Architecting an SOA-based System for Sharing of Peripheral Network Resources

Wandile Sibewu and Mamello Thinyane
Department of Computer Science
University of Fort Hare, Private Bad X1314, Alice 5700
Tel: +27 40 6022464, Fax: +27 40 6022464
email: {wsibewu, mthinyane}@ufh.ac.za

Abstract - A number of new developments are being made in the Information and Communication Technology (ICT) field and one of such is the adoption of Service Oriented Architecture (SOA), which not only allows humans to communicate with software services but also facilitates communication between different computing agents. At the same time ICT is increasingly being adopted as an enabler for economic development within Marginalized Rural Areas (MRAs) where resources are scarce and infrastructure is limited. The challenge of limited resources (e.g. peripheral devices) in MRAs can be addressed through the implementation of services that enable the sharing of such limited resources. This paper describes the research that has been done in developing Web Service wrappers for standard computing peripheral devices such as scanners, fax machines, and printers. The service is developed for use in MRAs, but can also work for Small Medium Enterprises (SMEs) or any environment that requires the convenience offered by the service. The paper will go on to discuss the architecture and the implementation details of the service.

Index Terms—Services Oriented Architecture, Legacy systems, Web services, Siyakhula Living Lab

I. INTRODUCTION

Service Oriented Computing (SOC) and Service Oriented Architecture (SOA) have become the popular choice for the implementation of services delivery systems which are distributed and interoperable [1]. SOA is defined as an information strategy or approach where applications make use of other services available on the internet to achieve their computational goals [2]. Some of the features that make SOA so appealing are loose coupling of the components and the fact that it is based on open standards, which makes SOA platform independent. In the ICT industry the popular use for SOA has been for modernization of legacy applications that are crucial to companies delivering their services to consumers and other companies [1, 3]. A common implementation of SOA is Web Services and the World Wide Web Consortium (W3C) defines Web Services "as a software system identified by a Uniform Resource Identifier (URI), whose public interface and binding are defined and described using XML, its definition can be discovered by other software systems and these systems may interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by internet protocols" [4].

Web service operations normal consist of 3 interacting entities; there is the service consumer which could be another service that wants to consume a service, there is a service provider which is the entity that owns the service, and a service registry where services can be published and discovered. Web services are constructed using open standards, such as Simple Object Access Protocol (SOAP) for exchanging eXtensible Markup Language (XML) messages, Web Services Description Language (WSDL) as the service description language and Universal Description Discovery and Integration (UDDI) as the services registry. Since Web services are an implementation of SOA, they provide an easy and flexible way for modernizing legacy system and making them available as web services, this is done by creating SOAP wrappers around core legacy system functionality making them able to process SOAP request [5]. NASCIO describes a legacy system as a system that is unable to provide adequate support for its core functionality requirements or meet expectations for use of modern technologies, such as workflow, Instant Messaging (IM) and user interface [6]. By this description, most document processing peripheral devices can be classified as legacy systems, because they depend mostly on the proper drivers being installed and only newer printers have native support for accessing the internet and can be accessed from the internet.

Computers have been able to share things such as files, peripheral devices and internet connections for a long time now. The sharing of devices such as printers has its advantages such as allowing a greater number of people to access scarce resources and reduce the cost of providing computer equipment, but sharing resources has its difficulties. Sharing printers for instance requires a certain level of skill, in acquiring the necessary device drivers and configuring the sharing of the devices for access by different platforms. These skills might be in abundance in urban areas, with those communities having been exposed to ICTs for a long time, but this is not always the case in the MRAs where the communities haven't been exposed to ICT over extensive periods of time.

This project is undertaken as part of the Siyakhula Living Lab (SLL) based in Dwesa region of the Eastern Cape. Dwesa represents a typical South Africa rural reality in that it suffers from poor infrastructure when it comes to electricity availability, minimal telecommunication infrastructure, poor road quality and sub-standard education facilities [7]. The ICT infrastructure that was introduced to the community when the SLL project was started has greatly improved the community's access to ICTs. The ICT equipment is housed at the local schools making them the only Digital Access Nodes (DANs) at the moment. This

setup has a few limitations in that people with the most access to these limited resources are the students and teachers [7]. This limitation of network availability is soon to change, because current effort is focused on extending the network to the rest of the community by creating Wi-Fi hotspots at the DANs. There is also work to implement Mobile WiMAX to increase the coverage area to most of the Dwesa community [8, 9]. These extensions will increase the need to find new ways of sharing peripheral devices across the network so that they can be accessed by the entire community of Dwesa.

