

Protocol Connectivity Invoker Module for Application Integration into SOA Middleware

Sikhumbuzo Ngwenya, Mamello Thinyane
Department of Computer Science
University of Fort Hare, P. Bag x1314, Alice 5700
Tel: +27 406022464, Fax: + 27 406022464
email: {sngwenya, mthinyane}@ufh.ac.za

Alfredo Terzoli
Department of Computer Science
Rhodes University, P. O. Box 94, Grahams town, 6140, South Africa
Tel: +27 466038602, Fax: +27 466361915
email: a.terzoli@ru.ac.za

Abstract – Service Oriented Architecture (SOA) Middleware platforms present a valuable option for extending the lifetime of mission-critical legacy systems. The fundamental concept of SOA is that it allows the design and implementation of software applications through the use of loosely coupled services with known interfaces that can be accessed without the knowledge of the implementation platform. This technique enhances interoperability, extensibility and flexibility of e-services hence it can be an option to improve the effectiveness and efficiency of Information and Communication for Development Projects (ICT4D). This paper presents a proposed connectivity module called Protocol Connectivity Invoker Module (PCIM) which allows the external integration of third party software applications into SOA middleware platforms. The module facilitates connection by invoking relevant connectivity protocols upon requests from third party software applications for communication with the middleware layer.

Index Terms—Application Integration, Connectivity, ICTD sustainability, SOA

I. INTRODUCTION

SOA is a paradigm for organizing and utilizing distributed applications that may be under the control of different ownership domains [1]. Fundamentally, SOA allows the design and implementation of applications using loosely coupled services whose interfaces can be accessed without the knowledge of the implementation platform [1]. Web services are a common form of SOA where interfaces are described using Web Service Definition Language (WSDL), while Simple Object Access Protocol (SOAP) transport data over HTTP, the Universal Description Discovery Integration (UDDI) acts as a directory service [2]. This pattern of application development provides a number of benefits such as improved application interoperability, flexibility and extensibility [2]. SOA also allows the integration of third party applications and the building of future proof ones [3]. As a result this technique can be seen as a mechanism to be adapted to support ICT4D e-service platforms.

II. BACKGROUND INFORMATION AND RELATED WORK

This work is undertaken in the context of ICT4D e-service development and provisioning within the Siyakhula Living Lab (SLL). SLL is an ICT4D initiative undertaken in Dwesa, a rural marginalized community in the Eastern Cape of South Africa [4]. The initial mode of e-services production at SLL has been that of informal development of standalone applications that serves to address specific requirements. For instance, eGovernment applications are developed to instigate and monitor a perpetual interaction between governments and its citizens [5]. This mode of application development pattern limits the application interoperability thereby suppressing potential benefits of ICTs in improving the livelihood of people. Hence an SOA middleware platform known as Teleweaver was designed and implemented to address the issues of application interoperability, efficiency and effectiveness.

In general the introduction of SOA middleware platforms brings about an array of challenges. As a relatively new strategy there is lack of skilled personnel to work on such initiatives. However, the initial disparate applications contain a critical mass of the legacy data as such they cannot easily be discarded. Hence there is need to integrate them into the SOA middleware platforms. Integration of such applications is also a complex issue because they were developed from various patterns using different protocols from different architectures. Hence each pattern would require a unique method of integration. For instance while an MVC application could possibly be integrated either through its View or Model, a procedural application would need to be redeveloped as it might not conform to any interface of the SOA middleware platform.

TeleWeaver is an SOA middleware platform that allows efficient integration of e-services through a myriad of protocols [6]. It uses Equinox container as a service environment which houses loosely coupled OSGI bundles. The functionalities of the individual bundles are exposed as web services through Apache CFX and can be consumed by any calling application that understands web services protocols. A service Invoker component facilitates this

interaction between the bundles by recording the properties of all published bundles in the container. Any external component which wants to communicate with bundles in the container interacts with the Service Invoker which has all the details of the present bundles. At present third party applications communicate with TeleWeaver through SOAP messages. One of the motives behind the use of TeleWeaver was to integrate third party applications and for that matter we would need a connectivity platform which caters for all kinds of applications with their associated connectivity protocols and the current design of TeleWeaver does not cater for this provision. This is a challenge which hampers the efforts of simplification of application interoperability hence we proposed the use of Protocol Connectivity Invoker Module (PCIM), which is an extension to the current service invoker, to address this challenge.

III. THE CONNECTIVITY MODEL

The fundamental concept of PCIM is that it should act as a protocol identifier which associates the relevant protocols to the calling applications. All possible connectivity protocols such as Extensible Markup Language –Remote Procedure Call (XML-RPC), HTTP, Representational State Transfer (REST) and Simple Object Access Protocol (SOAP) will be housed in this module which has other core components that regulates the handling of protocol translation and dispatching processes [7]. The challenge with this proposal is that if PCIM were to be located outside TeleWeaver it would need its own service environment thereby creating an unnecessary heterogeneous of execution environments. In light of this the PCIM should be housed within Teleweaver so as to simplify the complexity of connectivity. Figure 1 depicts the envisaged connectivity system.

