Journal of Network and Computer Applications 44 (2014) 134–151

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Review

Semantic web services: Standards, applications, challenges and solutions

Hassina Nacer [a,b,*], Djamil Aissani [c]

[a] Computer Science Department A/Mira University of Béjaia, Algeria
[b] Laboratory LIMED, Algeria
[c] Laboratory LAMOS University A/Mira of Béjaia, Algeria

ARTICLE INFO

Article history:
Received 28 February 2013
Received in revised form
7 December 2013
Accepted 30 April 2014
Available online 27 May 2014

Keywords:
Interoperability
Distributed middleware
Web services
Semantic web
Web services composition
Performance analysis

ABSTRACT

To share Knowledge coming from disparate and heterogeneous environments, we propose the use of semantic Web services to provide a common Knowledge format and meaning. Semantic Web services would require careful usage of combined technologies. On one hand, XML Web services technology because of their system agnostic nature and the ease of integration and, on the other hand, the semantic Web can define and link Web data in a way that it can be understood and used by software agents. This mixed technology may be a solution to functional interoperability, technical interoperabilty, semantic interoperability and flexible development in heterogeneous environments using the Internet as the main infrastructure. In this paper a survey of semantic Web services is realized to show that they ensure interoperability. Four aspects of Web services are presented: (1) Standards of XML Web services (eXtended Markup Language Web services) and their limits, (2) Languages and Tools of Semantic Annotation, (3) Web Services Composition, and (4) Performance Evaluation. Observations on some new challenges brought by semantic Web services are cited. Furthermore, a comparative study is presented, over the Internet and Intranet, based on numerical results using a discrete event between semantic Web services and distributed middleware, e.g. CORBA (Common Object Request Broker Architecture) and JAVA RMI (JAVA Remote Method Invocation).

Contents

* Corresponding author.
E-mail addresses: sino_nacer@yahoo.fr (H. Nacer), lamos_bejaia@hotmail.com (D. Aissani).

http://dx.doi.org/10.1016/j.jnca.2014.04.015
1084-8045/

1. Introduction

Globalization, cooperation and collaboration have substantially changed the software world and have contributed to the emergence of Knowledge sharing culture in open and large environments[1] (Beau et al., 2010; Sinderen, 2008). Actually, an individual human user cannot produce Knowledge and competencies but by collaboration with other human users and companies a rich source of Knowledge and competencies can be created. Thus, Knowledge representation becomes a central problem (Chein and Mugnier, 1992; Nonaka and Takeuchi, 1995; Davenport and Prusak, 1998; Gasmi et al., 2010). Knowledge should be used, shared, discovered and exchanged by any user (Human, Smart industrial device, Robot, Software agent, etc.) in heterogeneous and large environments. To solve the problem, a common format and meaning to represent a Knowledge is used. We refer to this design principle as "generic Knowledge". In order to realize, to represent and to save "generic Knowledge" over the Web, we propose the use of semantic Web services. These latter ones provide the necessary support by defining a standard mechanism for representing, publishing and locating Knowledge.

Semantic Web services would require a careful usage of combined technologies. On one hand, XML Web services technology because of their system agnostic nature, the ease of integration and communication and, on the other hand, the semantic Web can define and link Web data in a way that it can be understood and used by software agents. Semantics need to be introduced into the services so that functionalities can be identified on the meaning rather than on the syntax basis. Therefore, adding semantics enables structured information to be interpreted unambiguously. This mixed technology may be a solution for interoperability and flexible developments in heterogeneous environments (McIlraith et al., 2001).

To address interoperability problems in some identified areas of research (e.g. Resource Sharing, Competence Sharing, Negotiation, Web Services Discovery, Web Services Composition, Web Services Security and Web Services Interactions, Benna et al., 2008; Boudries et al., 2008; Nacer-Talantikite et al., 2009; Gasmi et al., 2010; Niu et al., 2011; Syncar et al., 2002; Nacer-Talantikite and Aissani, 2010; Zhou et al., 2011), standards-based approaches have been proposed such as CORBA, JAVA RMI and XML Web services (Nester et al., 1999; Orfali and Harkey, 1997; Soley and Stone, 1995; Group, 1998; Yu et al., 2008; http://www.omg.org; http://JAVA.sun.com/j2se/1.4.2/docs/guide/rmi/index.html; http://www.WebServices.org; http://www.ws-i.org). However, as a new distributed computing infrastructure, XML Web services technology is an effective mechanism for data and application integration over the Internet. It is characterized by its openness, dynamics, and loose coupling. It provides good support for resources sharing and cooperative works in heterogeneous environments. In addition, the arrival of the semantic Web represents a revolution for Knowledge description and storage.

The semantic Web was defined by Lee et al. (2001) as an extension of the current Web in which Knowledge is given a well-defined meaning by using ontologies (Charlet et al., 2003; Perez et al., 2004; http://www.daml.org/ontologies). These latter ones play an essential role in interoperability because they provide structured vocabularies that describe a formal specification of shared conceptualization (Heflin and Hendler, 2000; Wache, 2001; Cruz and Xiao, 2003; Lin and Ding, 2005; Tsai, 2007; Gordon et al., 2011). Ontologies contribute to resolve semantic heterogeneity by providing a shared comprehension of a given domain of interest. Furthermore, the main challenge of interoperability and data integration is still ontologies matching. The work in semantic Web demonstrates how ontologies can be used to address interoperability problems at the application level. Specifically, ontologies have been used during discovery to express the capabilities of services, as well as the requests for capabilities. Ontologies are used to improve communication between any user by specifying the semantics of the symbolic apparatus used in the communication process. More specifically, Jasper and Uschold (1999) identified three major uses of ontologies: (i) to assist in communication between human beings, (ii) to achieve interoperability among software systems, and (iii) to improve the design and the quality of software systems.

Semantic Web services were introduced first by McIlraith et al. (2001). Their goal was to provide a DAML-enabled agent programming capability that supports writing generic procedures for Web services-based task. And their vision was that agents will exploit users' constraints and preferences to help customize users' requests for automatic Web services discovery, execution, or composition and interoperation. But none of these ideas were entirely realized in 2001. However, these latter ones could be realizable nowadays. Semantic Web services can ensure interoperability at three levels regarding the most known distributed middleware proposed in the literature.