Some of the services that have been developed within SLL include but are not limited to: eCommerce, eGovernment, eHealth, eJudiciary, a help desk service, and a VoIP service [10]. Most of the services that have been developed for the SLL have been standard eServices and the project has started to move to more modern techniques of developing applications such as SOA. For example a project which aimed to develop web service wrappers for communication services such as IM, Email, has been undertaken within SLL [11]. The SLL services platform is evolving into an SOA platform and the existing "Legacy services" are going to be ported to this new architectural design through the use of the an SOA middleware (called Teleweaver) that is being developed by Reed House System and the two Universities that are involved with the SLL [12]. It is because of the fact that Dwesa, like any other rural community has limited resources and that the platform is already evolving into an SOA platform that we decided to develop an SOA-based system for sharing of peripheral devices that will leverage the existing platforms capabilities and give wide spread access to resources without the need of complex configurations.

In the following sections of this paper, we will discuss the related work that has been done to share devices using web services and other methods. We will discuss the methodology we have taken to address this problem and the system design that arose from the methodology. We will look at system functionality and finally conclude by giving recommendations for future research and possible extension to this project.

II. Related work

Work that has been done in the field of using web services to facilitate the communication\interaction with peripheral devices has been commissioned or undertaken with the help of large corporations. Baskerville *et al*(2005) highlights the most common use of SOA, besides developing new applications that can interact with each other, is for modernizing legacy systems [1]. Baskerville's study mainly focuses on the efforts of two European banks that want to compete in the global economy by offering their core banking functionality to their customers. This kind of web services usage has led to the quick adoption of SOA in the corporate world and is driving the spread of SOA to other industries.

An example of this spread can be seen by the number of projects in the industry automation field that have been done in the past few years. Service-Oriented Cross-layer infrastructure for Distributed smart Embedded devices (SOCRADES) is one such project which aims to investigate the usage of Web services in the area of automation [13]. This project is being funded by the European Community and its Service oriented Approach is being adapted to the level as low as the automation devices or facility automation [13]. Projects that are directly derived from the work at SOCRADES include work that is conducted at the University of Loughborough funded by Ford to provide an SOA system capable of supporting reconfiguration of production lines and Enterprise level device management for the Ford powertrain assembly machine [14]

Research funded by HP and conducted at PUCRS University in Brazil was creating Web services applications for HPs LaserJet 4600 series printers [15]. The main focus of this application was to develop two ChaiServices, one is used to provide the printing URL and the other is used to announce the device hosting the web service as a Universal Plug and Play (UPnP) device in the network. ChaiServer is HP's web services provider and it is implemented using Java, and so runs on a Chai VM [15]. UPnP was used to dynamically discover ChaiServices on the network.

HP ePrint is designed for small and large enterprises, with employees who move around to be able to print documents from any devices that can connect to the internet [16]. There are two ways that people can use this service:

- They can use a BlackBerry Device, to send a document to the BlackBerry Enterprise server, then the document is sent to the HP ePrint Enterprise Server finally the document is sent to the selected printer as a normal print job.

- They can attach the document that is to be printed in an email, with the destination email address being an address assigned to a printer by HP. The email is sent to the HP ePrint Enterprise Server, which then sends it to the printer corresponding to the email address.

This innovation requires an HP ePrint Enterprise server, for the mobile clients, any device running either BlackBerry OS or iOS or Android or webOS can access the printing service and another requirement is the actual printer, which can be an HP LaserJet or Inkjet or any other printer that is compatible with PCL5/6, PCL3 and PCL3GUI [17, 16].

