SOA-oriented approach and full distributed approach comparison

Elena Kaina, Yury Korolev

Saint-Petersburg Electrotechnical University "LETI" (ETU),
Saint-Petersburg, Russia.
{lenakaina, yury.king}@gmail.com

Abstract

In this paper we discuss two approaches for distributed software development. Former is classic Service Oriented (SOA), and last is kind of SOA based full distributed.

I. INTRODUCTION

SOA are being constructed as multi-layered service model which is in concordance with weak service connect, service reuse, ability to service discovery and composition.

SOA is a paradigm which is being used for designing, development, and management of discrete logical units (services) in computational environment. Application of this approach requires from developers to design applications as set of services even advantages of this solution are not obvious. Developers have to “go beyond” of their applications and figure out how to use existing services or investigate how their services will be used third party.

The big disadvantage in development with SOA is services might be allocated and linked only during design process. This is a limitation for process automation. It's required to weak connectivity and dependency between services, which should be done by people who is not from IT domain.

The advantages of software developed with SOA approach are:

- loose coupling what means that separate information system elements should be maximally autonomous, or independent from the other elements;
- re-using what allows to make processes more correlated due to the using of the same components;
- expandability.

SOA turned out to be a rather convenient and flexible approach for dynamically developing systems. However SOA embedding into different fields made it necessary:

- to simplify humans' interaction with services not involved in IT;
- to use the policy of single information input along with multiple usage of the very information about different services' objects.

Above mentioned troubles are the reasons for starting development of full distributed software design approach. This approach is based on top of SOA concept but with several extensions, such as Semantic Web, Web 2.0, Context Management and so on. Let's overview each of them.

II. SEMANTIC WEB

Semantic Web approach is based on development of special languages for presenting semantic information in form which can be used for automatic processing. Semantic principle which is provisions formal data description and dynamic models, which allow data processing automatization by logical reasoning. Combining this principle with service oriented design we can reach useful software characteristics such as scalability, semantic interoperability, formal models for services and ontologies, which allow full or partial problem resolving automatization, for instance service discovery, service concordance, composition and so on.

Web Service Modeling Ontology (WSMO) defines additional layer on top of existing web service models by adding semantic markup for functional and non functional dynamic service characteristics. WSMO is developing as a structure which includes concept model description, which defines web service characteristics, ontology, goals, web-services and proxies.

Figures 1 and 2 depict the comparison of pure SOA and Semantic SOA.

