CONSIDERING CONTRACTS FOR GOVERNANCE IN SERVICE-ORIENTED ARCHITECTURES

Jacqui Chetty[1] and Marijke Coetzee[2]

[1]Department of Business Information Technology
[2]The Academy for Information Technology
University of Johannesburg

[1]jacquic@uj.ac.za
(011) 559 1177
[2]marijkec@uj.ac.za
(011) 559 2907

ABSTRACT
Service Oriented Architecture (SOA) is a design paradigm that enables applications to be built from business processes. Services, service-orientation and related technology give organisations the ability to gain a competitive edge. This however, does not come without cost, due to the fact that organisations develop services quickly, very often without much thought to their management and maintenance. SOA governance is considered a subset of IT governance, to control the design and execution of services. Governance is a multifaceted concept and is addressed at strategic, operational and technical levels. The focus of vendor driven approaches to SOA governance currently is at the technical level, mainly to control the life-cycle of services and their associated policies.

To gain an insight into this level of SOA governance, this research investigates vendor approaches. In order to identify deficiencies in current approaches, the SOA reference model from OASIS is also used to identify components that ideally need to be governed. From this comparison, additional aspects are identified that need to be addressed by SOA governance. It becomes clear that the governance of service execution is critical in ensuring effective service-oriented architectures. This is particularly prevalent in aspects like security, where actions cannot be ambiguous as they are likely to affect the service execution outcome. This research proceeds to identify service contracts and the enforcement thereof as a means to comprehensively govern service interaction. The paper finally proposes a high-level contract management framework.

KEY WORDS

Service Oriented Architecture, governance, SOA reference model, service contract

CONSIDERING CONTRACTS FOR GOVERNANCE IN SERVICE-ORIENTED ARCHITECTURES

1 INTRODUCTION

Service Oriented Architecture (SOA) (Brown, et al. 2006) is a paradigm for organising and utilising distributed capabilities that may be under the control of different ownership domains, and implemented using a variety of technology stacks. SOA is a holistic approach to designing systems in a distributed environment, where integration is mandatory. Organisations gain a competitive edge by exposing their capabilities or business functions as services, to be re-used for different applications and purposes. Services are well-defined, self-contained, and do not depend on the context or state of other services. Even though SOA is protocol independent, web services technology (Graham, et al. 2002) is becoming the most common implementation of SOA.

SOAs develop in an evolutionary manner, and are different for each organisation and even for each department in an organisation. While most organisations commence their SOA drive with a pilot project, they quickly begin initiatives that span multiple departments or business organisations. Left ungoverned, an SOA could allow anyone to deploy a new service, or invoke and orchestrate any other combination of services. SOA governance is consequently introduced to manage and control the increasing number of services, in order to ensure reuse and consistency, and avoid duplication of work.

Typically, SOAs are governed via the integration of a variety of vendor-oriented governance solutions. These solutions can introduce problems like the violation of principles of SOA. SOA governance should ideally ensure that services are controlled, that they behave in the way they should, and are not misused. In order to assess whether a given service is behaving as it should when it is invoked, service contracts can be used to establish the reference points for monitoring and reporting by the SOA execution environment. By tracking the actual performance of a service and comparing it to the requirements specified in the service contract, non-compliant services can be identified and timely remedial action can be taken.

This paper considers service contracts as an important element of SOA governance. Section 2 provides a background on SOA governance. Section 3 discusses SOA governance technologies to identify deficiencies in current approaches. To identify additional aspects to be governed by SOA governance, section 4 evaluates the SOA reference model from OASIS. This evaluation identifies that for instance, security policies may be applied ambiguously when services interact. To address this concern, section 5 defines the concept of the service contract. Next, a high-level governance framework is introduced that places a central focus on the use of service contracts for governance. Finally, the paper is concluded.