To our best Knowledge, there exist some research works, in the literature, that gave as overview containing several topics about Web services in the same paper in order to prove that Web services technology contributes to solve the problem of interoperability: Wang et al. (2004) studied four layers (Service Security, Service Composition, Service Semantic, and Grid Services) in order to show that Web services were designed to tackle the problem of integration of heterogeneous sources and make heterogeneous systems interoperable. However, Web services have shortcomings to fully satisfy the requirements of interoperability. Cardoso and Sheth (2005) discussed the Web process life cycle phases (Semantic Web Services Annotation, Semantic Web Services Advertisement, Semantic Web Services Discovery, Semantic Web Services Selection, Semantic Process Composition and Execution Web process). They showed that Web services promise universal interoperability and integration. The key to achieve this relies on the efficiency of discovering appropriate Web services and composing them to build complex processes. Dong et al. (2013) presented a survey of semantic Web Services matchmakers in order to obtain an overview of the state of the art in this research area and to show that Web services provide a standard means for the interoperable operations between electronic devices over the Internet. They summarized six technical dimensions from the past literature and analyzed the typical semantic Web Services matchmakers mostly developed during these years. Gayathridevi and Manikandan (2013) presented five layers (Standards, the Semantic Web, Representation of Semantic

[1] Large-scale virtual worlds like the Internet such as Pervasive networking environments and Internet.

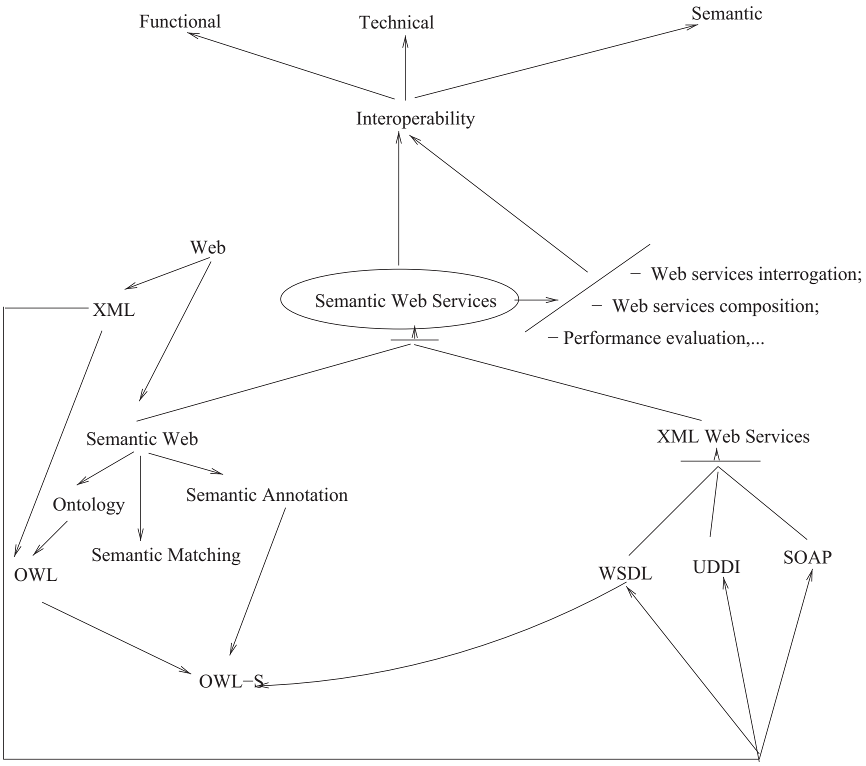

Fig. 1. Semantic Web services cartography: semantic Web services are a mixture of XML Web services (SOAP, WSDL, UDDI) and semantic Web technologies (ontology, semantic annotation). Semantic Web services are the result of the Web evolution in two directions: adding dynamic elements to the Web and enhancement of the syntactic description of Web services. They ensure interoperability at three levels: functional interoperability, technical interoperability and semantic interoperability. The most important applications of semantic Web services are interrogation, composition and performance evaluation. Furthermore, the motivation for these applications is more about solving interoperability problems.

Web Services, Semantic Web Services Composition, and Discovery). They showed that there is a lot of interest in finding ways to create an infrastructure where services could be described that should allow dynamic discovery, composition and invocation with even the Web services standards and recommendations becoming complex. The research around Web Services is intense; several other studies focus on a unique topic of Web services, such as survey on semantic Web services (http:\\www.w3.org/Submission/OWL-S) (Lee et al., 2001; Perez et al., 2004; Domingue et al., 2004; Mohebbi et al., 2012), survey on composition (Chakraborty and Joshi, 2001; Zheng et al., 2010; Bartalos and Bielikova, 2011; Portchelvi et al., 2012), and survey on comparison between middlewares (Juric et al., 2004; Gray, 2005; Narang et al., 2014; Artemio et al., 2012).

The motivation of the proposed survey is more about solving the problem of interoperability. Indeed, in the last few decades, the vision of semantic Web services moved toward this interoperability. Thus, our main contribution is to analyze the landscapes of benefits of semantic Web services. Thanks to this analysis, we gain an understanding of the current research and a summary of a series of potential issues, which would provide the foundation for future research in this area. In our survey, we have presented Web services, an emerging technology for the Web. Four aspects of Web services are presented: (1) Standards of XML Web services and their limits, (2) Languages and Tools of Semantic Annotation, (3) Web Services Composition, and (4) Performance Evaluation. They are critical to the success of interoperability in heterogeneous systems. For that, we propose a cartography of semantic Web services, see Fig. 1, which illustrates the link of semantic Web services with research areas and their impact on interoperability.

The paper is organized as follows. Section 2 is devoted to the survey of existing technologies and standards around Web services and semantic Web. A comparative study of semantic description languages is given. Section 3 covers research issues about Web services applications. Three comparison studies are given. Section 4 summarizes a comparative performance study between semantic Web services and other middlewares such as DCE (Distributed Computing Environment), JAVA RMI, COM/DCOM (Component Object Model /Distributed Component Object Model) and CORBA. Numerical simulations have been used to evaluate the results. Finally, in Section 5, we give concluding remarks, and we describe our future work briefly.

2. Standards

To achieve interoperability between organizations in heterogeneous environments, the integrated vision of a reference architecture is needed and associated standards must be agreed upon and cooperation needs of the organizations are taken into account. Indeed, the XML Web services standards (XML, UDDI (Universal Description, Discovery and Integration), WSDL (Web Services Description Language), and SOAP (Simple Object Access Protocol)) are the solution (Sinderen, 2008).