When Google was building its web based operating system (Google Chrome OS) they didn't want to develop a complex printing subsystem and print drivers for the many computing platform that are available these days, so they built a printing experience that enables any application (desktop, web or mobile) on any devices to print anywhere in the world [18]. The applications that can use the Google cloud Print access Google Cloud Print APIs to collect the necessary data to show a print dialog. This web service will associate printers with Google Accounts to make sharing printers much easier and quicker [18].

The web service will work with two types of printers, one is a concept that Google calls cloud-aware printers, these printers have native support for connecting to cloud services and they don't exist yet. The second kind of print is what they call legacy printers; these are the printers that we use today. These legacy printers will connect to Google Cloud Print through proxy software [19]. This proxy runs on Windows, Mac and Linux machines, it acts as Protocol Bridge between the Google Cloud Print and the native print

drive stack on existing PCS. Main functionality of the proxy is [19]:

- To detect local and network printers on a client machine and register them with the Goole Cloud Print.
- To listen for XMPP print request notification from Google Cloud Print, then fetch the requested job for that specific proxy from the Google Cloud Print and spools them to the local printer.
- To check for notifications and job status change and inform the Google Cloud Print. Newly added printers are registered with the Google Cloud Print and the operation of the proxy makes the service platform-independent.

III. METHODOLOGY

The research method chosen to help produce a product that not only attends to the needs of the community but is robust and easy to use, has been a combination of well established methods such as observation, literature review prototyping and evaluation.

A. Observation

Determining the user requirements for a project such as this one is a complicated matter because devices already have a set of functions that they perform. It is near impossible to conduct interviews and questionnaires cannot be conducted in order to obtain additional information on what users expect from a service such as this one, this led to observation being used as the primary method for the collection of user and system requirements. This method only assisted in determining which functionality should be in cooperated in the first prototypes. Observation is a qualitative research method that is used to collect data about a phenomenon in its natural setting [20].

In order to come up with a solution that will suit the community of Dwesa and leverage the potential of the network infrastructure that already exist, the following had to be observed:

- The location called Dwesa
- The physical network layout and configuration
- The interaction between the users and the peripheral devices

B. Literature review

Most of the information that is required to design and develop the system is already available in past research and on computer programming manuals, so a thorough literature review had to be conducted to mine the necessary information. Data was gathered on the following topics:

- Past and present projects.
- Services Oriented architecture (SOA) and Web Services.
- Resources sharing on different platforms.
- Different computing peripheral devices.

Researching about similar projects from the past and the present helped to shape this project, and helped in choosing the best technologies to use to deliver a complete solution. Since the project is about delivering an SOA based system for sharing peripheral devices, it was imperative that we know as much as possible about SOA and its different implementations, with Web Services being the chosen implementation for the service. Final literature had to be collected about the different peripheral devices that are used in Dwesa on a regular basis and how these devices are shared on the different computer platform that are used by the community of Dwesa.

C. Prototyping

As with any system that is going to be used by people, prototyping is essential to make sure that the system you are developing will be usable. Software prototyping refers to the activity of creating prototypes of software applications, where a prototype is a system that is initially incomplete, and is to be modified, expanded, supplemented, or supplanted [21]. Advantages associated with prototyping include reduced time and costs associated with software design and development; this is mainly because changes in the system are done early which brings down cost. All software development techniques have their disadvantages; some of prototyping disadvantages include user confusion between a prototype and completed product. Different prototyping approaches achieve different results, we have chosen to use vertical prototyping, this is because vertical prototyping allows us to develop a single subsystem in great detail and then test that subsystem before moving on to the next subsystem or module [22]. This has the advantage that the users can start using the finished subsystem as soon as possible giving feedback that will inform the modification and improvements if necessary.

D. Evaluation

Evaluation is defined as the process of assembling evidence that a system meets or fails to meet, the user requirements set at the beginning of the development life cycle [23]. Since Dwesa is an underprivileged community there is little computer skill, so when the SLL was started it was necessary to provide computer literacy training to the people in the community, teachers and the students at the schools. This training was initially provided by the University of Fort Hare and Rhodes University during the monthly visits to Dwesa [7]. The training serves two purposes one is to equip the community with the necessary computer skills that will allow them to participate in the global information community and the other is so they can help in evaluating and testing the projects that are developed to help them better their lives through the use of Information and Communication Technology (ICTs).