2 SOA GOVERNANCE

Governance is a set of processes, policies, behaviours, laws and institutions (Von Solms & Von Solms, 2006). These entities influence the approach of developing organisation strategies and objectives into a framework that consists of directives, policies, standards and procedures; implementing this framework operationally; and incorporating metrics that measure the level of compliance regarding the framework. New rules introduced by governance frameworks are forcing companies to rethink how they govern their IT processes. For instance, governance may require of publicly registered companies to show the effectiveness of their internal control structures and reporting procedures. This means organisations need to both control and validate human-to-machine, as well as machine-to-machine service-based interactions.

IT governance is considered a subset of governance (Carter, 2007). It is a framework that consists of processes, organisational structures and leadership, to ensure that the IT systems of an organisation align itself with the strategies and objectives of the organisation (Von Solms & Von Solms, 2006). In its turn, SOA governance can be seen as an extension of IT governance (Carter, 2007).

The term is commonly used to refer to the technology associated with SOA infrastructure such as web service management and security tools, and different Universal Description, Discovery and Integration (UDDI) (Clement, et al. 2004) implementations. The aim of SOA governance is to *control*, *enforce* and *monitor* services throughout their life-cycle (Marks & Bell, 2006), thus ensuring that the services can be reused in an accountable manner across domains of control. SOA governance includes all aspects of IT governance, and also special relationships, policies, and processes to address unique SOA aspects and artefacts. SOA governance is addressed by strategic and operational considerations, and technical mechanisms (Erl, 2008). As governance is a multifaceted concept that is addressed from various angles, the focus of this research is placed at the technical level, to provide metrics and measurements to assist governance decision-making.

The focus of this level of governance is to ensure that services are controlled, behave in the way they should, and are not misused. If a service is designed for a specific purpose and set of consumers, audit logs can for instance prove that the service behaved correctly when the service was invoked. Services should also be available, perform as intended, and be secure. Services that do not comply with these requirements are not governed, and they will inevitably be misused, become unreliable and insecure.

A first aspect of SOA governance to be investigated is the artifacts that are to be governed. In order to address this, the focus is now placed on vendor-oriented SOA governance. The next section provides an overview regarding approaches that are followed, and the artefacts governed by AmberPoint (Amberpoint, 2008), IBM (IBM, 2008) and HP (Systinet, 2008).

3 VENDOR-ORIENTED SOA GOVERNANCE

The vendors approach SOA governance by explicitly addressing visibility, control and trust (Amberpoint, 2008; IBM, 2008; Systinet, 2008). Table 1 consists of a column for each of the three vendors that provide a basic view of their approach to SOA governance. From their perspective, visibility ensures that services are monitored across the SOA lifecycle and that business flows are tracked to assess the business impact. Control is seen as ensuring that systems deliver a level of quality of service (QoS) that is expected from them within rules and regulations. If consumers are assured of the quality, predictability and transparency of terms and conditions of services, they can trust such services. A main focus of SOA governance is thus to guarantee trustworthy services that can be reused with a high level of assurance. Services need to be managed at design-, run-, change-, and life-time cycle (Marks & Bell, 2006). In order to achieve this, governance vendors employ mechanisms such as registries, repositories, policies, and lifecycle management. Each of theses mechanism is now briefly discussed to highlight the role that they play in governance.

Table 1: Vendors' approach to SOA governance

	AmberPoint	**IBM**	**HP Systinet**
APPROACH	Visibility, Control	Visibility, Control	Visibility, Control , Trust
REGISTRIES	Registry/repository integrated for interoperability	Registry/repository integrated for interoperability	Registry/repository integrated for interoperability
REPOSITORIES	Used during development		
POLICIES			
High-level Man.	Yes	Yes	Yes
3-tiers	Yes	Yes	Yes
Description	WSDL, WS-Policy	WSDL, WS-Policy	WSDL, WS-Policy
Basic interaction	SOAP, WS-Addressing, WS-Notification	SOAP, WS-Addressing, WS-Notification	SOAP, WS-Addressing
Security	WS-Policy, WS-Security, No mention	WS-Policy, WS-Security, WS-Secure Conversion	WS-Policy, WS-Security, WS-Secure Conversion
Reliability	WS-Reliability	WS-ReliableMessaging	WS-ReliableMessaging
Trust	WS-Trust, WS-Federation	WS-Trust, WS-Federation	WS-Trust, WS-Federation
SLA's	WSLA (web service level agreement)	WSLA (web service level agreement)	
Composition	WS-BPEL	WS-BPEL	WS-BPEL
LIFECYCLE MANAGEMENT	Development → Staging → Production	Plan → Define → Enable → Measure	