As commonly assumed in the literature, XML Web services are software components of distributed applications which provide services to other applications by using standard Internet technology.[2] They respect some properties, such as autonomous object components, loosely coupled, self-described, self-contained, reusable, modular, synchronous and asynchronous, that can be

[2] By using the standard Internet technology such as Internet protocols: SMTP (Simple Mail Transfer Protocol), HTTP (Hypertext Transfer Protocol), and FTP (File Transfer Protocol).

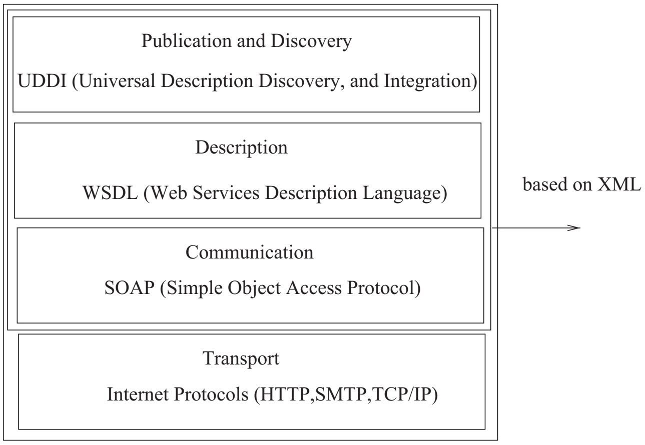
Publication and Discovery
UDDI (Universal Description Discovery, and Integration)
Description
WSDL (Web Services Description Language)
based on XML
Communication
SOAP (Simple Object Access Protocol)
Transport
Internet Protocols (HTTP,SMTP,TCP/IP)

Fig. 2. Reference architecture Web services: a Web service protocol stack is used to define, locate, implement and make Web services interact with each other. A Web service protocol stack typically stacks four protocols: (1) transport protocol (HTTP), (2) messaging protocol (SOAP), (3) description protocol (WSDL), and (4) discovery protocol (UDDI).

published, located, and invoked over the Web. Several definitions were advanced in the literature as follows:

According to Chappell and Jewell (2002), a Web service is a piece of business[3] logic, located somewhere over the Internet that is accessible through standard-based Internet protocols.

According to Benatallah et al. (2003), a Web service is an accessible application that other applications and humans can discover and invoke. The authors cited the following properties to an XML Web service: independent of specific platforms and computing paradigms, and easily composed.

The W3C group defined a Web service as being a software application identified by a URI, whose interfaces and bindings can be defined, described, and discovered as XML artifacts. Thus, Web services can tackle the challenge of heterogeneous sources and interoperability.

2.1. *Reference architecture*

XML Web services technology is SOA (An authority of Architecture Oriented Service) which is an architectural style to re-use and integrate sub-systems in existing systems in order to create new applications (Portchelvi et al., 2012; Papazoglou and Georgakopoulos, 2003; Willy; Karimi, 2011; Balasubramanian and Ruba, 2012; http://www.w3.org). SOA enables flexible integration of applications and resources by (i) representing each application or resource as a service with standardized interface, (ii) enabling a service to exchange structured information (Messages, Documents, Business objects), and (iii) coordinating and mediating between services (Dodani, 2004). SOA is based on a “service” concept. The major goal of SOA is to reverse the tendency which generally shows that the operational processes inside enterprises adapt to the imposed constraints by data processing. XML Web services architecture is based on SOA, and it takes the same actors. The provider of services publishes a contract of interface, defines the functionality and the execution of a Web service. The client consumes the service and uses a universal registry to discover available services. Once a service is located, the client extracts the interface contract in order to execute a service. The registry of services is a virtual database of available services. Each provider publishes a contract of an XML Web service interface in the registry with necessary information (Localization, Access Control, etc.)

The basic infrastructure of XML Web services is illustrated in Fig. 2.

The XML Web services technology was concretized around the specification of the W3C. It is divided into three areas: (1) Communication Protocols (SOAP), (2) Service Description (WSDL), and (3) Service Discovery (UDDI). These current standards revolve around XML to achieve platform independence features. We illustrate each standard of XML Web services technology as follows.

2.1.1. *XML*

XML which is a standard of W3C is a universal model of data representation and exchange. It is a simple format text, flexible and also independent of any manufacturer. Adding to this, it gives structure to documents and data (Harold and Means, 2004). It is extracted from SGML language[4] and it benefits from experiences of HTML's use (HyperText Mark-Up Language). Furthermore, XML offers portable and structured data on heterogeneous structure and programming languages. XML brings the following criteria to XML Web services architecture:

- *Extensibility*: A system can function correctly without losing its main properties during an update.
- *Neutrality*: The required constraints of an application are limited.
- *Structure*: XML represents both document structure and content, offers increased control of information granularity through transformation and query languages.
- *Interoperability*: Communication and data exchange between heterogeneous systems are possible.

2.1.2. *SOAP: simple object access protocol*

The SOAP protocol is an exchange message's process in heterogeneous environments for application-to-application communication based on XML and on standard protocol HTTP (Kadima and Monfort, 2004). SOAP, a standard of W3C, defines a set of rules to structure dialogs RPC (Remote Procedure Call) to exchange data. It ensures interoperability between components independent of transport mechanisms, operating systems and programming languages.

According to Harold and Means (2004), SOAP is a flexible protocol to connect distributed systems. The purpose of this protocol is to facilitate the access to software services to any user through the Internet.

2.1.3. *WSDL: Web services description language*

WSDL is a formal language of Web services description according to the standard XML. A WSDL file describes the functionality (Methods, Parameters) and the localization of a Web service (URI, Port, Protocol of invocation). According to W3C (http://www.w3.org/TR/wsdl), WSDL separates the description of abstract functionalities offered by a service from concrete details of service description. As in programming languages, a type signature defines the inputs and outputs for a function. It means that WSDL can be seen as a traditional function, subroutine or method.

2.1.4. *UDDI: universal description, discovery and integration*

The registry of Web services “UDDI” is a virtual database of existing XML Web services. It is similar to a CORBA trader and can

[3] It means that UDDI provides a universal registry for business to provide service listings (Web services description).