There are two ways in which system evaluations are carried out:

- Verification – Is used to determine that a software product is built correctly and does not contain any technical errors. It reviews the requirements, to verify that the right problem is being addressed by the solution and it ensures that the software performs functionally as specified [24].
- Validation – Helps to insure that the meaning and content of the rules meet some carefully defined criteria of adequacy [24]. Defining such criteria is the key to successfully conducting a validation procedure and demonstrating the level of acceptability of the system.

Software evaluation is an integral part of software

development; it needs to happen at every step of the software development process [22]. This can be done by doing functional and usability testing on the complete modules of the system, which intern will form part of the verification and validation of the system.

When the system is completed it will be deployed in Dwesa for evaluation and use by the Dwesa community. This presents us with the great opportunity to conduct a complete evaluation and field test of the system. Questionnaire and interviews will be conducted with the tester and evaluators of the system to find out both qualitative and quantitative data about the system. In adopting as semi-structured approach in constructing the question for both the questionnaire and interview will give a good mixture of both qualitative data and quantitative data. Questions such as:

- Is the system easy to use?
- Is the system consistent?
- Is the system faster or slow that the previous way of sharing resources?

Because most of the community members are illiterate, there is language barrier in Dwesa that make the construction and delivery of questionnaires and interviews difficult that it is to construct them for non-rural areas. Since most of the community speaks only isiXhosa some of the questionnaires and interview will be in their home language.

IV. System Architecture

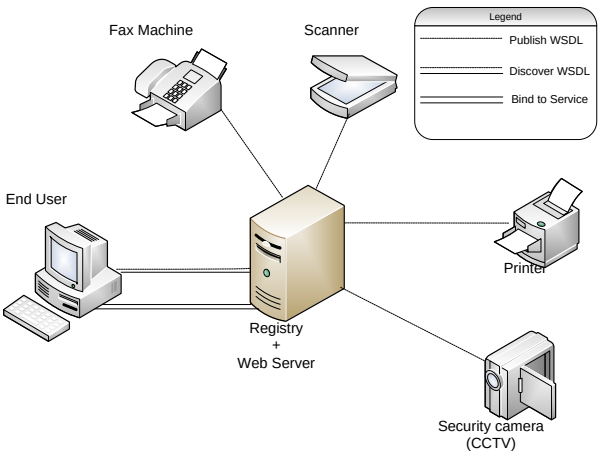

Figure 1: System Architecture adopted from [25]

The system is designed to follow the classical architectural design of Web Services to provide access to the resources as services (the specific function of a device). Users and other services (requesting agents) can request service descriptions from the Universal Description, Discovery and Integration (UDDI). Each services description for a resource will be published to the local affiliated UDDI registry. An affiliated UDDI registry is a registry that is deployed in a controlled environment but with limited access by authorized clients meaning that data may be shared with other registries in a controlled manner [26].

After requesting the service description and the request is authenticated, the requestor will receive a service description written as a Web Services Description Language (WSDL) document that contains all the information necessary to successfully invoke the service. After discovering the resource, requesting agents will communicate with the discovered service using XML-based messaging. Simple Object Access Protocol (SOAP) is a commonly used standard for sending and receiving XML messages. As the final stage before using a resource, a requesting agent has to bind to the service.

The Web Services stack is not limited to XML, WSDL, UDDI and SOAP, there are other specifications that need to been in cooperated when developing Web Services. Quality of Service (QoS), Security and management have to be addressed at all layers of the Web Services stack from the network which will use HTTP, XML-Based Messaging which will use SOAP, Service Description which will use WSDL to Service publication and Discovery which will use UDDI [27].

As mentioned in the introduction, currently in Dwesa the points of presences or the Digital Access Nodes (DANs) are the schools and the surrounding areas through Wi-Fi hotspots. The proposed application will be deployed in each DAN to manage interaction with local network peripheral resources through the service. The affiliated UDDI registries that will be available in each DAN will share data (service descriptions) with each other to promote sharing of network peripheral devices in Dwesa.

