

Towards Realization of Web Services-Based TSA from NGOSS TNA*

Mi-Jung Choi¹, Hong-Taek Ju², James W.K. Hong³, and Dong-Sik Yun⁴

¹ School of Computer Science, University of Waterloo, Canada
mjchoi@cs.uwaterloo.ca

² Dept. of Computer Science, Kyemyoung University, Korea
juht@kmu.ac.kr

³ Dept. of Computer Science and Engineering, POSTECH, Korea
jwkhong@postech.ac.kr

⁴ KT Network Technologies Labs, KT, Korea
dsyun@kt.co.kr

Abstract. To avoid frequent changes of OSS's architecture, TNA provides NGOSS architecture in technology-neutral manner. TNA can be mapped to appropriate TSAs using specific technologies such as XML, Java and CORBA. Web Service can be applied for NGOSS TSA. In this paper, we examine architectural principles of TNA and propose an application mechanism of Web services technologies to TNA.

1 Introduction

As the data communications, telecommunication and other types of communications converge, the complexity and heterogeneity of networks and provided services are rapidly increasing. So, future Operation Support Systems (OSSs) need to cope with these incremental changes and complexity of networks and services. To meet this requirement, TeleManagement Forum (TMF) provides a Next Generation Operations Systems and Software (NGOSS) framework for rapid and flexible integration of operation and business support systems [1].

The network services and resources should be easily modified by appropriate business goals and policies. As new network services emerge, the operation management system also needs to be newly developed and use up-to-date technologies. To avoid frequent changes and upgrades of OSS's architecture, TMF provides NGOSS architecture in technology-neutral manner. Technology-neutral architecture (TNA) [1] is the basic concept and component of NGOSS architecture. This TNA can be mapped to appropriate technology-specific architectures (TSAs) and this mapping can leverage industrial standard frameworks such as service-oriented architecture (SOA), component-based architecture and distributed computing.

* This research was supported in part by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment) (IITA-2005-C1090-0501-0018) and by the Electrical and Computer Engineering Division at POSTECH under the BK21 program of the Ministry of Education, Korea.

Web service is one possible technology for NGOSS architecture. Web service is an emerging technology, which is being standardized continuously. Web service is also distributed and services-oriented computing technology with strong support from the industry. Therefore, the mapping of NGOSS TNA to Web services-based TSA is a promising research area.

The 'realization' in this paper means that how essential parts of the NGOSS architecture can be implemented using a specific standards-based COTS technology. We choose Web service as the specific technology. In this paper, we first examine NGOSS TNA in the perspective of the concepts, requirements and components. Then, we propose a technology specific architecture using Web services technology.

2 Architecture

In this section, we describe the NGOSS TNA and the architecture of the current KT's OSS [2]. The difference in the architecture is also examined.

2.1 NGOSS TNA

The insulation of system architecture from technology details provides a number of related benefits [5]: First, it ensures the validity of the NGOSS architecture over time by supporting the deployment of new technologies without having to re-architect the entire OSS solution. Second, it provides the architectural underpinnings for the simultaneous use of multiple technologies in a single integrated OSS environment, supporting legacy systems and enabling technology migration over time. Finally, insisting that the architecture remain technology neutral helps to prevent system

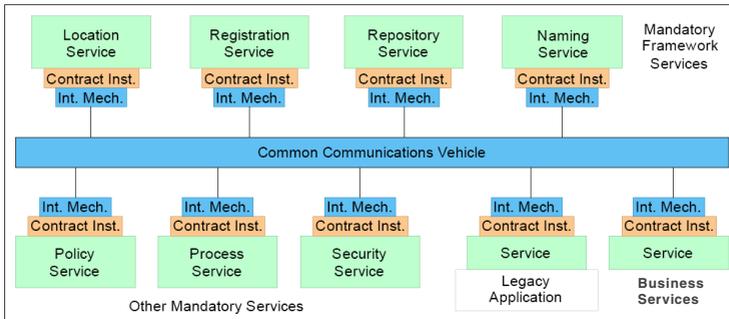


Fig. 1. This shows the detailed views of NGOSS technology-neutral architecture [1]. The service modules can communicate with each other through common communications vehicle (CCV) [1]. CCV is a kind of message bus independent of technology. The services are divided into mainly two parts: business services and framework services [3]. Business services provide the application level functionality that directly supports the implementation of a business process such as SLA management, billing mediation, QoS and etc. Framework services provide the infrastructure necessary to support the distributed nature of the NGOSS TNA. Contract [4] is the central concept of interoperability. Contract Instance is a runtime manifestation of an NGOSS contract implementation that provides one or more functions to other runtime entities. It represents the binding unit.

design decisions being taken too early in the development lifecycle, so the architecture is protected from over-specification. An excess of design detail in the core of the architecture would make it more difficult to find technologies with which to implement it.