[4] Standard Generalized Markup is a meta-language which makes it possible to define markup languages.

be considered as a DNS service for business applications. On one hand, it allows providers of services to record XML Web services under a standardized format and on the other hand, it concentrates on discovery process of XML Web services satisfying services' needs in SOA. UDDI becomes an intermediate standard between providers and clients through the Internet and it is a recommendation of W3C. According to Chappell and Jewell (2002), the UDDI project is an initiative of industry which tries to create an independent platform, to describe services, to discover businesses and to integrate services. It means that UDDI provides a universal registry for business to provide service listings (Web service description).

2.2. Observations

Although, the elementary XML Web services infrastructure, mainly based on syntactic standards (XML, UDDI, SOAP, WSDL), is significant and seems to play an important role in interoperability and it gives a new dimension to the co-operation and the collaborative work, these standards are not sufficient enough for the following research issues: Data Exchange, Competencies Sharing (Gasmi et al., 2010; Bouchaib et al., 2010), Communication machine to machine (Zhou et al., 2011), Web Services Discovery, Web Services Composition (Nacer-Talantikite et al., 2009), Web Services Selection (Mohd et al., 2011; Guangjun and Fei, 2011), Web Services Interrogation (Benna et al., 2008), Web Services Security (Nacer-Talantikite and Aissani, 2010; Kagal et al., 2004; Story et al., 2009), etc. Both intelligent human properties and powerful and structured mechanisms, in the current Web, are needed. It is obvious that there is a lack of semantics in XML Web services infrastructure. Nevertheless, recent developments in semantic Web provided better and new solutions to issues unsolved by the XML Web services technology. XML Web services related to semantic annotations become Semantic Web services (SWS).

2.3. Semantic annotations

Semantic Web services are at the convergence of two significant fields of research which are technologies of the Internet and XML Web services. The purpose of semantic Web services is to create a semantic Web of services whose properties, interfaces and effects are described in a non-ambiguous and exploitable way by software agents. The word "semantic Web" (Lee et al., 2001) refers to the vision of the future Web like a vast space of exchange of resources containing great volumes of information and interaction between humans and software agents allowing an effective exploitation of resources. The main goal of the semantic Web is to extend current interfaces oriented to human understanding in a standardized format interpretable by software agents.

2.3.1. Elementary definitions

- *Meta-data*: It is data on data. This word defines the included data in documents in order to provide information on electronic resources. This information is treated by search engines to satisfy the user's needs. Meta-data can be used not only for describing content, but also for organizing and classifying.
- *Ontology*: The word ontology comes from philosophy, which means "the Knowledge of what is to be in oneself" (Charlet et al., 2003; Perez et al., 2004; http://www.daml.org/ontologies). In data processing, ontology indicates a structured set of Knowledge in a domain of interest. Ontology is conceptualization of an application domain in a human-understandable and machine-readable form, and typically comprises the classes of entities, relations between the entities and the axioms which apply to the entities which exist in that domain. Several definitions have been made, of which the most used is the one given by Gruber in Perez et al. (2004). Several operations are possible on ontology, such as Mapping, Matching, Alignment, Transformation, Merging and Integration.
- *Semantic annotation*: An annotation assigns to an entity, which is in the text, a link to its semantic description. A semantic annotation is referred to an ontology. The idea is to have data through the Web defined and linked in such a way that its meaning is explicitly interpretable by software processes rather than just being implicitly interpretable by humans. Semantic annotation can be applied to any resource (File, Image, Web page, etc.).

2.3.2. Semantic description languages

Languages which support semantic meta-data representation are required so that any resource over the Web becomes accessible by any user. Various works about semantic description have already been proposed like RDF (Resource Description Framework) (Baget et al., 2003), RDFS (RDF Schema) (Charlet et al., 2003), OIL (Ontology Inference Layer), DAML (Darpa Agent Markup Language (Lacot, 2013), OWL (Web Ontology Language) (http://www.daml.org/ontologies), OWL-S (Ontology Web Language for Services) (Martin et al., 2004; http://www.w3.org/Submission/OWL-S), WSML (Web Service Modeling Language) (Domingue et al., 2004; http://www.wsmo.org/

Table 1
Comparison between semantic description languages.

Approach	Resource	Property	Concept	Research
RDF	Any URI	Functional	XML, Triple (object, attribute,value)	Knowledge engineering, semantic Web (WSem), etc.
RDFS	Any URI	Functional Non-functional	RDF, classes and sub-properties, constraints	WSem, categorization, indexing, etc.
OIL	Any URI	/	Frame, DL	WSem, intelligence artificial, inter-ontology relations, etc.
DAML+OIL	Any URI	/	XML-S, RDFS extended RelationShip, DL	Ontology sharing, ontology construction vocabulary, etc.
OWL	Any URI	Functional Non-functional	W3c's adaptation of DAML+OIL	WSem, interoperability knowledge sharing, capabilities, etc.
OWL-S	WSDL1.0 WSDL1.1	Functional Non-functional	OWL ontology	WS discovery, composition, execution, monitoring, etc.
WSML	WSDL1.0 WSDL1.1	Functional Non-functional	Logical formalisms	Interoperability, WS, etc.
WSMO	WSDL1.0 WSDL1.1	Functional Non-functional	WSML ontology	WS discovery, composition, etc.
WSDL-S	WSDL1.2 UDDI3.0	Functional Non-functional	OWL ontology	WS discovery, composition, etc.
SAWSDL	WSDL1.1 WSDL2.0 XML-S	Functional	Independent of any semantic model	WS discovery, invocation, WS-policy, BPEL, etc.

Table 2
Comparison between semantic annotations tools.

Tools	Interpretation	Type	ResType
Annotea	Human	Manual	HTML,XML
Yawas	Human	Manual	HTML,Text,XML
Shoe	Software agent	Manual	HTML
Aero-Daml	Software agent	Automatic	Text
Smore	Human, software agent	Manual	HTML, email, image
Kim	Human, software agent	Automatic	Structured documents or no
Cohse	Human, software agent	Manual, automatic	HTML
Mnm	Human, software agent	Manual, semi-automatic, automatic	HTML
OntoMat	Software agent	Manual, semi-automatic	HTML
Meteor-S	Software agent	Manual, semi-automatic	XML, WSDL, UDDI

Table 3
Comparison between semantic annotations tools.