V. System Functionality

The resource sharing system is going to provide multi document processing peripheral devices as web services, but at the time of writing this paper only printing has been implemented. The choice to start by implementing printing was informed by the observation done in the community and by the fact that we adopted vertical prototyping, which permits one to concentrate on one subsystem when developing. The following diagram shows the interaction that has to occur for the system to print a document.

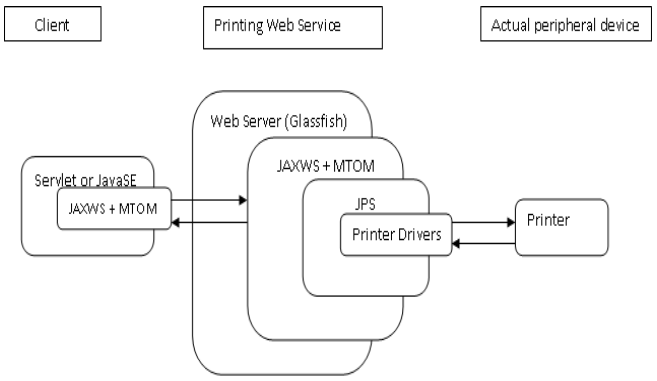

Figure 2: Low Level Architecture

At the time of writing this paper, the available functionality on the services is printing documents. A user can run a simple web service client written in Java SE, which allows the attachment of a document to a SOAP message that will be sent by Java API for XML Web Services (JAXWS) to the Print Web Service. JAXWS is a technology for building web services and clients that communicate using XML; it allows developers to write message-oriented as well as RPC-oriented web services.

The SOAP message is then received by JAXWS on the Print Web Service; the attachment is extracted and is passed to the Java Print Services (JPS). JPS is an API that is

designed to support printing on all Java platforms including platforms that require a small footprint, but also supports the current Java 2 Print API. This API uses the document to construct the document that's sent to the printer and it conducts the communication with the printer drivers. Finally the printer driver\printing service that is on the machine hosting the Print Web Service does the actually printing.

Ideally the system will be used by three types of users namely the administrator, the normal user and other services, which will all have access to functionality that is suitable for their respective roles:

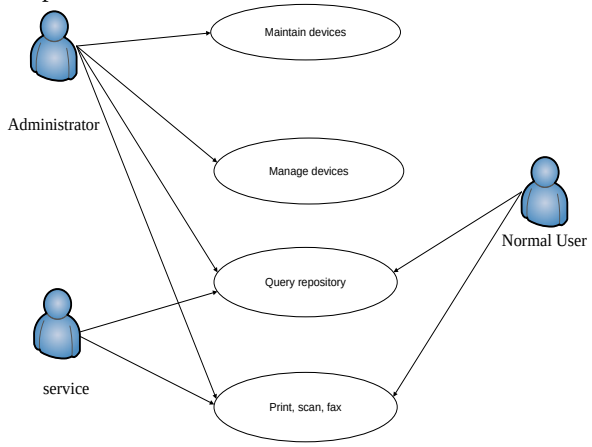

Figure 3: Use case diagram

A. *The administrator*

This role will be give to a person that has basic computer literacy training and is almost always around Dwesa to help out with any problems that might arise with the service. The administrators role will be to perform minor maintenance of the physical peripheral devices, this includes adding more paper to the printer, reboot the devices, changing the ink cartridges, changing toner, reporting hardware failure etc.
On the technical aspect of the service, the responsibilities of the administrator will include manually managing things such as print jobs, by cancelling jobs that are not completing. He\she will also be able to monitor things such as access control and accounting to make sure that the system is not being abused by anyone. The administrator will also be able to add and remove peripheral devices on the system and finally since the administrator is also a member of the Dwesa community he\she will have all the rights that normal users will enjoy.