Registries: The registry is the first and foremost enabling technology for SOA governance. It is a dynamic record of the SOA environment that is used to control and monitor services. A registry holds metadata about services such as its history, who is allowed to make changes to it, who has access to it and how it can be used. A registry that has become an industry standard is Universal Description, Discovery and Integration (UDDI) (Alencar, et al. 2003).

Repositories: Repositories govern the life cycle of services to ensure that a service's records are kept at all stages of its lifecycle. The repository keeps a record of all source code that the organisation develops and provides an audit trail of any previous versions, therefore controlling and monitoring services. The repository can be a separate element or form part of a registry.

Policies: Policies govern the behavior of a service by supplying the rules and constraints that a service needs for successful interaction (Erl, 2006). These characteristics include behavior, preferences, technical limitations and quality of service. Rules and constraints are machine-to-machine specifications that are expressed programmatically as assertions and grouped into various combinations (Erl, 2008). Table 1 illustrates that policies are dealt with at different organisational tiers such as management, architectural and technical. Policy management systems ensure that policies comply with organisational standards, are visible and that they are associated with services. Vendors support the definition of a variety of policies to address aspects such as security, trust, reliability, service-level agreements and composition, as shown in table 1.

Lifecycle management: SOA Lifecycle Management assists with governance by monitoring and controlling policies and processes across the complete SOA lifecycle (Marks & Bell, 2006). It ensures that any changes to a service is monitored and controlled to ensure that the quality of the service remains consistent. Without it, policies may be violated, which may result in noncompliant inefficient services.

By using the abovementioned mechanisms to implement SOA Governance, developers thus have visibility to available services. With a registry and/or a repository in place, they are able to get detailed information about services by means of their metadata attributes that detail all aspects of the service. In addition, by managing all services aspects in a central location, lifecycle management, change management and impact analysis are facilitated.

An analysis of the table and vendors' approaches identifies that:

- Vendors generally approach SOA governance from their own perspective. Some follow a registry-based approach to control services and policies, and others govern the execution of service interaction. Organisations attempting to address governance comprehensively thus need to integrate various tools to be able to do so.
- Policies are the key element to vendor approaches. Various different types of policies are defined in machine-readable syntax, and are automatically associated with services to control service interaction.
- Vendors support interoperability by adhering to WS specifications.
- Organisations implementing service-oriented systems can be locked into the approach of a specific vendor. This may be to the detriment of the implementation of service-oriented principles such as loose coupling and composability.

In order to further identify SOA aspects and artefacts that need to be governed, the following section investigates the SOA Reference Model. This may identify additional elements that need to be employed to strengthen governance.

4 SOA REFERENCE MODEL

The SOA Reference Model (RM), based on the OASIS SOA RM v1.0 (Brown, et al. 2006) is an abstract framework that focuses on describing services, and the significant relationships and key concepts between them. Key concepts related to the SOA Reference Model namely, visibility, interaction and real world effect are described next.

Visibility: Visibility is when a service consumer has a description of the service and the necessary rules that apply to the service, available to them.

Interaction: Interaction is characterized by actions that occur from passing information between services in the form of messages, or by altering the state of a shared resource. The structure and semantics of exchanged messages is described by an Information Model. A Behaviour Model gives an understanding of service actions, responses, and temporal dependencies between actions on the service. The essence of interaction is grounded in a particular execution context.

Execution context is the agreed upon elements and conditions under which interaction can take place (Brown, et al. 2006) within a specific instantiation of a service (Estes, et al. 2006). Different instances of the same service thus have different execution contexts. The execution context may also evolve during a service interaction. The outcome of execution context is either a change of state or the exchange of information. This is referred to as the real world effect.

Real World Effect: The real world effect is a change of state that has occurred by services participating in the exchange of messages.