Tools	Language	Storage	Ontology
Annotea	RDF	External	RDF
Yawas	Text	Local	DataBase
Shoe	HTML	Local	Shoe
Aero-Daml	DAML+OIL	External	DAML+OIL
Smore	RDF, OWL	External	RDF, OWL
Kim	RDF, OWL LITE	External	PROTON, KIMLO, KIMSO
Cohse	RDF, DAML+OIL	External	DAML+OIL
Mnm	RDF	External	RDFS, DAML+OIL, KMI, OCML
OntoMat	RDF	Local, external	OWL
Meteor-S	JAVA	External	RDFS, DAML, OWL

Table 4
Comparison between semantic annotations tools.

Tools	Technology	Extraction
Annotea	HTTP servers, annotation interface	/
Yawas	JAVA using DOM	/
Shoe	Autonomous application in JAVA	/
Aero-Daml	Autonomous application	Aero text and training
Smore	Specific server	/
Kim	GATE, SESAME, LUCENE, Web user interface, etc.	KIM KB+LUCENE
Cohse	Complexe architecture	Cohse agent DLRS,training
Mnm	Complexe architecture	AMICLARE engine
OntoMat	Autonomous application in JAVA	AMICLARE engine
Meteor-S	/	Several matching algorithms

wsml), WSMO (Web Service Modeling Ontology) (http://www.wsmo.org), WSDL-S (Web Services Description Language Semantic) (Akkiraju et al., 2005) and SAWSDL (Semantic Annotations for Web Services Description Language) (http://www.w3.org/TR/2007/REC-sawsdl-20070828). Table 1 illustrates a comparative study between the above semantic description languages regarding the following criteria:

- *Resource*: It identifies the resource type over the Web (a Web page, a Web service, etc.). Each Web resource can be identified by a URI.
- *Property*: It identified the used property type in the document: Functional properties (such as Resource, Inputs), or non-functional properties (such as Implementation, RelationShip).
- *Concept*: It describes the basic property of the semantic model if it exists or not (RDF, DL[5] (http://dl.kr.org), XML-S[6], etc.).
- *Research*: It specifies the research area.

2.3.3. *Semantic annotations tools*

A comparative study of several tools of semantic annotations is presented in Tables 2–4. We take into account the following most popular tools: Annotea (Kahan et al., 2001), Kim (Popov et al., 2003; Kiryakov et al., 2005), Yawas (Denoue, 2013), Shoe (SHOE, 2013), Smore (SMORE, 2013), Cohse (COHSE, 2013), MnM (Vera et al., 2002), OntoMat (Annotizer, 2013) and Meteor-S (Patil et al., 2004). This study shows that the systems of semantic annotations vary in their architecture, tools for information extraction, methods and annotations' languages. These tools depend on their use (Collaboration, Search, Integration, etc.) and they do not gather all the performances and the expected properties for an annotations' system. Furthermore, we can find in the literature other tools which concentrated specially on Web services annotation as Meteor-S tool. These tools are ASSAM (Hess et al., 2004), EIDOS (Carman and Knoblock, 2007), EDITOR (Belhajjame et al., 2008), etc. In addition to these tools, many plug-ins provide user's interface to annotate Web services (from WSDL to OWL-S). The characteristics' list of the following plug-in is described in a call to candidature for a recommendation by W3C (http://www.w3.org/TR/2007/REC-sawsdl-20070828):

1. Woden4SAWSDL and Radiant of Laboratory Lumina of University Georgia.
2. WSMO4J Grounding, WSMO Studio, WSMO Grounding and WSMO-Lite of DRI.
3. SAWSDL4J of any specific organization.
4. OWL-S of SAWSDL prospect.
5. Semantic Tools for Web services of IBM Alpha Works, RDF Mapping Implementation.

The above semantic annotations tools were studied according to the following criteria:

- *Interpretation*: It identifies the annotation's user (human, software agent).
- *Type*: It defines the operation of the semantic annotation process (manual, semi-automatic, automatic).
- *ResType*: It defines the annotated resource type (HTML, Text, etc.).
- *Language*: It describes the programming language.
- *Storage*: Annotations can be stored either locally (Local) or in annotations' servers (External).
- *Ontology*: It describes the semantic model (RDF, OWL, etc.).

[5] Description logic.
[6] XML Schema.

Table 5
The advantages and the drawbacks of the different semantic annotations tools.

Tools	Advantages	Drawbacks	Application cases
Annotea	Semi-structured documents External ontologies	Manual	User's collaboration over the Web
Yawas	Any kind of documents	Manual Local ontologies	Search engines
Shoe	/	Manual HTML Local ontologies	Creating ontoglogies, adding semantics to Web pages, describing Web resources
Aero-Daml	Automatic External ontologies	Text	Content analysis applications, graphical tools, application knowledge bases
Smore	Multimedia External ontologies	Manual	
Kim	Automatic Any kind of documents External ontologies	/	Knowledge management, classification, filtering application
Cohse	Hybrid External ontologies	HTML	Automation of link generation, distributed information management, keyword recognition
Mnm	Hybrid External ontologies	HTML	Training data, information extraction engine
OntoMat	Hybrid External ontologies	HTML	A wide range of applications in the semantic Web
Meteor-S	Semi-automatic Structured documents External ontologies	/	Adding semantics to WSDL files

- *Technology*: It specifies the used technology (GATE (GATE, 2013) SESAME,[7] etc.).
- *Extraction engine*: It describes the information extraction engine and training (LUCENE,[8] AMICLARE (AMICLARE, 2013), etc.)

At the end of this comparative analysis between the different semantic tools, we summarize the advantages, the drawbacks and the applications cases for each tool in Table 5.

3. Applications: Web services interrogation, functional properties based Web services composition and QoS based Web services composition

As a new technology, Web services paradigm is more than just a distributed system, it provides the software foundation for next generation companies. It offers companies the capabilities to integrate distributed Web services into a business process, for example, interrogation, discovery, composition and selection of Web services.

3.1. Web services interrogation and exploration

We present in this section a comparative study between several languages of interrogation Web services such as LARKS (Language for Advertisement and Request for Knowledge Sharing) (Syncar et al., 2002), SwellQ (Condack and Schwabe, 2005) and XSRL (XML Service Request Language) (Papazoglou et al., 2002) in Table 6 according to the following criteria:

- *Concept*: The basic principle of the language (Frame, Xquery, etc.).
- *Property*: It defines the properties (Functional properties such as Inputs, Outputs or non-Functional properties such as Implementation and QoS).
- *Semantic*: It defines if the service is described semantically (Yes or No).
- *Model*: It specifies the used model such as ITL,[9] LOOM (LOOM, 2013) and KIF,[10] or a syntactic model such as WSDL.
- *Composition*: It specifies if the composition is possible (Yes or No).
- *Result*: It specifies the result of the request.

