ISSN:2249-5789

Towards Enhanced Service Oriented Architecture to Improve E-Services Applications

Usama M. Maabed, Ahmed M. El-Fatatry, and Adel A. El-Zoghabi
Department of Information Technology, Institute of Graduate Studies and Research, Alexandria University.
Alexandria, Egypt
umaabed@alexu.edu.eg, elfatatry@alexu.edu.eg, zoghabi@alexu.edu.eg

Abstract

Service oriented architecture has greatly influenced web-based development. However, the dream of system integration and automatic service discovery has only been partially fulfilled. The shutdown of the universal registry in 2006 raised several concerns on the visionary promise of the service oriented computing. Most notably, the ability to assemble application components from loosely coupled network of services. In addition, creating dynamic business processes that span the entire organizations' computing platforms is another concern. The aim of this paper is to analyze the current limitations and challenges of SOA and to introduce an approach to address such limitations. The proposed approach makes use of the functionality of the service registry layer within the SOA. This is done through dealing with the registry layer as a service provider rather than as yellow pages, which is expected to overcome weaknesses in current architectures and meet the raising challenges.

1. Introduction

Service Oriented Computing (SOC) paradigm uses services to support the development of rapid, low-cost, interoperable, evolvable, and massively distributed applications [1]. SOA comes up as the key to bring this vision to reality. The vision that considers a global Web service directory as the backbone and the key enabler of e-commerce has been challenged by the shutdown of the universal business registry in 2006 [2]. The consequences to this failure have led to limiting SOA implementations to be more intranets based. We argue that extending the registry layer's functionality will add needed momentum to both SOA deployments and the global electronic marketplace vision.

Our research aims to analyze the mandatory requirements and extend the web registry layer's functionality within SOA to overcome the current limitations. In this paper, we propose an enhanced SOA that makes use of the major non-functional services. The focus is to improve the service discovery and selection process, which in turn leads to building consumer trust and satisfaction.

The paper is organized as follows: section 2 covers the basic concepts of SOA, section 3 focuses on SOA and the related research issues, section 4 investigates the current SOA problems, section 5 presents the mandatory requirements to improve current SOA, and section 6 discusses the proposed architecture. Conclusions and future work are summarized in section 7.

2. Background

This section discusses the importance of SOA as an abstract architecture; the aim is to introduce a definition that does not rely on any particular technology or implementation. Software architecture is the bridge between business goals and software systems [3].

Architectural design is important for at least three reasons.

- First, an architectural description makes a complex system intellectually tractable by characterizing it at a high level of abstraction [4].
- Second, architectural design lets designers exploit recurring organizational patterns. Such patterns, or architectural styles, ease the design process by providing routine solutions for certain classes of problems [4].
- Third, it makes the decision maker aware of the overall deployment roadmap.

SOA started as an idea that promises to achieve business interoperability through reachable web services. Services can be used by consumers repeatedly and considered as a model consisting of three major parts: service provider, service consumer, and directory (registry) of services.

SOA main objectives are:

- Reducing redundancy by avoiding the need to rebuild functionality for new applications, and delivering applications faster with lower cost. This results in an increased reuse of services, which leads to a reduction in the development costs.
- Decoupling business logic from the source of provision, which enables more flexibility as this logic can be easily changed, updated, or even replaced to address new business needs.

- Composing services in new ways without having to change or rewrite them.

SOA has many and different definitions some are generic such as: "A set of components which can be invoked, and whose interface descriptions can be published and discovered." [5], others came as industry based from the providers of a specific technology such as “SOA is an architectural approach to creating systems built from autonomous services” [6]. Some came from business analysts, research’s labs and groups such as "SOA is a style of multi-tier computing that helps organizations share logic and data among multiple applications and usage modes." [7]

Such definitions miss the consumer needs and satisfaction consequently; the following problems resulted as follows.

- every major standards body with multiple groups attempted to define SOA, added to the confusion and slowing adoption of SOA, and
- A lack of common vocabulary across the industry has made it difficult to share best practices [8].

Based on our survey, it is found that a more accepted definition is the ITIL service definition, which is “A service is a means of delivering value to customers by facilitating outcomes customers want to achieve without the ownership of specific costs and risks” [9].

Finding suitable web services that help to achieve a certain goal is widely considered as a key task in such shared global marketplace of services. Finding a suitable Web service depends on a technical interface described using Web Service Description Language (WSDL).