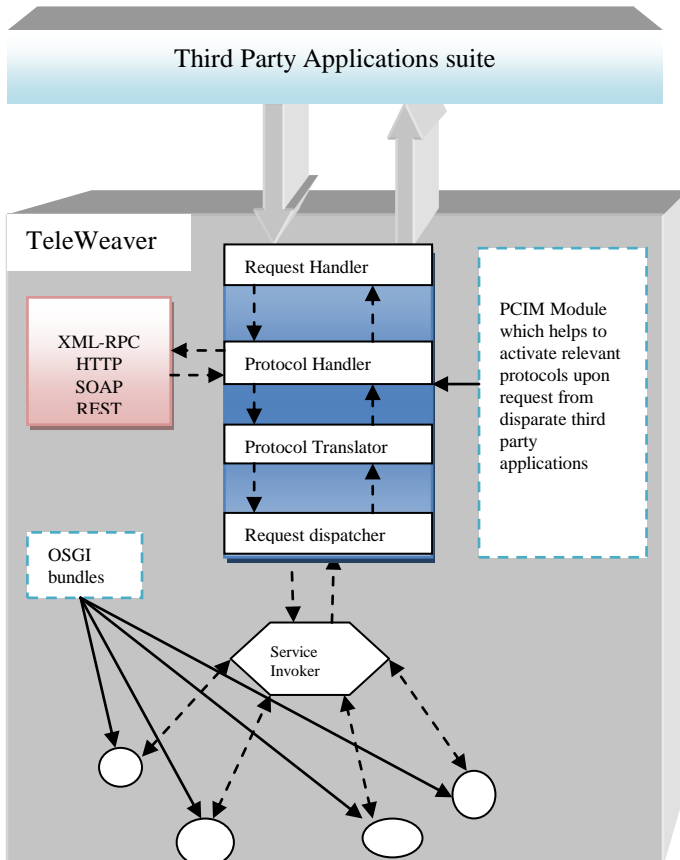


Figure 1: PCIM Connectivity diagram

The use of such a module is anticipated to provide a generic method of connecting various third party software applications on to the SOA middleware thereby increasing application interoperability and extensibility.

IV. CONCLUSION

This paper has presented an application connectivity model which enables the integration of third party software applications onto SOA middleware platforms. One of the challenges in this model is that of compatibility between connectivity protocols. Some protocols like REST are stateless as such it would not interface easily with other stateful protocols hence we suggest the introduction of a translation module within the PCIM. Currently each request would need its own service invoker within TeleWeaver hence it is felt that PCIM should reside inside the Service Invoker to create a Super Service Invoker which performs service invoking and the PCIM processes in a single platform.

V. REFERENCES

- [1] Canfora, G., Fasolino, A. R., Frattolillo, G., & Tramontana, P. (2008). A wrapping approach for migrating legacy system interactive functionalities to Service Oriented Architectures. *Systems and Software*. 4 (81), 463--480.
- [2] David, C. & Tyler, J. (2010). *Java Web Services*, O'Reilly. 72--95, USA.
- [3] Schroth, C. & Janner, T.. (2007). Web 2.0 and SOA: Converging concepts enabling the internet of services. *IT professional*. 9 (3), 36--41.
- [4] Pade-Khene, C., Palmer, R. & Kavhai, M. (2010). A baseline study of a Dwesa rural community for the Siyakhula Information and Communication Technology for Development project: understanding the reality on the ground. *Information Development*. 26 (4), 265--288.
- [5] Gupta, M. P. & Jana, D. (2003). E-government evaluation: A framework and case study. *Government Information Quarterly*. 20 (4), 365--387.
- [6] Ntshinga, L., Thinyane, M. & Terzoli, A.(2011). Blueprints for Software Components within Teleweaver, a Custom OSGi Container. *In the Proceedings of the 14th Southern Africa Telecommunication Networks and Applications Conference (SATNAC) 2011*, East London, South Africa.
- [7] Laurent, S.S., Johnston, J. & Dumbill, E(2001). Programming web services with *XML-RPC*, O'Reilly Media.

Acknowledgements: This work is undertaken within the Telkom centre of Excellence in ICT for development at the University of Fort Hare with funding from Telkom, Tellabs, Saab Grintek Technologies, Easttel, Khula Holdings and THRIP which is managed by the National Research Foundation.

Sikhumbuzo Ngwenya received his Masters of Science degree in 2010 from the University of Fort Hare and is presently studying towards his PhD degree in Computer Science degree at the same institution. His research interests include the development of eService applications to enhance Rural Community Development through ICTD initiatives.