To gain an understanding of these concepts, consider the following example: There exists a ServiceA, whose service description is made available to others. Its associated policy, Policy1 is also available to service consumers. *Visibility* is thus an aspect that is addressed by current SOA governance technology through for instance, registries.

Furthermore, Policy1 contains 2 rules. Rule 1 states that if the service consumer is internal to the organisation, a username/password parameter is sufficient, but no QoS guarantees are applicable; alternatively, rule 2 states that if the service consumer is external to the organisation, a certificate must be presented, credit card details will be encrypted and QoS guarantees are applicable. Policy1 governs the interactions of ServiceA with its consumers, but may also be applicable to many other services. It is now possible that a service consumer, external to the organisation, supplies a username/password, and is granted access to ServiceA unintentionally. This happens because the service consumer has not agreed to a service contract, and is choosing to follow rule 1. Consequently, in this *interaction*, ServiceA may be improperly used and successive service interactions containing credit card details may be unencrypted. This highlights the fact that a policy may be applied improperly, as the consumer has not agreed to use rule 2, and the execution context of ServiceA may differ from one instantiation to the next. It may also differ for different types of service consumers. If this interaction is not actively monitored, the fact that policies are not properly applied may go unnoticed, and the interaction is not adequately controlled. Current SOA governance technology does not sufficiently address this problem.

Finally, the *change of state* occurs for example if ServiceA is accessed to reserve a seat on a flight. This results in ServiceA reserving a seat and receiving money, and the service consumer receiving a reserved seat in exchange for money. To ensure that proper governance of service interaction takes place, the change in state also needs to be monitored. If governance of the visibility, interaction, and change in state is not performed, the result is that a service can be misused; timely remedial action does not occur; or an invalid change of state has occurred. This highlights the following:

- Policies are not agreed to by service consumers, can be applied ambiguously by enforcement points, leading to an improper change in state.
- Different instances of the same service have different execution contexts.
- The execution context of a service interaction needs to be actively monitored.

Using policies for governance cannot prevent this situation from occurring. The next section introduces the service contract, to identify the role that it may play to strengthen SOA Governance.

5 SERVICE CONTRACTS

Service contracts form the foundation for communication between services and therefore represent the most fundamental architectural element of an SOA (Erl, 2008). It supports the relationship between a service and its consumer, and can assist to establish an agreement, and maintain trust between parties. It is not required for the agreement to be entered into legally or to be explicitly negotiated (OASIS SOA Reference Model Technical Committee, 2006; Jencmen & Yehudai, 2006).

Service contracts are typically unique to a specific service/consumer relationship. It contains formal policies, as well as agreements that are unique to the parties. Furthermore, only semantic information that the organisation wants to make public forms part of the contract (Erl, 2008). A service contract is said to be in place when a valid interaction has taken place (OASIS SOA Reference Model Technical Committee, 2006). Because consumers may vary, there may be multiple service contracts for a single service. SOA governance consequently becomes a process that produces services with a service contract that can be trusted.

Different combinations of policies, applicable to a service, are attached to its service contract. A policy combination that suits given parties is chosen, and the said parties are in agreement regarding the chosen policy combination. Next, a definition of both a policy and service contract is given to distinguish between these concepts.

Policy: A policy is the rules and constraints that govern different aspects of service interaction such as security or reliability. It can be applied to any number of contracts. Examples of policy statements include:

▪All interactions with services must be secured with SSL.

▪All users should be authenticated with encrypted passwords.

▪The service should be available 95% of time.

Service contract: A service contract provides a precise and unambiguous agreement as to how a service and its consumer will interact. It provides a formal definition of the functional and non-functional aspects of the service. The functional aspects include the service endpoint, service operations, input and output messages supported by each operation, and the data representation model of each message's content. The non-functional aspects include the rules and constraints that govern the interaction of service operations. It can also include higher business-level characteristics that are not fundamental to the service interaction, such as legal requirements. A service contract thus consists of various types of policy statements that can be considered as the clauses of the service contract.