B. *Normal User*

Community members that wish to use a peripheral devices to accomplish a task qualify as a normal user, these user are going to be the ones that use the system on a regular basis. These users will be able to query the registry for the services that they want to use and their location in relation to the DANs. Once the user has found a service that suits his/her needs and location, they a can use that services either for printing, scanning and or for faxing. The final user of the system will be other services that are running on the Dwesa network and need to utilize the system to accomplish a task

C. *Services*

As mentioned in previous section of this paper that the Dwesa network is evolving to a distributed platform, this is possible because of the adoption of SOA. A few projects have been developed specifically as web services, the remaining eServices are being integrated into SOA [12]. This brings a lot of opportunities for great interconnection and interoperability between services that run on this platform. When the move from isolated eServices to SOA based services is completed, services such as eCommerce, eJudiciary and network monitoring systems can utilize functionally from the service such print. Other web service on the Dwesa platform can print data such as user logs to printers that are located nearest to administrator of that particular service.

VI. Conclusion and Future Work

The paper has shown through supporting literature that there is a paradigm shift in application development and that SOA is a popular architecture for providing secure and distributed applications. This paper has shown that the flexibility and interoperability offered by SOA have made the choice for business to migrate core "Legacy System" to a more distributed environment easy and that helps them to integrate functionality with their partners so as to grow their business [1]. Adopting this strategy, an SOA based resource sharing service is possible and can be used in MRAs to manage access to the limited resources. Leveraging the existence of applications such as device drivers, native printing services on operating system and the availability of API for interacting with peripheral devices, we can easily develop web service wrappers that will allow us to provide access to devices to anyone with an internet connection. Finally the paper has shown that not only is a service such as the discussed here useful in MRAs that have a shortage of these peripheral resources, it can also be used by anyone wanting a service that works with any platform, this platform independency is possible because the service is based on open standards.

There is still some work to be done on the service for it to be reliable more than the onboard drivers, iterative design is continuing to improve and provide more functionality to the system. As part of an ongoing research into delivering SOA applications to MRAs, we are looking into alternatives to SOAP based web services, techniques such as REST web services, which promise to simplify the development of web services.

A possible extension to the system could be to add more devices that are not just for document processing, devices such as security systems (IP cameras, motion detectors, etc). Since the initial deployment of the service is the SLL, another possible future extension for the service could be to localize it for isiXhosa speaking community, since that is the dominant language in that area.

VII. Acknowledgements

This work is undertaken with the Telkom Center of Excellence in ICT for Development at the University of Fort Hare, with funding from Telkom, Tellabs, Saab Grintek Technologies, Easttel and THRIP.

VIII. References

1. Baskerville, R., Cavallari, M., Hjort-Madsen, K., Pries-Heje, J., Sorrentino, M., Virili, F., *Extensible architectures: the strategic value of service oriented*