3.2. Synthesis of web services composition

An automatic and dynamic Web services composition is a highly complex task. On one hand, the proposed standards (XML, WSDL, UDDI, and SOAP) of XML Web services technology do not answer the problems of Web services discovery and composition by a software agent. And on the other hand, the semantic annotations for Web services and requests are not yet mature. To understand the Web services composition, it is instructive to consider the basic definition of a composition. Several available Web services, developed by different providers, may be combined to create a Virtual Web service. According to Tari et al. (2009), Virtual services provide a different view of existing services, a view that can be useful to provide more availability for an existing service, or to create a specialized service for a particular application. According to Gardarin (2002), Web services composition is a technique which assembles Web services in order to achieve a particular goal, via primitives of control (Test, Treatment of Exception, etc.) and exchange (Sending and Reception of Messages). According to Fensel et al. (2002), the composition is a process which functions, in an intelligent way, in order to discover services automatically, to negotiate between them and to compose them in more complex services.

3.2.1. Properties of composition

Web services composition includes two processes:

1. *Discovery of user's goals*: A process which translates the goals of a user into precis and formal goals;

[7] RDF Registry.
[8] Search Engine of textual information.
[9] Information Terminological Language.
[10] Knowledge Interchange Format.

Table 6
Comparison between Web services interrogation languages.

Language	**SwellQl**	**Approach based** on LARKS	**XSRL**
Concept	Interrogation of an ontology	Use of frame	Exploitation of Xquery or AI Planing
Property	Functional Non-functional	Functional Non-functional	Attributes, property of XML
Semantic	Yes	Yes	No
Model	SwellOnt Ontology, other	ITL, LOOM, KIF languages	WSDL
Composition	No	Yes	Orchestration
Result	Concepts of SwellOnt Ontology	/	XML document

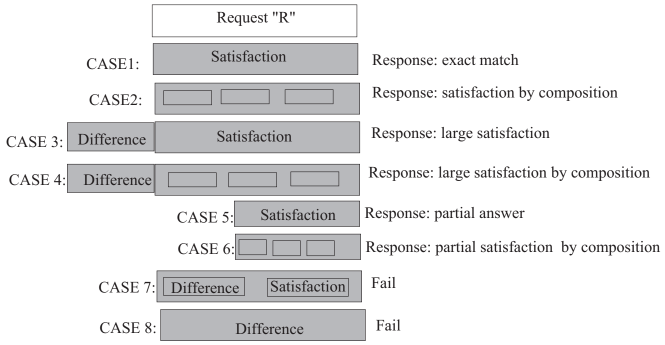

Fig. 3. Various possible answers to a submitted request: the matching between the request R and the discovered Web services gives eight cases. *Case* 1: an exact simple Web service that satisfies the request, *Case* 2: an exact composite Web service that satisfies the request, *Case* 3: a simple Web service containing addition information with regard to the request, *Case* 4: a composite Web service containing addition information regarding the request, *Case* 5: a simple Web service which does not satisfy completely the request, *Case* 6: a composite Web service which does not satisfy completely the request, *Case* 7: a simple Web service contains a part that satisfies the request and a part that does not satisfy the request, and *Case* 8: a simple Web service that does not satisfy the request.

2. *Composition of discovered services*: A process which meets the awaited needs for a user.

The response to a request's user may be one of the eight following cases; see Fig. 3

- *Case* 1: The user's goals are completely covered by a single Web service (the user's needs and the announced possibilities of the single service are perfectly matched).
- *Case* 2: The user's goals are completely covered by several Web services (the user's needs and the announced possibilities of services composition are perfectly matched).
- *Case* 3: The user's goals can be completely covered by a single Web service. However, the user can receive objects which are not suitable for him (Opposite subsumption).
- *Case* 4: The user's goals can be completely covered by several Web services. However, the user can receive services which are not suitable for him (Opposite subsumption).
- *Case* 5: The user's goals cannot be completely covered by a single Web service (the user's needs and the announced possibilities of the single service are partially equivalent (Subsumption)). Nevertheless, the discovered service does not provide any non-suitable object for the user.
- *Case* 6: The user's goals cannot be completely covered by several Web services (the user's needs and the announced possibilities of several services are partially matched). Nevertheless, the discovered services do not provide any non-suitable object for the user (Subsumption).
- *Case* 7: The user's goals are completely covered by a single Web service. Nevertheless, the discovered service provides non-suitable object for the user.
- *Case* 8: The user's goals are disjoined of the announced possibilities of all existing Web services.

3.2.2. *Literature review on Web services composition*

Like all Internet based computer applications, composition Web services are some of the most common concerns for interoperability and distributed middleware. There have been several research efforts on Web services composition. We surveyed the Web services composition development literature from these last years. Many industry standards have been developed, such as BPEL4WS (Business Process Execution Language for Web Services) and BPML (Business Process Modeling Language). BPEL4WS provides a language for the formal specification of business processes and business interaction protocols. BPML is an XML-based meta-language developed by the Business Process Management Initiative (BPMI) as a means of modeling business processes. BPML supports advanced semantics such as nested processes and complex compensated transactions that are not addressed by BPEL4WS. Besides business process standard specifications proposed by industry, many academic research activities have also been resolved by various models: Petri Nets (Hamadi and Benatallah, 2003), Logical Programming (Lammermann, 2002; Rao and Su, 2004; Waldinger, 2001), Markov Process (Prashant et al., 2005), Matching Algorithm or Chaining (Hull and Su, 2005; Medjahed et al., 2003; Sirin et al., 2003; Han et al., 2008), AI Planning (Peer, 2004; Sirin et al., 2004), Graphs (Hashemian and Mavaddat, 2005), Semantic Network (Nacer-Talantikite et al., 2009), States Machine or Finite States Automaton (Berardi, 2005; Bultan et al., 2003; Esmaeilsabzali, 2004), Workflow techniques (Casati et al., 2001; Georgakopoulos et al., 2002), Genetic Algorithms (Michael and Gero, 2007; Canfora et al., 2005), and KP (Tao et al., in press; Tao and Kwei-Jay, 2005).