2.2 KT NeOSS

Currently, KT has developed and used New Operation Support System (NeOSS) [2]. NeOSS is an integrated OSS platform for all KT services. But, NeOSS does not fully support NGOSS’s architecture. NeOSS is more focused on NGOSS’s business process framework such as enhanced telecom operations map (eTOM) and shared information data (SID).

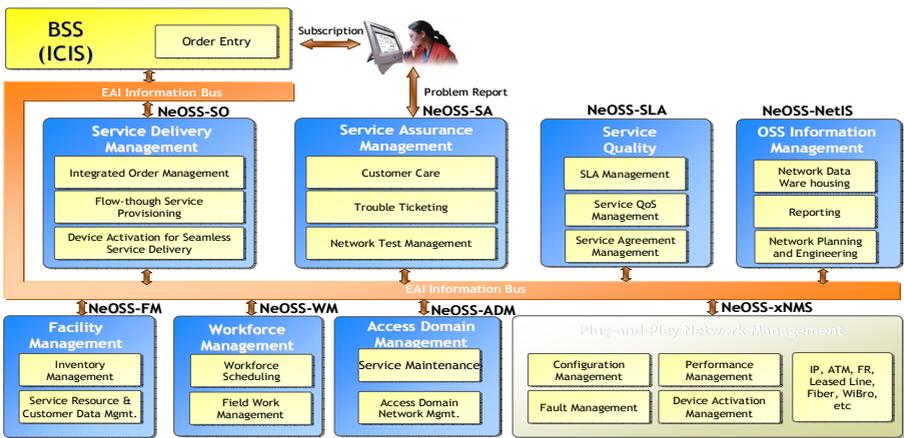


Fig. 2. This shows the architecture and functions of NeOSS. The BSS system receives the service order and requests from customers and delivers the information to correspondent modules. EAI information bus transfers messages between each management function module. NeOSS function modules are NeOSS-SO (Service Order), SA (Service Assurance), SLA (Service Level Agreement), NetIS (Network Information Support System), FM (Facility Management), and etc. The yellow box is the detailed function of each management module. For example, NeOSS-SA is a module that integrates the fault management with the concept of flow-through processing. When it gets a notification of a trouble ticket, it dispatches relative information to the affected services in order to correct the problem.

The difference between NGOSS TNA and NeOSS’ architecture is as follows. EAI information bus of NeOSS plays a role of CCV of TNA. However, NeOSS does not have the concept of pre- and post-conditions. NeOSS does not have the concepts of maintaining invariant attributes and behavior, another crucial concepts of contracts. Therefore, there is an interface between management function modules in NeOSS. Moreover, NeOSS includes management services such as SLA, SA, SO, FM and etc. and these services provide management functions for the business process. But, NeOSS does not include framework services of NGOSS TNA, so it does not support the distributed nature of TNA.

3 Alignment of NGOSS TNA

In this section, we present a method to how the technology maps to the concepts of the NGOSS TNA and additionally describe detailed mappings of the technology-neutral shared information and contract models. The core architectural principles to be examined for alignment of NGOSS TNA and TSA are below.

First, the **architecture** needs to provide the distribution support. The architecture provides for communication between business services and between process control and other system services. This entails support for location independence, distributed transactions and other aspects of component/service interaction in a distributed systems environment. The architecture does not stipulate the precise mechanism for communication with services or any inter-service protocol, other than the technology-neutral methods for conveying business semantics and content between participating entities.

Second, the **architecture** should support the separation of business process from software implementation. The decoupling of business process from service implementation requires that the interface and semantics of services be rigorously defined in a technology-neutral way. This is achieved by the NGOSS “Contract”- a mechanism for formally defining the business semantics of a distributed service. A contract is initially defined in technology-neutral form, expressing the business functionality of the service. Technology-specific information required for service invocation is added later, and includes such information as the supported communications protocols and invocation patterns. The NGOSS contract registry provides support for run-time discovery of services based on Contracts.

Finally, the **OSS** needs to provide the shared information for management information model. NGOSS stipulates that any business information that needs to be shared between services should be considered the property of the enterprise as a whole and not of any particular application or component. It uses the concept of shared information, which describes all shared business information in a specific NGOSS deployment. Information services coordinate access to information through well-defined interfaces and are responsible for information integrity and for managing information stewardship, where specific entities in the system have transient responsibility for a shared information entity. To design technology-specific architecture, the technology must align with these NGOSS architectural principles.

