyMidRate
USD	2021-08-26	1.0000000000	1.0000000000	84.6197000000	85.9084000000
USD	2021-08-11	1.0000000000	1.0000000000	84.4427000000	85.7286000000
USD	2021-08-06	1.0000000000	1.0000000000	84.4427000000	85.7286000000
USD	2021-08-05	1.0000000000	1.0000000000	84.4427000000	85.7286000000
USD	2021-08-04	1.0000000000	1.0000000000	84.4427000000	85.7286000000
USD	2021-08-03	1.0000000000	1.0000000000	84.3556000000	85.6402000000
USD	2021-08-02	1.0000000000	1.0000000000	84.3556000000	85.6402000000

Figure 4: Searching Exchange Rates using Currency



Currency	Date	CurrencyBID	CurrencyMIDRATE	BaseCurrencyBid	BaseCurrencyMidRate
USD	2021-08-26	1.0000000000	1.0000000000	84.6197000000	85.9084000000

Figure 5: Searching Exchange Rates using Currency and Day

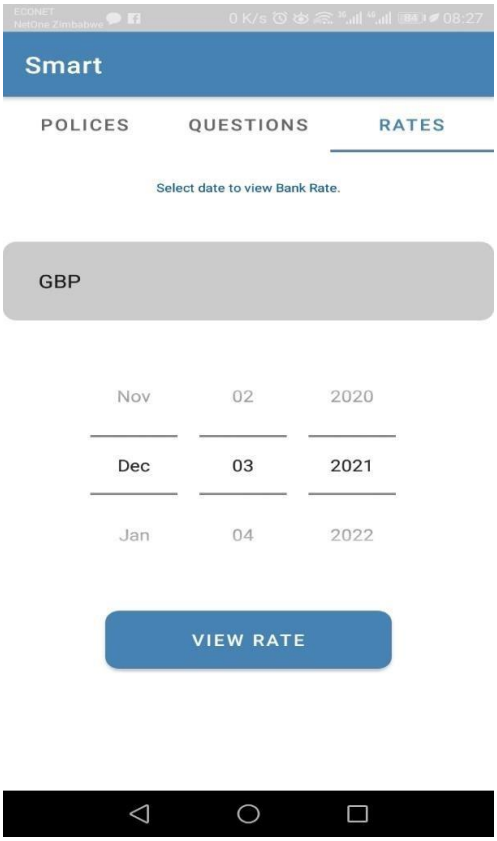


Figure 6: Clients Searching for Exchanges using Currency and Day

Searching For Weather Conditions

When a client wants to transport their cars form one place to another, it is the duty of Agents to make sure that the weather condition for that certain area is perfect for clients to transport without interferences. The diagram below shows the results of weather conditions in Harare on 3 December 2021 at 10.40am.

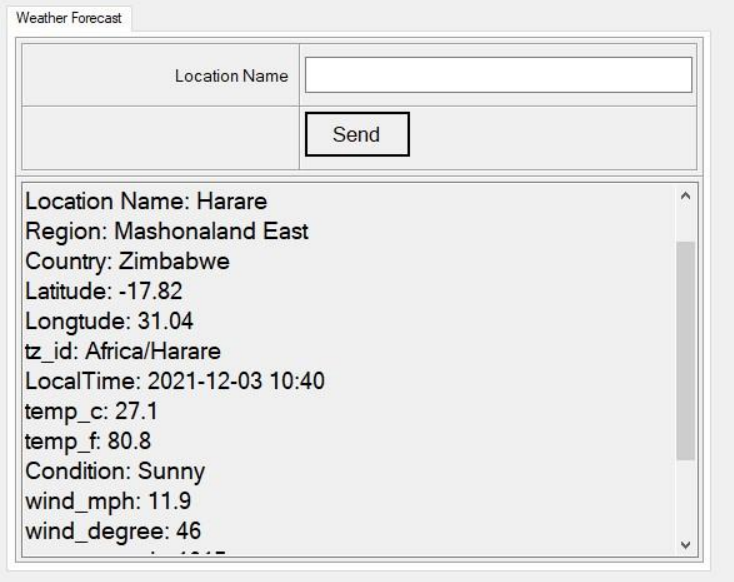


Figure 7: Searching for Weather Conditions

Requesting For New Policies

The agents can request for new policies from insurance companies, the result below shows the type of insurance that SIAM has given by Nyaradzo as an Insurance Company:

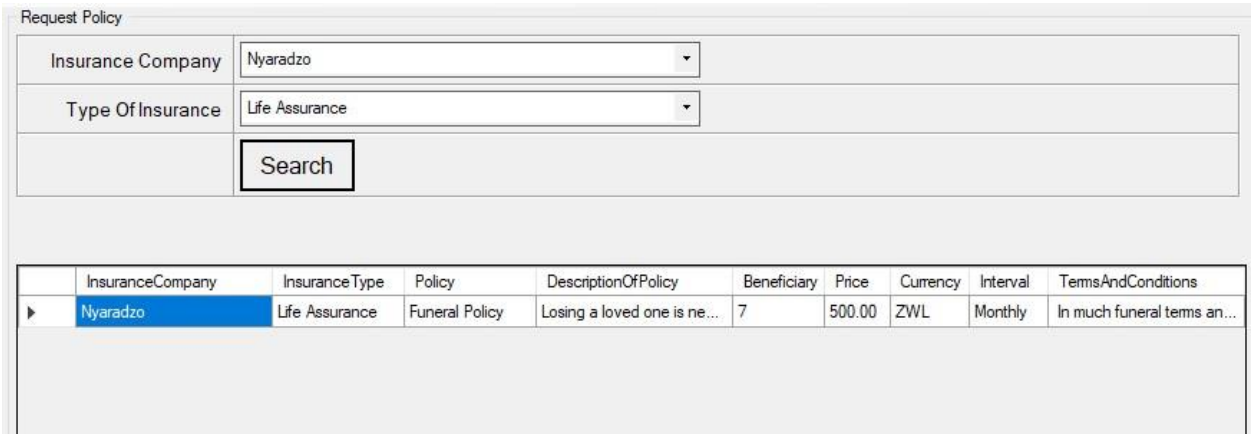


Figure 8: Requesting for new Policies form insurance companies

Clients Questions

Using Mobile App or Web-based app, after the client sends a message to the Agent, the client can view the sent message and view whether it was replied or not as shown below:

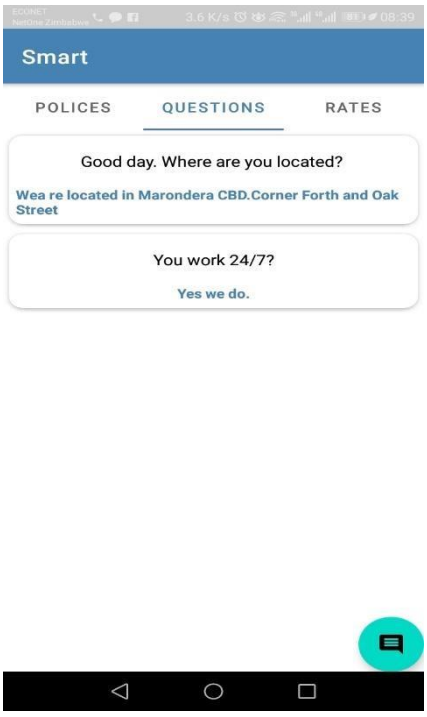


Figure 9: Questions and Replies

View Clients Questions and Reply Question

The System testing which is a desktop application, provides a platform for agents to view clients' questions and reply the question sent, the diagrams are illustrated below:

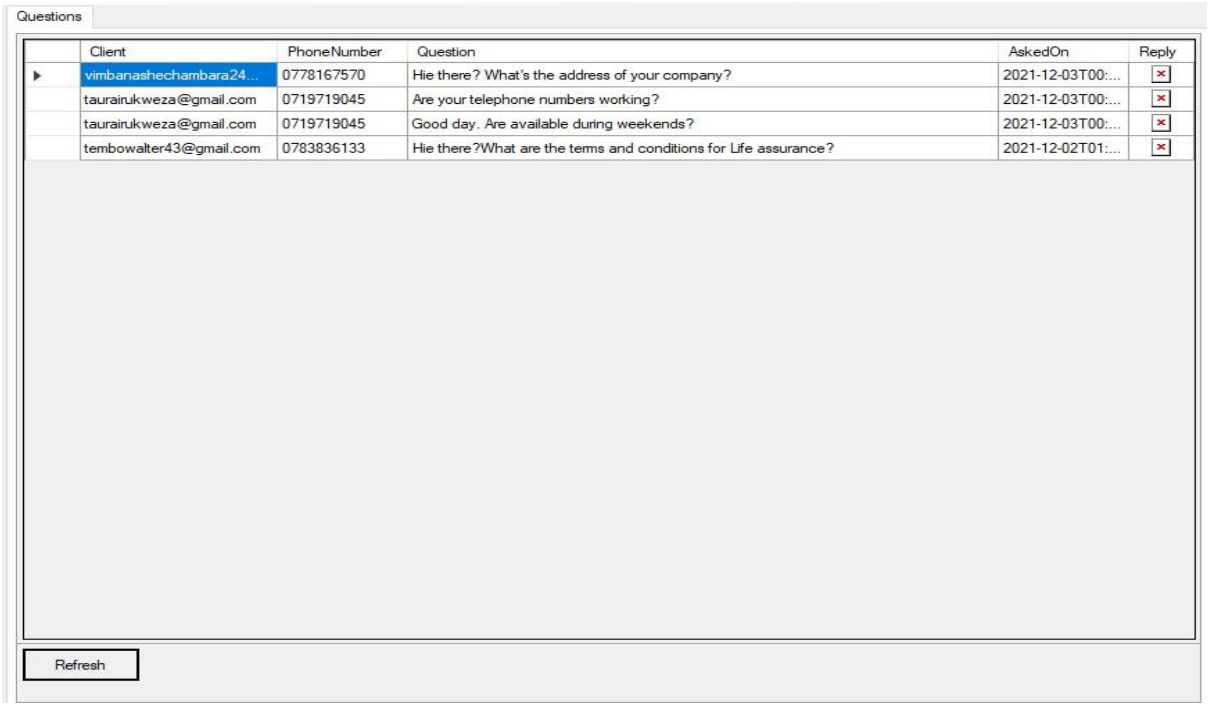


Figure 10: View Clients Questions

After Agent click on reply on a particular client's question, the below form is shown to the agent

View Clients Requests and Responses

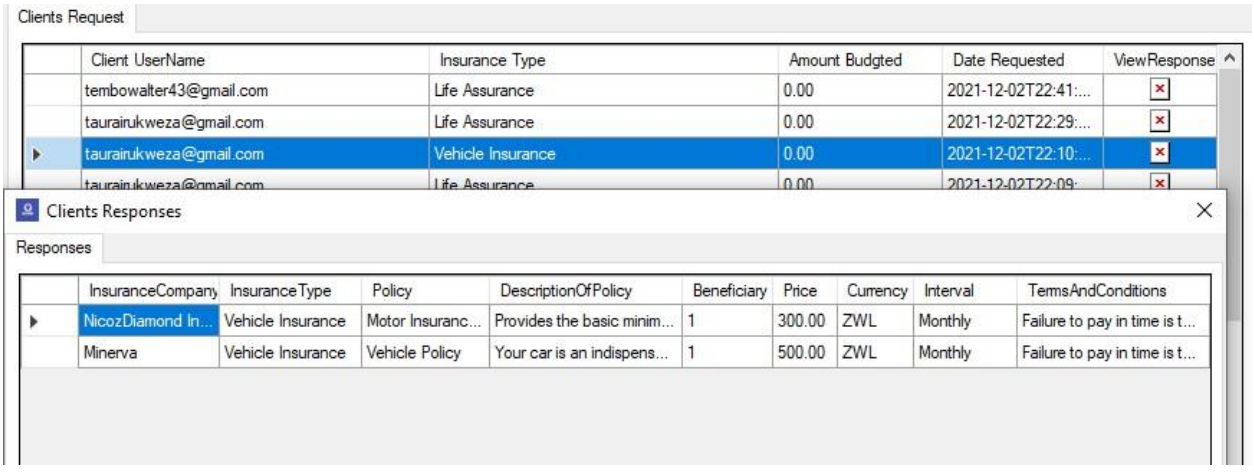


Figure 11: Clients Request and Responses

Adding Insurance Companies

The agents can save an insurance company they want to do business with as shown below:

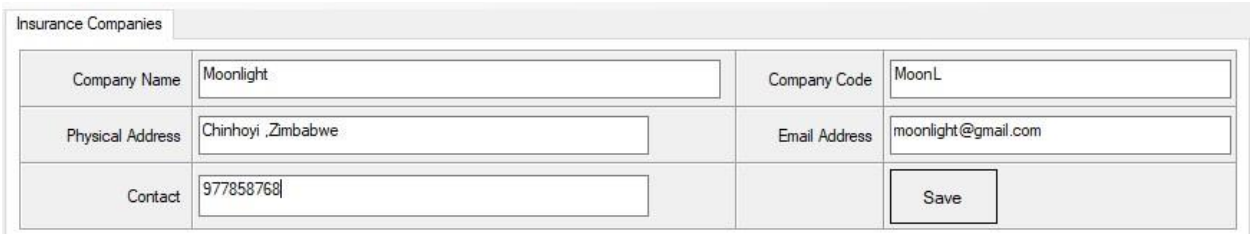


Figure 12: Adding Insurance Companies

Requesting for Insurance Policy Information

The Client System Testing which is either a mobile app or web-based app, provide a platform to request for insurance policies whenever the client wants to choose a policy. The diagram to illustrate this is shown below:

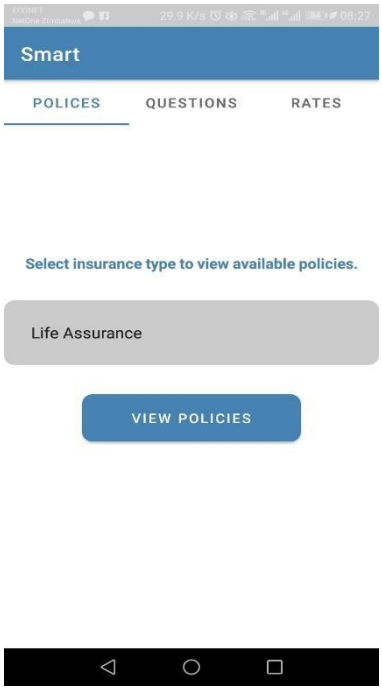


Figure 13: Requesting for Insurance Policy Information

xv. Conclusion

The research brought forward information on how much problems were faced because of the non communication of systems through the interviews done. The design methodology was applied and proper steps were carried out to complete each task. A better understanding of the business framework led to the development of an interoperability toolbox. A prototype of the system was developed which was tested and proved to be working well. The problems that were faced by the SIAM Company reduced because the systems could communicate effectively with each other without boundaries of different operating systems. The system was named SIAM Interoperability toolbox, the system prototype proved to be very effective to the extent that some companies also what the same system for them.

xvi. Future Research

Lines of research that may follow:

- Assess the utility of the created theory on interoperation and interoperability with professionals from different proficient parts such as commerce and prerequisites investigator, venture directors, policymakers, CIOs, CFOs, CEOs, planners, legal ability, engineers, et cetera.
- Assess the transferability of the theory on interoperation and interoperability to be utilized in other spaces such as the health care and public sector.
- Assess the transferability of the down to earth theory to be utilized in other nations exterior Zimbabwe;

Reference

1. George, T. Aphrodite, and P. Michael (2006.), "Interoperability among Heterogeneous Services," in Services Computing, SCC '06. IEEE International Conference on, pp. 174-181.
2. Gustavo A, Fabio Casati, Harumi Kuno and Vijay Machiraju (2004), Web Services: Concepts, Architectures, and Applications, Springer-Verlag Berlin Heidelberg
3. Grid Wise™ Architecture Council (2008), "Interoperability Context-Setting Framework".
4. Cox, D. E., and Kreger, H. (2005). "Management of the Service-Oriented-Architecture Life Cycle," IBM Systems Journal 44(4), 709-726.
5. Marks, E. A., and Bell, M. (2006). Service-Oriented Architecture: a Planning and Implementation Guide for Business and Technology, John Wiley & Sons, Hoboken, NJ
6. Papazoglou, M. P., and Georgeakopoulos, D. (2003). "Service-Oriented Computing," Communications of the ACM 46(10), 25-28
7. Jeffrey Kaplan et al (2005)., "Roadmap for Open ICT Ecosystems," Berkman Center Publication Series, p. 4 - 5.
8. Jonathan Zittrain (2008), The Future of the Internet – and How to Stop It (Yale University Press).
9. Jonathan Zittrain (2006), "The Generative Internet," 119 Harv. L. Rev. 1974.
10. Eric von Hippel (2002), Open-source projects as horizontal innovation networks - by and for users, MIT Sloan School of Management.
11. Clayton Christensen (1997), The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail (Harvard Business School Press).
12. The Component Object Model Specification, Oct. 1995
13. N. Brown and C. Kindel (1996), Distributed Component Object Model Protocol -- DCOM/1.0, Nov.
14. Y. Huang and C. Kintala (1993), "Software implemented fault tolerance: Technologies and experience", Proc. IEEE Fault-Tolerant Computing Symp., pp. 2-9, -June
15. Windows NT Magazine (1997) "Clustering Solutions for Windows NT", pp. 54-95.