Figure 4 shows our proposed classification of the most known approaches of Web services composition.

Functional properties based Web services composition: We limit ourselves to some existing techniques and we present a comparative study in Table 7 according to the following criteria:

- *Interaction* (*I*): It indicates the used model of interaction (Behav (Behavioral),[11] atomic[12]).
- *Representation* (*R*): It illustrates the used formalism to model a Web service (Tree, Petri Nets, etc.).

[11] Web services, based on a behavioral model, are often known under the name of “gray box”, i.e. they are described by the ordering of the execution of their operations (Berardi, 2005).

[12] Atomic Web services are seen as “black box entities”, i.e. they are described by their parameters (Inputs, Outputs, Conditions and Effects) (Sirin et al., 2003).

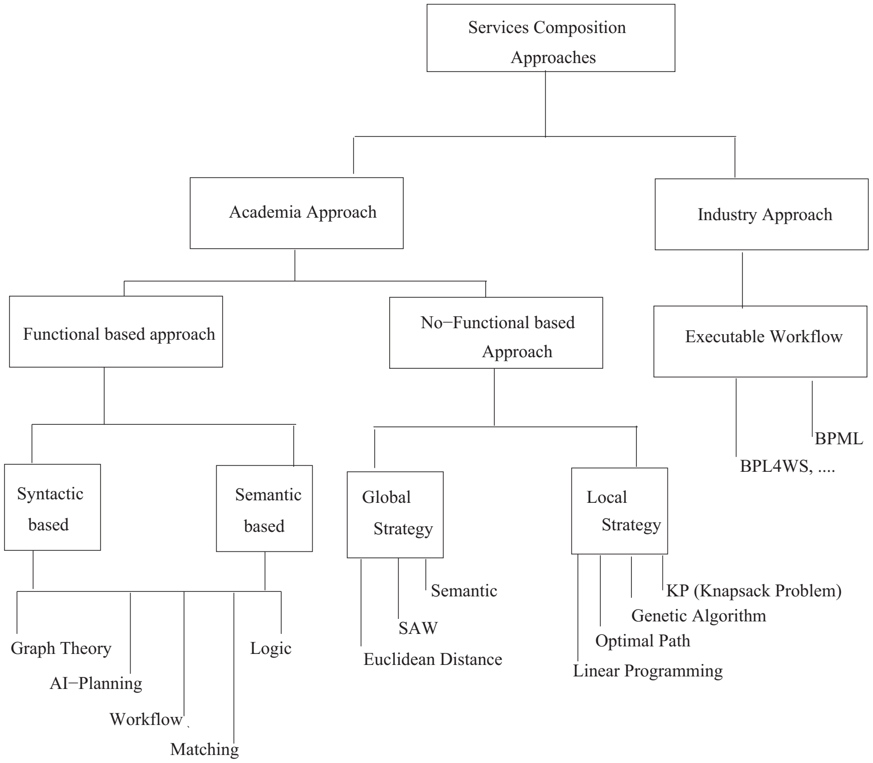

Fig. 4. Classification of Web services composition approaches. Two big classes are defined in Web services composition: industry based approaches and academia based approaches. A tremendous number of research papers had been published, focusing on different aspects of the Web services composition. The research tends to focus on issues related to the non-Functional properties and Functional properties. We divided them into two groups: first, approaches considering only the syntactic description and second, approaches considering semantic description. These usually deal also with a specific model of interaction such as graph theory, AI-Planning, Workflow, Matching, and Logic. The approaches based on non-functional properties concern the quality of the execution process affected by the QoS (Quality of Service) attributes which are non-functional properties of services.

- *Description* (*Dp*): It explains the used description language (Syntactic, Semantic).
- *Discovery* (*D*): It specifies if the step of Web services discovery is included or not.
- *Algorithm* (*A*): It describes the used tools to assemble Web services.
- *Typology* (*T*): It defines the composition type.

Note: DPDL: Deterministic Propositional Dynamic Logic, Structure: Structure Property of Petri Nets, Exploration: Algorithms of graph exploration, Conversation: Flow of interaction between services, Chaining: Algorithms of chaining, Composability: Rules of composition, Hybrid: Structure property of semantic networks and chaining algorithm.

QoS based Web services composition: For composite services, one of the QoS issues is to identify the optimal service selection to meet user's QoS requirement (User's constraints). Several Web services may share similar functionalities, but possess different non-functional properties. When discovering and composing Web services, it is essential to take into account non-functional properties. QoS based Web services selection, which is a part of Web services composition problem, is a multi-criteria decision mechanism that requires Knowledge about services and their QoS description. This problem is known to be NP-hard (Lee, 2003; Zeng and Benatallah, 2004; Canfora et al., 2005). Quality of a Web service is a set of non-functional attributes that may impact the quality offered by the Web service. Each QoS attribute is measured by one or more constraints. It can represent the execution time of a service, the cost of a service, even the available security features, etc. QoS research for Web services is an active research area (Alrifai et al., 2009; Bartalos and Bieliková, 2009, 2010; Jiang et al., 2010; Tao et al., in press; Tao and Kwei-Jay, 2005; Michael and Gero, 2007; Canfora et al., 2005). Many works from both industry and academia have addressed the QoS management for Web services using different algorithms.

Moreover, we give a comparative study of these approaches in Table 8 according to the following criteria:

- *Strat*: It defines the strategy of Web services selection (Local optimization, Global optimization).
- *Sem*: QoS ontology (Semantic Quality) is established to overcome the ontological conflicts that may occur between users and providers. QoS ontology defines the QoS properties and their relationships, and it establishes shared conceptions between users and providers.
- *C_Quant*: Quantitative QoS may include a number of non-functional properties such as cost, response time, availability and reputation.
- *C_Qualt*: Qualitative QoS may include a number of non-functional properties annotated semantically by an ontology.
- *Contr*: The approach's ability to accept both kinds of users' constraints (Equality, Inequality) is a key to satisfying user's requirements.
- *Algorithm*: The used algorithm in selection process (SAW (Simple Additive Weighting), Exhaustive Search, etc.).

In the literature, it has been stated that current Web services composition does not have a reference model. Several classification works of Web services composition can be found in several survey papers (Chakraborty and Joshi, 2001; Medjahed, 2004; Rao and Su,

Table 7
Synthesis of functional properties based Web services composition approaches.