TMF proposed two **technologies** of XML [5] and CORBA [6] for NGOSS TNA and specified technology application notes with these technologies. OSS/J initiative proposed Java technology as a specific technology for TNA [7]. XML has a natural affinity with communications management. The use of XML to validate, manipulate and define the structure of application specific information models using general-purpose tools becomes an attractive possibility. However, XML has a substantial overhead associated with the text-only encoding of data. Also, XML/HTTP solutions (e.g. SOAP) suffer as yet from the lack of availability of common distributed processing support services provided by more mature platforms, e.g. CORBA and J2EE services [5].

CORBA is a **distributed** processing platform. Therefore, CORBA supports communication method and framework services for distributed processing. Also, CORBA supports interface definition with specifying CORBA IDLs. But, CORBA

does not provide information modeling, so CORBA can define shared information using XML or other languages [6]. J2EE directly implements the principles of NGOSS TNA such as distribution support and separation of business process from implementation. But, J2EE architecture does not have any explicit support for the concept of shared information or federated information services as defined in NGOSS TNA [7].

4 Our Approach Using Web Services

Web services are very complex and include a lot of standard specifications. At this moment, the standardization efforts are still going on. So, it is not easy to specify the appropriate Web services technologies applicable to NGOSS architecture. We more focus on the NGOSS specification from a Web services perspective. Our to-be architecture is extended from this TNA architecture using Web services technologies.

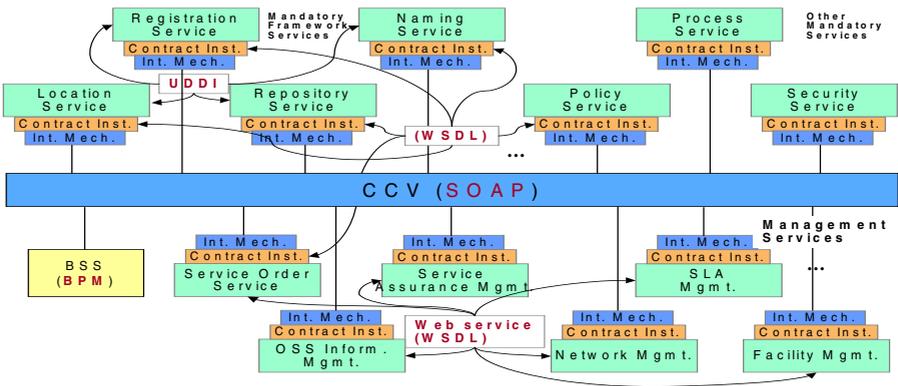


Fig. 3. This shows our proposed technology-specific architecture. XML is used for the definition and encoding of messages in data communication protocols. SOAP is a protocol that can be used as a CCV-based solution. Web Service Definition Language (WSDL) can be used to define contracts between process entities through SOAP. UDDI provides a comprehensive mechanism for locating services at run time by storing service interfaces in their WSDL format. That is, UDDI supports framework services. Application programs can search the UDDI repository to find the interface that they require, download the WSDL description and use the binding information to communicate with the interface over a suitable communication channel. Other framework services and management operation services such as SLA, NM and SA can be defined as new services using WSDL. Web Services technologies can be applied in a number of areas to assist with the management of business processes. Web Services Business Process Execution Language (WS-BPEL) can define business process and the BPEL engine handles the process sequentially.

5 Concluding Remarks

We examined the concept, architectural principles and components of NGOSS TNA. We proposed our technology-specific architecture using Web services. Our work is

the early stage, so we need to concrete our TSA with more knowledge of Web services technology.

To validate our proposed architecture, we will implement a prototype of Web services-based TSA. Then, we will implement one of management operation services of NGOSS such as QoS management or SLA and perform a test on our implementation system. We will apply our proposed architecture to the architecture of KT's NeOSS. Finally, we will extract performance metrics of Web services-based TSA and conduct performance analysis.

References

1. TMF, "The NGOSS Technology-Neutral Architecture", TMF053, Version 5.3, Nov. 2005
2. KT, "NeOSS (New Operations Support System)", <http://www.kt.co.kr>
3. TMF, "NGOSS Architecture Technology Neutral Specification - Distributed Transparency Framework Services", TMF 053F, Version 1.0, Jan. 2004
4. TMF. "NGOSS Architecture Technology Neutral Specification - Contract Description: Business and System Views", TMF 053B, Version 4.4, Feb. 2004
5. TMF, "NGOSS Phase 1 Technology Application Note - XML", TMF057, Version 1.5, Dec. 2001
6. TMF, "NGOSS Phase 1 Technology Application Note - CORBA", TMF055, Version 1.5, Aug. 2001
7. C. Ashford, "OSS through Java as an Implementation of NGOSS", White paper, Apr. 2004