Web services are built such as software components or classes, with the specification of both functional requirements (operations and messages) and non-functional requirements such as quality of services (e.g. timeless, security, availability, reliability, scalability, and manageability). Therefore, a Web service description should adhere to the well-established requirements of the consumers and cover not only functional, but also non-functional properties (NFP) of a service [10], [11].

Considering the previously stated needs, we introduce a more value-based definition for SOA within our research, which is:

“SOA is an architecture of functional and non-functional services that deliver value to a consumer in secure and trusted working environment and gain consumer satisfaction”

This definition does not rely on any particular technology or implementation of SOA. It focuses on the service usability needs by emphasizing not only on how service can be used, but also on how to encourage this use. Meeting today’s problem and needs requires empowering SOA with means to handle non-functional characteristics such as:

- Stability and availability.
- Trusted by its consumer.
- Committed to what it should do.
- Accepted output quality.
- Clear legality control.

3. Related Works

3.1. Work on Improving Current SOA

The failure of public registries in 2006 brought about more questions on SOA potential and its ability to achieve the expected business interconnects. Fears of depending on poor managed public registries results in increasing the locally based inter-organization SOA implementations.

This left only the less important business services where the risk and failure are less critical to be offered by web services intermediaries in more open working environment. The question remains how SOA can break such rigid boundaries to achieve business interoperability and become more reachable services. Recent researches approaches have investigated this major challenge.

Basic SOA does not address the principal concerns such as organization of service supply and demand, service transaction management, security, trust, and other concerns that are necessary for a robust and reliable SOA [12].

To address these concerns, extended SOA (xSOA) concept has been developed by Papazoglou and Van den Heuvel [13], which consists of three layers.

- The web services foundation layer; in which basic services are published, found, and bound.
- The web services composition layer that includes the necessary roles and functionality for the aggregation of multiple services into, and
- The management and monitoring layer, which ensures, that services in the lower layers are managed where each layer depends on the functionalities of the lower level.

The proposed extended SOA considers the main business requirements from a higher level of abstraction. However, a clear roadmap to resolve current problems facing SOA implementations is missing.

Rainer developed a framework for the analysis of service directories in business networks and provided design options for combining separate and distributed service directories [13], Legner and Heutschi identify three objectives for the adoption of SOA: 'to realize a standardized integration infrastructure', 'to decouple application domains', and 'to create flexible user/business process integration' [14].

Although empirical data related to the web services market and the transaction volumes are not publicly available, Legner used a 5 years data collected by researches that are restricted to open Web service directories such as e-Sigma.com, RemoteMethods, StrikeIron, and X-methods that survived the shakeout phase in 2006. Analyzing the current market structure of B2B web services and the current classification of Informediary, e-Hub, and e-Market with focus in web register, the following trends will materialize.

- With the increasing maturity of web services technologies, the offering of coarse-grained web services, which provide reusable business-level functionality, will increase.
- Due to the falling transaction costs, specialist service providers will be able to realize scale efficiencies in repetitive tasks and establish a commercial offering of B2B web services.

Although web services specialists emerge in specific domains, such as compliance, online validation and alerting but commercial web services, which enable companies to out-task discrete, repetitive tasks to specialized service providers, continue to be relatively scarce.

Scholten, Fischer, and Zirpins [15] provided an extended comparative analysis, based on mapping Web service quality factors to types of intermediaries and found there are improvement gaps not considered and taken seriously by the intermediaries, also more focus and a better monitoring of quality could reduce the customer's pre-selection time and support his choice of quality.

Although, this analysis clarifies the importance of the quality factors in promoting web services usage. However, it focuses on identifying differences among intermediaries and did not consider the current challenges or provide new ideas for enhancing the current SOA offering.

3.2.Web Service Discovery Initiatives

The main challenge of service discovery is the use of automated tools to discover accurately services with minimal user involvement [1]. This automation requires formulation of the requesters' needs in a way that corresponds to the service description, either by name or by function.

When reviewing the Web Service description initiatives, at least two factors should be taken into consideration, namely: the scope of the description and the formalism used to express it.

A Semantic Web Services representation is shaped by three main initiatives: OWL-S (Ontology Web Language for Web Services), WSMO (Web Services Modeling Ontology) and SAWSDL (Semantic Annotations for WSDL) [16]. The first two share a quite similar approach to a SWS description, automate web services interactions by defining the description of a service using standardized language, and in some sense are interoperable due to the activities of the research communities [17].