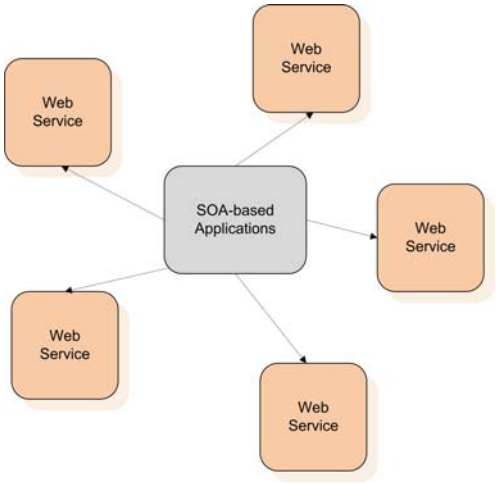

Figure 1. Pure SOA

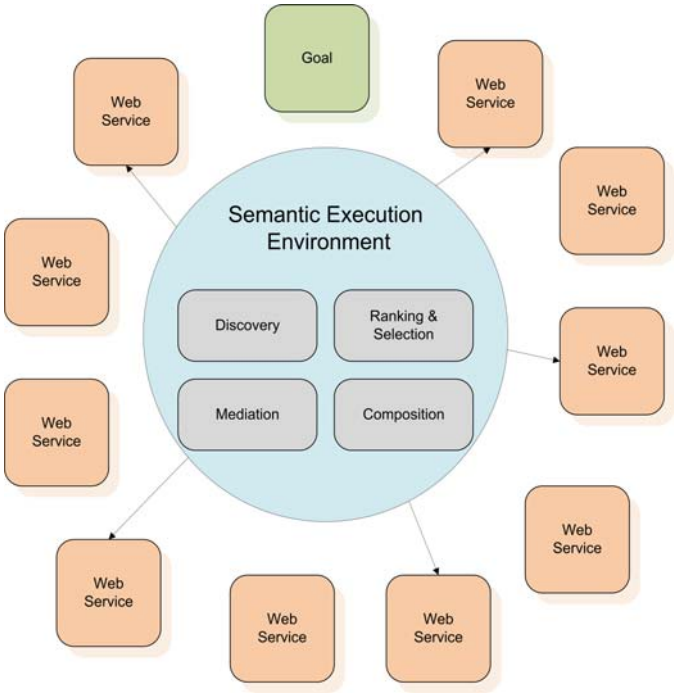

Figure 2. Semantic SOA

The semantic application environment of Semantic SOA supplies futher function of compatibility providing:

- Discovery – detection of different services for the task given.
- Selection – choice of the most suitable services among all available ones.
- Mediation – contradiction resolution (in data, protocols, processes) between combined web-services.
- Replacement – displacement of services with their equivalents.

Futher the summary table of web-services and semantic web-services differences is given.

Table 1. Differences between web services and semantic web services

Items	**Web services**	**Semantic web services**
Service Requestor	Human	Agents
Service Publication	UDDI	May not be necessary
Description Language	WSDL(Service taxanomy)	Ontology-based language
Service Descriptions	Operations, messages and binding	Ontology-based Goals and Capabilities
Message Exchange	Syntactic-based	Semantic-based
Core technologies	UDDI/WSDL/SOAP	Ontology and Reasoning
Service Composition	Passive using WSBPEL	Goal-driven approach by agents
Choreography/ Orchestration	Passive using WSBPEL	Automatic generation by BPM
Heterogeneity	Passive	Semantic mediation
Service Discovery/ Selection	Search repository by human	Automatic discovery by agents
Transportation Protocols	SOAP	SOAP, and other XML protocols

Conceptual model WSMO is a top-level model which defines ontology, which is being used for business service modeling. It contains descriptions for ontology, web-services, goals, proxies [1].

Ontology provisions formal semantic definition for WSMO. Two of main distinguishes of ontology are distributed conceptualization principle and formal semantic.

Web services define functional capabilities and one or more number if interfaces which give for clients ability to invoke it. Service's facilities are being modeling by using pre-conditions and assumption about state of information space and external world, before running, outcoming post-conditions and results, which define state after service running. Service interfaces are being spitted to choreography and orchestration. Choreography defines invoking method and orchestration defines functional decomposition in terms of other services.

Goals presents request descriptions, which user wants to reach. WSMO goals are being described in request application domain terms of particular service. The Goal of WSMO are being characterized by request facility and invoke method.

Proxies present elements, which aim to resolution of structure, semantic or conceptual mismatch which are possible between different components in WSMO environment.

III. WEB 2.0 TECHNOLOGIES

In order to simplify interaction with sevices for an user irrelevant to IT the Web 2.0 technology has begun to be used along with SOA.

Web 2.0 is a set of new web technologies, which transform Internet to the platform for dynamic content creation, particularly collective and for the distribution of such sort of content. Using Internet in form of such kind of platform stimulates new generation of internet community and sort of hosting services as social networks, wiki resources, which is unleash user creativity and simplify collaboration and information exchange between them.

Business-value. Web 2.0 technologies help to create communities, which are supports customer loyalty consolidation. Moreover, those technologies help high-skilled employees which are not IT-specialists. They improve productivity of such specialists and boost their opportunities for collaboration, allow to create new objects like Enterprise Mashup by themself. Such variants of applying Web 2.0 not received wide distribution yet against simpler variants like blogs, social networks, and twitter-like applications, which became usual in our life [2].

Mashup technologies is the most perspective business appliance Web 2.0. It allows to high-skilled specialists to create situational applications by themselves. The standard building blocks of Mashup objects are SOA services.