Current standards and technology is widely available to support basic forms of service contracts. For web services, a service contract is collectively viewed as the technical service description defined by WSDL (Christensen, et al. 2001), XSD schemas (Davidson, et al. 1999) and a set of policy documents. Specifications that are used to define policy documents include WS-Policy (Bajaj, et al. 2006), which is used as a container for specifying a range of policy considerations. Specifications such as WS-Security (Hallam-Baker, et al. 2006), Web Services Business Process Execution Language (WS-BPEL) (Alves, et al. 2007), Web Service Level Agreements (WSLA) (Dan, et al. 2003) and Web Service Offerings Language (WSOL) are used to specify a variety of non-functional requirements. Future developments such as the Ontology Web Language for Services (OWL-S) (Burstein, et al. 2004) aim to provide a better language for defining service contracts.

The following section describes a framework for SOA governance that centrally positions service contracts in its approach.

6 GOVERNANCE-BY-CONTRACT FRAMEWORK

As stated, the aim of SOA governance is to *control*, *enforce* and *monitor* services throughout their life-cycle, ensuring that they can be reused in an accountable manner and across domains of control. To address this, the framework for governance-by-contract consists of two phases. The first phase addresses *control* by service contract design, and the second, *enforce* and *monitor* by the enablement of governance-by-contract. The framework does not aim to replace current SOA governance technology, but rather aims to define an approach to using such technology. The main focus of the framework is to address the role that service contracts can play to strengthen SOA governance. The first phase to be addressed is service contract design.

6.1 Service contract design

For governance, *control* means to ensure that adequate measures are in place to provide assurance that objectives will be achieved and undesirable events will be prevented or detected and corrected (IT Governance Institute, 2007). For SOA governance this means creating, implementing and managing policies and service contracts to provide rules and constraints for a service and its consumers to follow. Also, because the service contract is shared amongst service consumers, its design is particularly important. Service consumers agreeing to the service contract become dependent on its definition. Therefore, service contracts need to be carefully designed, maintained and versioned after their initial release.

Service contracts, designed with a view on service governance should be created as follows:

- Standardise the vocabulary that will be used to describe policies and service contracts.
- Design the functional interface of the service.
- Design the non-functional requirements of the service such as security, reliability, or service-level agreements. This process should formally consider governance frameworks that the organisation complies with such as Cobit (IT Governance Institute, 2007).
- Identify each possible execution context required for a service interaction. Service consumer or group of consumers may require different levels of, for instance, service-level agreements or security.
- Identify policies required by each execution context of a service.
- Associate policies to the service contract for a specific execution context.

Because a service contract is specific to the interaction between a consumer and the service, it can establish reference points for monitoring and tracking whether or not service consumers are abiding by the requirements specified in the service contract. Therefore, designing a service contract with governance in mind will strengthen the governance process.

Furthermore, the framework is based on the notion of varying levels of service contracts. For example, the service contract of a service that provides weather reports for portal applications do not need a high level of governance, but that same service may need strict governance if it is being used by a military system. The weather service used by portal applications interacts with a basic service contract that consists of functional specifications. Consequently, a low level of SOA governance needs to be implemented. For instance, the visibility of the service can be ensured through a registry. On the other hand, the weather service used by a military system needs to be protected by associating information security policies to its service contract. In this case, governance of the service execution is required to ensure that rules and constraints attached to the service contract are properly applied. As more non-functional aspects are added to a service contract, the required degree of governance thus increases.

Policies are applied to each service contract according to the non-functional requirements of the service. Previous research identified that service contracts can be structured according to such aspects (Jencmen, 2006; Cubera, 2007). The framework now proposes that service contracts are structured according to three high-level categories, namely:

- *Basic:* A basic contract addresses the functional aspects of a service such as how to locate the service and what the service is about. Such a contract is used when a service has minimum requirements with respect to governance, as it has little impact on the performance of the organisation.
- *QoS:* A QoS service contract addresses non-functional aspects such as security, reliable delivery, and performance. There are a variety of QoS aspects that can be included to increase the quality of the service. Services with such requirements have a significant impact on the performance of the organisation and need to be measured to ensure that they meet their requirements. These types of service contracts differ for service consumers and may be negotiable.
- *Behavioural:* To consider the dependencies between the functions provided by the service, the behavioural service contract defines the expected behaviour of

a service participating in a conversation with others. The conversation can be an orchestration or a composition of services. This contract includes requirements to ensure that a service will behave appropriately in a sequential context. As conversations take place across different domains, governance of these aspects is vital to maintain trust between a service and its consumer.