An important issue in the Web service discovery process is the services providers and requestor negotiation process. This process aimed at finding an integrate solution that is optimal for both sides based on the agreed SLA. Considering the criteria in SLA without having the effective monitoring and measurement capability will result in disputes and the loss of faith in the provided service, and its provider as well.

Web services providers use resources to provide services to requestor in return for benefits. On the other hand, web services requestors pay for the services from providers in return for benefits as well. With an increasing number of web services providing similar functionalities, Quality of Service (QoS) is becoming an important criterion for the selection of the best available service. QoS parameters include availability, response time, customer satisfaction, trust, and others.

The main approaches for Web service discovery mechanism include the following [18] [19] .

- UDDI (Universal Description, Discovery, and Integration): is a standard for centralized repositories used for web services registration, discovery and integration. The first UDDI Business Registry (UBR) nodes were run by IBM, Microsoft, SAP and NTT Com.
- Specialized portals: provide the web services using focused crawlers as well as manual registration and provide searching through HTML interface such as Methods, BindingPoint, StrikeIron, WebServiceX.NET, RemoteMethods, Woogle, and eSynaps.
- Web Search Engines: are able to restrict the search to WSDL files such as Google and Baidu.

These approaches can be compared by focusing on the following aspects: number of services, quality of

information associated with them and time and efforts to give the desired results. The last two approaches are based almost on the same concept of Web Services Inspection Language (WSIL); but decentralize the centralized model of service publication within UDDI registry and distribute it as pieces.

As a result, this takes the problem from UDDI and distributes it on the other web servers, which allows service offering without any control on the service quality, or maintaining quality control part, which might be non-compliant with the global vision of SOA.

While the structure of service directories have emerged into a standard for local SOA implementations, the question remains how service directories should be organized in a business network, i.e. when multiple companies with individual SOA solutions interact.

One major difference between our research work and the previously described initiatives is in dealing with the current problems from architectural-point of view. This approach is expected to reach a more accepted solution and tend to focus more on the practical implementation side. In our research, we analyze the mandatory requirements for successful SOA taking into consideration the current needs and challenges.

4. Problem Definition

Although a number of approaches addressed the topic of automatic discovery of services in multi-agent systems or based on description logic reasoning settings, they failed to provide a viable solution for the discovery problem due to the lack of a clear conceptual model and the incomplete understanding of the necessary requirements for SOA.

As the level of complexity continues to evolve, traditional architectures are not capable of dealing with the current problems [20]. Although there have been numerous efforts that tend to focus on applying new technologies to develop web services, there is less focus on SOA architecture improvement so that it can meet its visionary objectives.

One essential components of SOA is the UDDI or registry, which used to be defined as a repository or yellow pages for software interfaces that are published by the service provider. It suffers from as the following limitations [21]:

- UDDI has not been intended to serve as a search engine for Web service discovery.
- UDDI registration is voluntary and thus can easily become passive.
- UDDI does not provide any guarantee of the validity or the quality of information it contains.
- A disconnection exists between UDDI and the current Web. UDDI is incapable of providing quality-of service (QoS) measurements for registered web services.
- UDDI does not maintain or provide any Web service life-cycle management.

We argue that the failure of the UDDI in 2006 was a result of dealing with it as yellow pages rather than as a service provider. This result in a huge number of fake and invalid services, and losing trust in utilizing such public registries. In addition, more Intranet based SOA deployments rather than Extranet and Internet based deployment. Such implementation approach apparently destroyed the interoperability.

Such concerns require the re-thinking of the overall architecture and the introduction of new ideas that can enhance the current SOA implementations and provide the expected features that encourage and promote web services usage.

Although there have been extensive efforts to enhance the discovery of web services, many of them failed to address the issues of handling discovery operations across multiple registries [22]; and handling the service's trust and security issues. QoS has not been clearly standardized.

UDDI specification version3 does not include QoS as part of its publication or inquiry APIs. In addition, it does not provide the ability to associate QoS information within the registry [23].

These concerns have been considered in a complementary draft model known as Web Services Quality Model (WSQM), which is under development and published by OASIS [24]. WSQM is a conceptual model in which eight quality related roles have been defined which are: acquirer, service developer, service provider, service consumer, quality information provider, quality assurer, quality authenticator, and quality manager. It also organizes the quality factor in three major categories, which are system-information quality group, service-measurement quality group and business-value quality group. WSQM covers a number of quality related activities, but it assumes an ideal and ready working environment.

In reality, many service providers is still straggling in the right SOA implementation, and many recent implementation still consider closed and domestic implementation rather than open and Internet based.