Web 2.0 Technologies allows to reveal business services, which exist in environment and out of it; to combine them in mashup objects and deliver final result to end users.

IV. CONTEXT MANAGEMENT

Context Management has become another one full distributed approach's addition to SOA. Context Management is dynamic process which uses an data object in one application to present some information in another application which also includes that object [3]. For instance, in public healthcare when patient information has been received by one application it will be available in other application.

Infrastructure of Context Management. Management of context information is put into effect sharing this information between the concerned entities. Here applications provide low-level information context. Context-dependent services consume applications' and other services' incoming information in order to create their own high-level context which in its turn can be offered to the services too. So no one of the structures has full information about the context [4]. Figure 3 shows the process of information context interchanging between services and applications.

Each entity provides some part of combined Context Management functionality and is depicted as “Context Manager” block in figure 3.

These blocks can provide different functional possibilities:

- Context Sources which supply any concerned identity with context information. There are 2 types of them:
  1. Context Wrappers adopt raw application data in terms of one model.
  2. Context Reasoners interpret context requests and correlates heterogeneous models.
- Context Brokers monitor different sources and their context information.
- Context Stores presave information used for answering incoming requests.

It is important to notice that each functional organization has its own context model corresponding to the certain ontology [5].

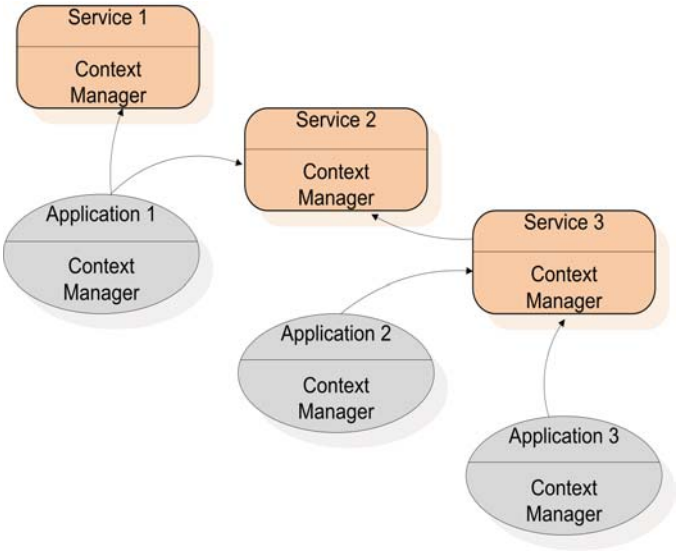

Figure 3. Context information flow

V. CONCLUSION

So both approaches: pure SOA and Full Distributed Approach were considered in this article. Each of reviewed technology contributes in development of smart environments which could be combined to create useful, flexible appliances for constant changing requirements. We can say that SOA in this approach took roots and capture allied domains by crossing with new technologies. Semantic web provides ability for integration and services collaboration by semantic languages. Web 2.0 technologies simplify user-service communication and allow to combine services on mashup approach and use only final result. Context management increase service usability, by avoiding multiple data input. That technologies provides decrease overload and increase quality of service. With future development of full distributed approach communication between human and computer will be more intellectual, easy and humane. Full Distributed Approach is the extent of evolving SOA approach which increases flexibility and usability of employing SOA architectured software.

REFERENCES

[1] V. Deretsky, “Application development in service oriented environment of semantic web”, UkrPROG, Kiev, Ukraine 2010.

[2] IBM Solutions Smart SOA Technical review “Understanding SOA meaning and modern IT innovations”, 2009.

[3] Krummenacher R., Domingue J., “SOA4All: Towards a Global Service Delivery Platform”, Semantic Technology Institute, University of Innsbruck, Austria, 2010.

[4] Ramparany F., Broens T., Pierson J., “Context Management and Semantic Modelling for Ambient Intelligence”, Centre for Telematics and Information Technology, Telematica Institut, 2005.

[5] Cadenas A., Ruiz C., Larizgoitia I. “Context Management in Mobile Environments: a Semantic Approach”, Polytechnical University of Madrid, 2009.