Establishing levels of service contracts to assist with governance is a challenge that will require significant attention in the future. Service contracts cannot be developed in isolation, but their development must be guided by current governance frameworks. The next paragraph addresses the second phase of governance-by-contract.

6.2 Enablement of governance-by-contract

The quality of service execution can be seen as a reflection on the level of SOA governance. If the health of a service degrades during service execution, the consumer is directly affected. To ensure the health of a service, service contract clauses are *enforced* and *monitored* by applicable enforcement and governance points.

Enforce means to compel components to abide by the rules and constraints (Hawkins, 1995). For SOA governance this means implementing mechanisms to coerce a service and its consumers to abide by the rules and constraints of service contracts. This means to implement the logic for the various governance aspects such as enforcement of encryption requirements, exceptions, events, or counters, as defined by the service contract.

Monitor means to ensure that the right things are done and that these are in line with policies (IT Governance Institute, 2007). For SOA governance this means confirming whether or not service contracts are being properly applied. Monitoring is performed by an external point to monitor the operational state of the service. This is done by observing the change in the state of real world values. A monitor has a predefined set of rules, defined according to the service contract, which would observe when values cross certain thresholds and the QoS of the service deteriorates. The monitor would then raise an alert and provides feedback to the organisation so as to assist with governance. The monitor is passive and would not actively manipulate the service.

Although there are many current SOA governance technologies to govern services and their execution, there are no standards or methodologies to capture governance requirements from which to build a formal service governance model. The governance-by-contract approach is a first step to ensure that the service contract is not to be circumvented (Erl, 2008) by discouraging the improper application of policy rules, and the misuse of a service or an invalid change in state.

7 CONCLUSION

SOA governance is a very important and current topic that is being addressed by the IT community. It resides at the intersection between a new technology, namely SOA, and IT governance. To ensure the success of SOA, firm and consistent governance is needed.

Current approaches to SOA governance may lead to policies being applied ambiguously when service interaction occurs. To address this problem, service contracts are created for specific consumers or groups of consumers and these are agreed upon. The proposed governance-by-contract framework identifies how service contracts are designed with governance in mind, and includes mechanisms to control, enforce and monitor services. The governance-by-contract approach addresses aspects such as security requirements, service-level agreements based on QoS and key process indicators, and performance management, as set out in the service contract. The framework does not aim to replace current SOA governance technology, but rather seeks to use this technology to approach governance comprehensively.

This paper has introduced the concept of governance-by-contract. There is still much work to be done regarding the design of service contracts and their enablement. Future research aims to investigate, for example, information security governance frameworks in order to define a formal approach to defining the information security policies of a service contract and its enablement.

8 REFERENCES

Alencar, P., Cowan, D. & Kalali, B., (2003), A Service-Oriented Monitoring Registry. *ACM Digital Library*, 107-121, Oct., 2003.

Alves, A., Arkin, A., Askary, S., Barreto, C., Bloch, B., Curbera, F., Ford, M., Goland, Y., Guízar, A., Kartha, N., Liu, C.K., Khalaf, R., König, D., Marin, M., Mehta, V., Thatte, S., van der Rijn, D., Yendluri, P. & Yiu, A. (Editors). (2007). Web Services Business Process Execution Language Version 2.0. Available from: http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html. (Accessed 10 March 2008).

Amberpoint. (2008), Available from: http://www.amberpoint.com. (Accessed 15 October 2007).