Dealing with web services identifies the following major problems:

4.1.Service registry does not support trust-based discovery.

In an open environment, where anyone can publish services, a consumer may select a service with poor quality, expensive, or even a harmful service. Therefore, mechanisms are needed to help consumers to distinguish the well-defined services from bad services [25]. Trust and reputation play an important role in Web service selection. Reputation and trust management is a fundamental problem in SOA as service providers and consumers may have no history of dealing with each other. At the same time, service consumers need to know not only what a service can do, but also how well a service can do that. Before using web services, many concerns must be raised before making a decision on either to use it or not. Such concerns can be summarized as follows.

- Will the service do what is mentioned in its descriptive part? if yes; with which degree of quality the service will be achieved?
- Can we trust this service provider?
- What is the opinion of other consumers with respect to dealing with a particular provider?

Quality of Service (QoS) deals with various parameters that can be monitored during service invocation and use. This approach provides a means to evaluate a service quality levels. Considering that, consumers might have no access to these monitoring mechanisms or to its infrastructure; consumers need a way to trust these measurements and evaluate them against their own experience on service usage. The latter is referred to as Quality of Experience (QoE) [26].

Reputation is a general concept widely used in all aspects of knowledge ranging from humanities, arts, and social sciences to digital sciences [27]. Reputation systems have been used to manage reputation of service providers, according to the QoS provided as well as service consumers, according to their usage of services. Different researches with different approaches dealt with reputation. One of the main advantages of having centralized reputation service is the feasibility for an individual to know consistent reputation about other service providers based on previous experiences.

Service selection problem has been tackled by various researches. Blum [28] has discussed using UDDI to store QoS related attributes. However, large numbers of response to QoS feedback and the collection of real time mass QoS data easily lead to the overload of the registries server. At the same time, some web services, which have good QoS performance and favored by the consumer are excessively invoked and lead to distortion of QoS [29].

It is expected that this centric approach will suffer from a single point of failure and the stored information may become outdated in dynamic networking environment. Some researches focused on client-side quality evaluation [30] which results in multidimensional QoS models presented from a consumer's perspective. Taking into consideration that QoS are defined in various ways and measured by different metrics, it is apparent that the above approaches cannot effectively resolve the trustworthiness of QoS and provide accurate result to the consumer.

4.2.Lack of proper support for searching and selecting web services based on quality attributes.

Rapid growth of published web services makes their discovery more difficult due to the existence of many web services, which have similar functional characteristics [31]. In such case, the question is how to search and choose between a number of services that provide the same functionality but differ in their non-functional characteristics. Addressing such concern highlights two major issues:

- The performance of searching in directories.
- Dealing with consumers' feedback.

4.2.1 The low performance of searching in directories. UDDI is organized according to categories of business activities. Service providers publish web services by adding webs service's WSDL specifications to the appropriate UDDI directory category. Through a well-defined API, service requester can browse the UDDI directory by category to discover existing services potentially relevant to what they need to query. However, this category-based service discovery is clearly insufficient, because it relies on the shared common-sense understanding of an application domain and among the developers who publish and request the specified services *[32]*. Service directories only offer similarity-matching operations, which is a limited and a time consuming process. Fardin et al *[33]* performed an experiment to measure the impact of registry based processing time by defining the Web service response time (Rt) as:

$$Rt = t_{Request} + t_{Reply} + t_{Processing}. \quad (1)$$

The experiment showed that the average response time of the service directory plus the required matching, without considering the network delay, is 66,453ms. The response time that was calculated for finding a single service. Thus, he concluded that:

- The source of the delay is the service directory alone.
- In complex situations, such as service composition, a substantial amount of matching would be performed.
- In case of failure, the discovery operation must be repeated, that is time consuming.
- Furthermore, service directories spend a substantial amount of time on matching, but they never record the results for future usage. Therefore, the similarity matching operations utilized by current service registries is time consuming and limited.

4.2.2 No consideration for feedback from previous transactions. There is no consistent way for selecting web services based on non-functional parameters. Consumers perceive quality through the prism of their own experience and it is important to see how their evaluation of the quality provided is mapped to the specific quality parameters offered by the provider *[26]*. Allowing consumers to evaluate and rate the QoS of the provider based on their experience can be used for ranking the provider's reputation, and for creating consumer profile, which can be used for better recommendations during service search and selection.

4.3.There is no mechanism for dispute resolution.