architecture in banking. In European Conference on Information Systems (ECIS), 2005.
2. Ort, E., *Service-Oriented Architecture Web Services: Concepts, Technologies and Tools*, Available at: http://java.sun.com/developer/technicalArticles/WebServices/soa2/, [Accessed November 2010]
3. Barnes, A., M., and Cilliers, C., *A Holistic Evaluation Strategy for the Modernization of Legacy Systems*. In Southern Africa Telecommunication Networks and Applications Conference, Stellenbosch, 2010.
4. World Wide Web Consortium, *Web Service Architecture* available at: http://www.w3.org/TR/ws-gloss [accessed November 2010]
5. Millard, D.E. *et al, Design Patterns for Wrapping Similar Legacy Systems with Common Services Interfaces.* In 4th European Conference on Web Service (ECOWS), 2006
6. NASCIO, *Digital States At Risk! Modernizing Legacy System.* Available at: http://www.nascio.org/publications/documents/NASCIO-DigitalStatesAtRisk.pdf, [Accessed May 2011]
7. Thinyane, M., Slay, H., Terzoli, A., Clayton, P., *A Preliminary Investigation into the Implementation of ICTs in Marginalized Communities.* In Southern Africa Telecommunication Networks and Applications Conference (SATNAC), Stellenbosch, 2006
8. Ndlovu, N., Jere, N., and Thinyane, M. *Investigating wireless network deployment configurations for marginalized rural areas.* In the 12th Annual Conference on World Wide Web applications-Durban, 2010
9. Shumba, Z., Thinayane, M. *An Investigation of the Viability of Mobile WiMAX in Rural Context*. In Southern Africa Telecommunication Networks and Applications Conference (SATNAC), Stellenbosch, 2010.
10. Scott, M., S., *Investigation and Development of an e-Judiciary Service for Citizen Oriented Judiciary System for Rural Communities.* M.Sc thesis, University of Fort Hare, 2009
11. Samalenge, J., Thinyane, M., *deploying Web Services in Rural Communities for Services of Personal Communication Synchronous and Asynchronous.* In Southern Africa Telecommunication Networks and Applications Conference (SATNAC), Swaziland, 2009
12. Moyo, T., Thinyane, M., *Reengineering Legacy Applications for SOA Middleware Integration*. In Southern Africa Telecommunication Networks and Applications Conference, Stellenbosch, 2010
13. Bangemann, T., *et al, Integration of Automation Devices in Web Service supporting Systems.* In 30th IFAC Workshop on Real-Time Programming and 4th International Workshop on Real-Time Software, pp.161-166
14. Phaithoonbuathong, P., Harrison, R., West, A., Monfared, R., Kirkham, T., *Web services-based automation for the control and monitoring of production systems*, International Journal of Computer Integrated Manufacturing, Vol. 23, No. 2, pp 126-145, 2010
15. da Silva, OA., Meneguzzi, R. F., de Souza Chnieder, P.H., Barcelos, R. B., Rodrigues, V. O., Waldman, D., da Silva, A. C. B., *Providing printing Web services*, IEEE 5th International Workshop on Network Appliances. Liverpool Proceegings, 2002
16. HP, *HP ePrint Enterprise Mobile Printing*. Available at: http://h71028.www7.hp.com/enterprise/us/en/ipg/HPeprint-solution.html, [Accessed March 2011]
17. HP HIT PRINT, Cloud printing in the enterprise: liberating the mobile print experience from cables, operating systems and physical boundaries. [Online] Available at: http://h71028.www7.hp.com/enterprise/downloads/4AA0-8149ENW-June2010.pdf [Accessed; November 2010]
18. Google Code Labs. Google Cloud Print. Available at: http://code.google.com/apis/cloudprint/docs/overview.html, [Accessed March 2011]
19. The chromium Projects. *Chromium Print Proxy*. Available at: http://www.chromium.org/developers/design-documents/google-cloud-print-proxy-design, [Accessed November 2010]
20. Kaplan, B., Maxwell, J., *Evaluating the Organizational Impact of Healthcare Information Systems*, Springer New York, pp.30-55, 2005
21. Budde, R., Kautz, K., Kuhlenkamp, K., Züllighoven H., *What is prototyping?*, Information Technology & People, Vol. 6 Issue: 2/3, pp.89 – 95, 1992
22. Naumann, J., D., Jenkins, A., M., *Prototyping: The New Paradigm for Systems Development.* In MIS Quartely, 1982
23. Forsberg, k., Mooz, H., *The relationship of systems engineering to the project cycle*, in The 12th INTERNET World Congress on Project Management, Oslo, Norway, 1994
24. Matheu, F., *Life cycle document management system for construction.* Universtat Polit, 2005
25. Sibewu, W., Thinyane, M., *Investigating SOA Based Sharing of Peripheral Network Resources.* In Southern Africa Telecommunication Networks and Applications Conference (SATNAC), Stellenbosch, 2010.
26. OASIS, *Introduction to UDDI: Important Features and Functional Concepts*, Available at:http://uddi.org/pubs/uddi-tech-wp.pdf, [Accessed November 2010]
27. Endrei, M., Ang, J., Arsanjani, A., Chua, S., Comte, P., Krogdahl, P., Luo, M., Newling, T., *Patterns: Service-Oriented Architecture and Web Services*, pp 31, IBM corp., International Technical Support Organization, 2004

Wandile Sibewu received his BSc. and BSc. Honours (Computer Science) degrees at the University of Fort Hare and is presently studying towards his Master of Science (Computer Science) degree at the same institution. His research interests include mobile applications and cloud computing.

Mamello Thinyane is a Computer Science Senior Lecturer and Research Coordinator of the Telkom Centre of Excellence at the University of Fort Hare. He holds a Doctoral Degree in Computer Science from the University of Rhodes, Grahamstown – South Africa. His research interests include ICTs in rural development, wireless networking technologies and e-learning solutions