Bajaj, S., Box, D., Chappell, D., Curbera, F., Daniels, G., Hallam-Baker, P., Hondo, M., Kaler, C., Langworthy, D., Nadalin, A., Nagaratnam, N., Prafullchandra, H., von Riegen, C., Roth, D., Schlimmer (Editor), J., Sharp, C., Shewchuk, J., Vedamuthu, A., Yalçinalp, U. & Orchard, D. (2006). Web Services Policy 1.2 - Framework (WS-Policy). Available from: http://www.w3.org/Submission/WS-Policy.

Brown, P. F., Hamilton, B. A. Laskey, K., MacKenzie, C. M., McCabe, F. & Metz, R. (Editors). (2006). OASIS: Reference Model For Service-Oriented Architecture 1.0. Available from: http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf. (Accessed 20 September 2007).

Burstein, M., Hobbs, J., Lassila, O., Martin, D., (editor), McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, P., Sirin, E., Srinivasan, N. & Sycara, K. (2004). OWL-S: Semantic Markup for Web Services. Available from: http://www.w3.org/Submission/OWL-S. (Accessed 26 March 2008).

Carter, S. (2007). *The new language of business: SOA and Web 2.0.* IBM Press.

Christensen, E., Curbera, F., Meredith, G. & Weerawarana, S. (2001). Web Services Description Language (WSDL) Version 1.1. Available from: http://www.w3.org/TR/wsdl. (Accessed 14 November 2007).

Clement, L., Hately, A., Rogers, T. & von Riegen, C. (Editors). (2004). OASIS: UDDI Version 3.0.2. Available from: http://uddi.org/pubs/uddi_v3.htm (Accessed 17 April 2008).

Curbera, F, (2007), Component Contracts in Service-Oriented Architectures, *Computer*, vol. 40, no. 11, pp. 74-80, Nov., 2007.

Dan, A., Franck, R., Keller, A., King, R.P. & Ludwig, H. (Editors). (2003). Web Service Level Agreement (WSLA) Language Specification. Available from: http://www.research.ibm.com/wsla/WSLASpecV1-20030128.pdf. (Accessed 20 February 2008).

Davidson, A., Fuchs, M., Hedin, M., Jain, M., Koistinen, J., Lloyd, C., Maloney, M. & Schwarzhof, K. (1999). Schema for Object-Oriented XML 2.0. Available from: http://www.w3.org/TR/NOTE-SOX/. (Accessed 4 October 2007).

Erl, T. (2006), *Service Oriented Architecture: Concepts, Technology, and Design.* New York: Prentice Hall

Erl, T. (2008), SOA Principles of Service Design: Indiana:Prentice Hall

Estes, K., Kesavarapu, K., Shipe, B. & Strom M. (2006). *Service-Oriented Architecture*. Unpublished manuscript.

Graham, S., Gottschalk, K., Kreger, H. & Snell, J. (2002), Introduction to Web services architecture, IBM Systems Journal, Volume 41, Number 2

Hallam-Baker, P., Kaler, C., Monzillo, R. & Nadalin, A. (Editors). (2006). Web Services Security: SOAP Message Security 1.1. Available from: http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf. (Accessed 10 January 2008).

Hawkins, J.M., (compiled by), (1995). *The South African Oxford School Dictionary.* 5th edition. Cape Town: Oxford University Press.

IBM. (2008). Available from: http://www-306.ibm.com/software/solutions/soa. (Acessed 28 November 2007).

IT Governance Institute. (2007). *Cobit 4.1.* Illoinois: IT Governance Institute.

Jencmen, A. & Yehudai, A. (2006), Fortified Web Services Contracts for Trusted Components, *Computer,* pp. 919-926, Sep., 2006

Marks, E.A. & Bell,, M. (2006), *Service-oriented Architecture A Planning and Implementation Guide for Business and Technology.* New Jersey:Wiley

OASIS SOA Reference Model Technical Committee (2006). Policies and Contracts. Available from: http://wiki.oasis-open.org/soa-rm/TheArchitecture/PoliciesAndContracts (Accessed 25 January 2008).

Systinet. (2008). Available from: (https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-130-27%5E1461_4000_100__). (Accessed 20 October 2007).

Von Solms, S.H. & Von Solms, (2006). *Information Security Governance*. Unpublished draft.