Conflict is a natural part of human communication. Internet based transactions, such as web services, come with new forms of disputes that require specialized handling from legal point of view and its supported authorities as well. Dealing with web services brings concerns such as:

- The failure of a service provider to fulfill the agreed upon demand in the service level agreement.
- Handling legal consequences of a dispute.

Alternative dispute resolution (ADR) is generally defined as processes that are 'alternative' to traditional court proceedings (litigation). Online dispute resolution (ODR) extends this trend even further. While ADR represents a move from a fixed and formal process to a more flexible one, ODR (by designating cyberspace as a location for dispute resolution) moves ADR from a physical to a virtual place [34]. The main trends in ODR, either using multi-agent or web services, focus on how to develop system used for Internet-related disputes. Considerable research works has been invested in the definition of online resolution services for legal and commercial disputes. For example, Ebay has a defined process for resolving transactional disputes over the sale of items [35]. Bonnet [36] has proposed Web service model for online dispute resolution. This solution came as a very specific kind of e-services. The focus was security needs and non-repudiation property that require the support of new types of communication adapted to dispute resolution processes, which includes:

- Negotiation, mediation, and arbitration.
- Security mechanisms such as integrity and confidentiality of sensible data.
- The main limitation of such approaches is that they do not integrate in the SOA life cycle.

4.4.No validation for service descriptions.

Lack of validation in UDDI registries lead to having two categories of problems:

- Invalid and unusable web services:

Independent research carried out by SalCentral shows that over 67% of entries in UDDI registries are either invalidly formatted or validly formatted but unavailable *[37]*. Fan *[38]* crawled five service registries. From 2432, registered services at the time of crawling only 1544 entries had a valid WSDL URL, which is about 63%. After removing services with invalid WSDL entries, only 640 valid entries were identified, which is 26 % of all registered services. Such huge numbers of invalid entities in UDDI has been one of the main reasons for the shutdown of the Universal Business Registry in 2006.

- Poorly written WSDL documents:

UDDI provides keyword-based matching for Web service discovery. Finding a suitable Web service depends on a technical interface described using WSDL. A discoverer looks for third-party web services by sending keyword-based queries to a registry, which returns a list of candidate services in response. Then, the discoverer analyzes the retrieved services and selects the most suitable one that matches his/her needs.

Poorly written WSDL document, besides reducing its chances of being properly retrieved by a registry, hinders human discoverers' ability to understand and select the service afterward [39]. Pasley [40] explored the negative side effects of using XSD wild-cards such xsd:any and xsd:any-Attribute in web services interfaces, and such approach requires significant cost that includes increased complexity, handwritten validation, and vague interface contracts. Instead of using XSD wild cards, a versioning strategy for WSDL documents has been introduced. Rodriguez et al [39] addresses the quality of WSDL documents from the perspective of a discoverer, pursuing recurrent problems that attempt to prevent the understandability and discoverability of a service. He presents a catalog of

eight bad practices that frequently occur in a body of public WSDL documents, also supplies each problem with a practical solution. Crasso et al [41] identified the most common bad practices in WSDL documents, and recommend the following steps for improvements.

- Separating the schema from the definition of the offered operations,
- Removing repeated WSDL and XSD code.
- Putting error information within fault messages and only conveying operation results within output messages.
- Replacing WSDL element names with explanatory names if original names are cryptic.
- Moving non-cohesive operations from their original port type to separate port types.

The above research focused on problems in service discovery, without providing a mechanism or business model to evaluate the service readiness for public use.

5. Architecture Requirements

Having analyzed the major problems of SOA, in this section we analyze the requirements for the solution

5.1.Monitor service availability and validity.

One of the most challenging problems in web services is to predict the QoS performance and provide a precise QoS value for the Web service consumers [29]. SOA needs to be aware of the expected quality that a provider is expected to deliver. Various QoS attributes can be monitored and shared such as availability, accessibility, response time, throughput, and security.

5.2.Evaluate and rank both services and service providers.

It is required to have the functional description of the Web service and QoS information to evaluate and rank the service provider reputation based on the performance on certain QoS parameters. This will provide the means to consider the QoS during searching for web services. In addition, Registering the agreements between service provides and consumers with SLA is required as part of monitoring and validating the consumer's feedback against the provided services.

5.3.Providing recommendations for choosing service

Although there are already available, recommendation systems but still not fully integrated with SOA. Considering Consumer profiling and recommendation technologies can provide two main enhancements.

- Enhancing the search performance by narrowing the search scope.
- Gain more customer satisfaction by providing search results meet the relevant behavioral needs.

5.4.Web services disputes handling.

The following issues need to be considered to facilitate the dispute handling integration with current SOA.

- A crucial issue in any dispute handling is identifying the complainer and defendant. Current registries do not require a clear identification of the interacting parties. It just acts as a bridge to join a provider with a consumer. There is a need for a standardized mechanism for registering the identities of the interacting parties to refer to in case of having problems later on.
- The acceptance and the general conditions of SLA must be available to the dispute resolver, which will clarify what agreed on comparing to what the complainer is asking for.
- Another requirement involves the evidence that Web service functionality was available and has been utilized successfully by consumer during the time of dispute. This will provide the dispute handler a way to understand what has happened during the agreement process.
- The dispute resolver must be ascertained that the content of the repository has remained unchanged when gathering evidence from electronic repositories,
- Information concerning disputes must be protected until disputes are resolved.
- The results of dispute resolution should be connected to the provider ranking system.

6. Proposed Architecture

This work is motivated by the need for empowering and decomposing the service registry layer into self-contained layers that deal with the required functionality. This approach overcomes the current architectural limitations, and proposes a solution that achieves the desirable properties and perspectives of web services to realize an efficient SOA, and ultimately an agile e-business.

The proposed solution focuses on the service registry layer for a number of reasons listed below.

- It is the primary interface for service consumers.
- Without such central repository, other solutions result in point to point solution with great burden to manage and sustain.

As the QoS is multidimensional, it may include time, price, reliability, availability and reputation. The monitored QoS includes not only the customers' related evaluation (reputation), but also the QoS data detected by monitors (time, reliability, availability).

Compared to WSQM, our approach considers the following:

- Quality information provider is part of service discovery.
- The Quality authenticator and Quality assurer to be the control role and the auditing authority in providing quality information by UDDI.

Our proposed architecture, shown in figure 1, ensures that service providers fulfill the following.

- Register only qualified services,
- Provide valid service with accepted quality,
- Provide quality feedback on the provided services.
- Maintain requester's feedback and satisfaction.

This is achieved by adding specific components that provide the required functionalities.

SMod. (Service Moderator) is the main interface to the service providers and requesters that handles the following.

- Service validation,
- Service registration,
- Service provider profiling
- Service requester profiling,
- Registration with its correspond SLA,
- Recommend a service and service provider based on specific requester needs and the service provider history, and
- All related communication with other service registries including referring to other SMod and enforcing any domestic or cross boundaries regulations and rules.

SMon. (Service Monitoring) guarantees a clean registry with valid services by updating the service registry with the status of the Web service such as: service availability, response time, and the quality of its output.

SEva. (Service Evaluation) evaluates the following aspects.

- Service provider history by reviewing complains and evaluation coming from the requester,
- Service requester history and considers the evaluation of previous transactions.

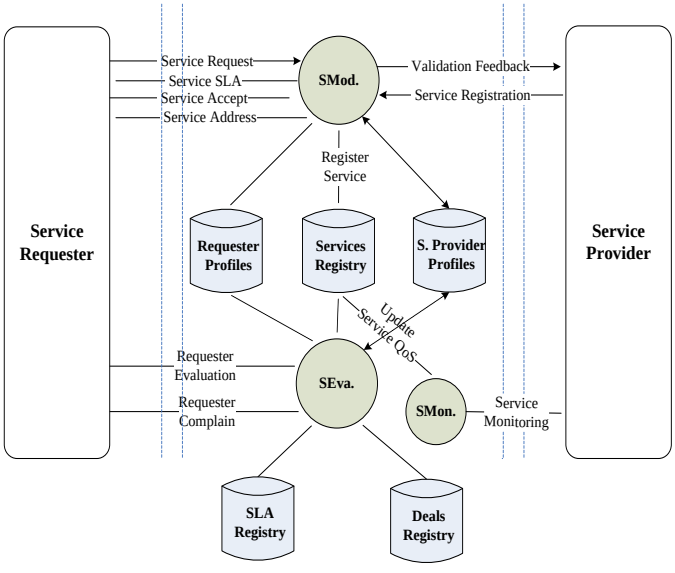

Figure 1: Enhanced Service Oriented Architecture

7. Conclusion and Future Work

The shutdown of the universal business registry in 2006 put questions on the visionary promise of SOA. The dream of system integration and automatic service discovery has been partially fulfilled at best.

In this paper, we provided an analysis of both the problems that lead to the current situation and the requirements for the solution.

Extending the functions of Web service registry can improve the overall SOA architecture. This approach provides better flexibility in the usage of web services. Our proposed architecture provides the necessary mechanics for building trust in SOAs. Our future work will include more improvements in the architecture specifications and will implement the proposed architecture on a number of prototypes.

References

[1] M. P. Papazoglou, P. Traverso, S. Dustdar and F. Leymann, "Service-Oriented Computing: State of the Art and Research Challenges," *IEEE Computer*, vol. 40, no. 11, pp. 38-45, 2007.

[2] C. Legner, "Do web services foster specialization? - An analysis of commercial web services directories," in *Wirtschaftinformatik Proceedings 2009*, 2009.

[3] L. O'Brien, L. Bass and P. Merson, "Quality Attributes and Service-Oriented Architectures," Pittsburgh, PA., 2005.

[4] D. Garlan, R. Allen and J. Ockerbloom, "Architectural Mismatch or Why it's hard to build systems out of existing parts," in *ICSE 95 Proceedings of the 17th international conference on Software engineering* , Seattle, Washington, 1995.

[5] H. Haas and A. Brown, "Web Services Glossary," W3C, [Online]. Available: http://www.w3.org/TR/ws-gloss/. [Accessed 19/6/2010].

[6] J. Evdemon, "Principles of Service Design: Service Patterns and Anti-Patterns," Microsoft Corporation, [Online]. Available: http://msdn.microsoft.com/en-us/library/ms954638.aspx. [Accessed 10 8 2010].

[7] R. W. Schulte and Y. V. Natis, "Service Oriented Architectures, Part 1," Gartner, 1996.

[8] S. Durvasula, M. Guttmann, A. Kumar, J. Lamb, T. Mitchell, B. Oral, Y. Pai, T. Sedlack, H. Sharma and S. Sundaresan, "SOA Practitioners' Guide Part 1, Why Services-Oriented Architecture?," The SOA Alliance, 2006.

[9] S. Taylor, V. Lloyd and C. Rudd, "ITIL version3 Service Design," OGC Publication; TSO (The Stationery Office), UK, 2007.

[10] W. Abramowicz, M. Kaczmarek and D. Zyskowski, "Duality in Web Services Reliability," in *Proceedings of AICT-ICIW '06*, Washington, DC, USA, 2006.

[11] W. Abramowicz, A. Godlewska, J. Gwizdala, T. Jakubowski, M. Kaczmarek, M. Kowalkiewicz and D. Zyskowski, "A Survey of QoS Computation for Web Service Profiling," in *Proceedings of ISCA 18th International Conference on Computer Applications in Industry and Engineering*, Hawai, USA, 2005.

[12] M. P. Papazoglou, Web Services: Principles and Technology, Prentice Hall, 2008, pp. 1-752.

[13] R. Alt and M. Smits, "Design Options for Service Directories in Business Networks," in *Proceedings of the 18th European Conference on Information Systems (ECIS '10)*, Pretoria, South Africa, 2010.

[14] C. Legner and R. Heutschi, "SOA Adoption in Practice- Findings from Early SOA Implementations," in *Proceedings of the Fifteenth European Conference on Information Systems*, 2007.

[15] U. Scholten, R. Fischer and C. Zirpins, "Perspectives for Web Service Intermediaries: How Influence on Quality Makes the Difference," in *EC-Web 2009 Proceedings of the 10th International Conference on E-Commerce and Web Technologies*, Linz, Austria, 2009.

[16] W. Abramowicz, R. Hofman, W. Suryn and D. Zyskowski, "SQuaRE based Web Services Quality Model," in *IMECS 2008, Proceedings of the International MultiConference of Engineers and Computer Scientists*, Hong Kong, 2008.

[17] A. Ankolekar, D. Martin, D. McGuinness, S. McIlraith, M. Paolucci and B. Parsia, "OWL-S' Relationship to Selected Other Technologies," W3C, 2004.

[18] D. Bachlechner, H. Lausen, K. Siorpaes, D. Fensel and I. Toma, "Web Service Discovery–A Reality Check," in *Proceedings of 3rd European Semantic Web Conference (ESWC 2006)*, Budva, Montenegro, 2006.

[19] S. Bashir, F. Khan, Y. M. Javed, A. Khan and M. Khiyal, "INDEXER BASED DYNAMIC WEB," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 7, no. 2, pp. 153-159, 2010.

[20] J. Bih, "Service oriented architecture (SOA) a new paradigm to implement dynamic e-business solutions," *Ubiquity*, vol. 7, no. 30, p. Article No. 4 , 2006.

[21] A. Yousief, H. Arafat and A. Saleh, "Merging UDDI and Search Engines the Future Image for Web Services' Discovery," *The Online Journal on Electronics and Electrical Engineering (OJEEE)*, vol. 3, no. 4, pp. 478-483, 2011.

[22] E. Al-Masri and Q. H. Mahmoud, "Crawling Multiple UDDI Business Registries," in *WWW '07 Proceedings of the 16th International Conference on World Wide Web*, Banff, Alberta, Canada, 2007.

[23] E. Al-Masri and Q. H. Mahmoud, "Discovering the Best Web Service," in *Proceedings of the 16th international conference on World Wide Web WWW 07*, New York, NY, USA, 2007.

[24] E. Kim and Y. Lee, "Web Service Quality Model v1.0. Committee Draft," The Organization for the Advancement of Structured Information Standards, OASIS , 2008.

[25] Y. Wang and J. Vassileva, "Toward Trust and Reputation Based Web Service Selection:A Survey," *International Transactions on Systems Science and Applications (ITSSA)*, vol. 3, no. 2, p. 118–132, 2007.

[26] K. Tserpes, F. Aisopos, D. Kyriazis and T. Varvarigou, "A recommender mechanism for service selection in service-oriented," *Future Generation Computer Systems*, 2011.

[27] A. Arenas, B. Aziz and C. Silaghi, "Reputation management in collaborative computing systems," *Security and Communication Networks,* vol. 3, no. 6, p. 546–564, 2010.

[28] A. Blum, "UDDI as an extended web services registry: versioning, quality of service, and more," *SOA World magazine,* vol. 4, no. 6, p. 41, 2004.

[29] Q. Tao, H. Chang, C. Gu and Y. Yi, "A novel prediction approach for trustworthy QoS of web services," *Expert Systems with Applications,* vol. 39, no. 3, p. 3676–3681, 2012.

[30] M. F. M. Fudzee and J. H. Abawajy, "QoS-based adaptation service selection broker," *Future Generation Computer Systems,* vol. 27, no. 3, pp. 256-264, 2011.

[31] T. Rajendran and P. Balasubramanie, "Analysis on the Study of QoS-Aware Web Services Discovery," *Journal Of Computing,* vol. 1, no. 1, pp. 119-130, 2009.

[32] L. Bai and M. Liu, "Fuzzy sets and similarity relations for semantic web service matching," *Computers & Mathematics with Applications,* vol. 61, no. 8, p. 2281–2286, 2011.

[33] A. M. Fardin, N. B. Naser and N. M. Ali, "Empower service directories with knowledge," *Knowledge-Based Systems,* vol. 30, pp. 172-184, 2012.

[34] B. Abrahams and J. Zeleznikow, "A Multi-Agent Architecture for Online Dispute Resolution Services," in *5th International Workshop on Online Dispute Resolution, in conjunction with the 21st International Conference on Legal Knowledge and Information Systems (JURIX 2008)*, Firenze, Italy, 2008.

[35] M. Billings and L. A. Watts, "Understanding Dispute Resolution Online: Using Text to Reflect Personal and Substantive Issues in Conflict," in *CHI '10 Proceedings of the 28th international conference on Human factors in computing systems*, 2010.

[36] V. Bonnet, K. Boudaoud, M. Gagnebin, J. Harms and T. Schultz, "Online Dispute Resolution Systems as Web Services," in *Hewlett-Packard Open View University Association Workshop*, 2002.

[37] T. Modi, "WSIL: Do we need another Web Services Specification?," May 2004. [Online]. Available: http://www.webservicesarchitect.com/content/articles/modi01.asp. [Accessed May 2010].

[38] J. Fan and S. Kambhampati, "A snapshot of public Web Services," *ACM SIGMOD Record,* vol. 34, no. 1, pp. 24-32, 2005.

[39] J. M. Rodriguez, M. Crasso, A. Zunino and M. Campo, "Improving Web Service descriptions for effective service discovery," *Science of Computer Programming,* vol. 75, no. 11, pp. 1001-1021, 2010.

[40] J. Pasley, "Avoid XML schema wildcards for Web Service interfaces," *IEEE Internet Computing,* vol. 10, no. 3, pp. 72-79, 2006.

[41] M. Crasso, J. M. Rodriguez, A. Zunino and M. Campo, "Revising WSDL Documents: Why and How," *IEEE Internet Computing,* vol. 14, no. 5, pp. 48-56, 2010.