Ref.	I	R	Dp	D	A	T
Casati et al. (2000, 2001)	Atomic	Process	Syntactic	Yes	E-flow	Dynamic
Waldinger (2001)	Atomic	Theorem	Syntactic	/	Program synthesis	Automatic
Georgakopoulos et al. (2002)	/	Business Process	Syntactic	/	Multi-enterprise process (MEP)	Dynamic
Lammermann (2002)	Atomic	Propositional Logic	Syntactic		Structural synthesis of programs (SSPs)	Automatic
Hamadi and Benatallah (2003)	Behav	Petris-Nets	Syntactic	No	Petri Net based algebra	Automatic
Medjahed et al. (2003), Medjahed (2004)	Atomic	Direct graph	Semantic	Yes	Composability model	Automatic
Sirin et al. (2003)	Atomic	Service profile	Semantic	No	Inference engine based matching	Semi-automatic
Benatallah et al. (2003)	Atomic	Interface	Semantic	Yes	Conversation	Automatic
Bultan et al. (2003)	Behav	Mealy machine (FSM)	Syntactic	No	Conversation	Automatic
Sirin et al. (2004)	Atomic	Process	Semantic	No	HTN Planning	Automatic
Peer (2004)	Atomic	PDDL	Semantic	No	AI-Planning	Automatic
Esmaeilsabzali (2004) Berardi (2005)	Behav	FSM Automaton	Syntactic	No	Synthesis of FSM	Automatic
Aversano et al. (2004)	Atomic	/	Semantic	Yes	Matching	Automatic
Arpinar et al. (2005)	Atomic	Interface	Semantic	Yes	Matching	Semi-automatic
Gerber and Schmidt (2005)	Atomic	PDDL	Semantic	No	AI Planning, Xplan	Automatic
Shin and Lee (2007)	Atomic	Graph	Semantic	/	Graphsearch	Automatic
Nacer-Talantikite et al. (2009)	Atomic	Semantic networks	Semantic	Yes	Inference engine	Automatic
Han et al. (2008)	Atomic	Process	Semantic context	Yes	/	Automatic
Gehlot and Edupuganti (2009)	Atomic	Colored Petri Nets (CPNs)	Syntactic	No	CPN tools	Automatic
Liu et al. (2010)	Atomic	DDL[a]	Semantic	Yes	Matching	Automatic
Yue et al. (2010)	Atomic	Direct weighted graph	Syntactic	No	Matching	Automatic
Durcik (2010)	Atomic	PDDL	Semantic	No	AI Planning	Automatic
Lee et al. (2010)	Atomic	/	/	No	AI Planning	Automatic
Cherifi et al. (2011)	Behav	Graph	Syntactic	Yes	Matching	Automatic
Vadivelou et al. (2011)	Atomic	Graph	Semantic	Yes	Graph Search	Automatic
Hashemian and Mavaddat (2005), Elmaghraoui et al. (2011)	Atomic	Graph	Semantic	Yes	Graph Search	Automatic
Samuel and Sasipraba (2011)	Atomic	Graph	Syntactic	Yes	Weighted planning graph	Automatic
Henni and Atmani (2012)	Atomic	PDDL	Semantic	No	AI Planning	Automatic
Cherifi et al. (2013)	Behav	Graph	Syntactic	Yes	Graph Search	Automatic

[a] Dynamic description logic.

Table 8
Synthesis of non-functional properties based Web services composition approaches.

Ref.	Strat	Sem	C_Quant	C_Qualt	Contr	Algorithm
Jaeger and Muhl (2006)	Local	No	Yes	No	No	SAW
Liu et al. (2004)	Local	No	Yes	No	No	SAW
Taher et al. (2005)	Local	No	Yes	No	No	Euclidean distance
Huang et al. (2009)	Global	No	Yes	No	No	Linear programming
Mohd et al. (2011)	Global	No	Yes	No	No	SAW + Optimal path
Li et al. (2007)	Global	No	Yes	No	Yes	H_MCWS
Tao et al., in press and Tao and Kwei-Jay (2005)	Global	No	Yes	No	Yes	MCSP, MCSP-K
Tao and Kwei-Jay (2004)	Global	No	Yes	No	One constraint on the cost	Exhaustive search
Tao and Kwei-Jay (2004)	Global	No	Yes	No	One constraint on the cost	Dynamic programming
Tao and Kwei-Jay (2004)	Global	No	Yes	No	One constraint on the cost	Pisinger
Tao et al., in press	Global	No	Yes	No	Yes	BBLP
Tao et al., in press	Global	No	Yes	No	Yes	WS_HEU
Michael and Gero (2007)	Global	No	Yes	No	Yes	Genetic algorithm
Canfora et al. (2005)	Global	No	Yes	No	Yes	Genetic algorithm
Geetha and Sathyavathy (2005) and Rajendran et al. (2010)	Local	Yes	Yes	Yes	Yes	Matching
Guangjun and Fei (2011) and Wang et al. (2006)	Local	Yes	Yes	Yes	Constraint of equality	Matching + SAW
Ren et al. (2007)	Local	No	Yes	No	No	SAW

2004; Dustdar and Schreiner, 2005; Bucchiarone and Gnesi, 2006; Zheng et al., 2010; Bartalos and Bielikova, 2011; Portchelvi et al., 2012; Cherifi et al., 2013). Nevertheless, these classifications are either incomplete or ambiguous. Firstly, none of the existing classifications are based on the representation model of a Web service, on the model of interaction in the Composition, on the functional based approaches, or on the non-functional based approaches. Secondly, the terminology is vague in some composition classifications. And finally, the lack of clear classification targets is the most significant weakness of the existing classification works of Web services composition. According to Chakraborty and Joshi (2001), the composition approaches can be classified into (i) Proactive composition, which is a composition 'out-line' of available services, assembled in advance in order to create new services, (ii) Reactive composition which refers to a process of virtual Web services creation on-line, (iii) Mandatory composite service which corresponds to the class of virtual Web services, where all the component services must take part in the execution of virtual services and (iv) Optional composite services which allow us to satisfy the user's request without the participation of some services. According to Medjahed (2004), four kinds of compositions are identified: (i) Static composition and (ii) Dynamic composition, which concerns the moment when Web services are composed, (iii) Manual composition and (iv) Automatic composition, which defines the way in which the composition is obtained. In Rao and Su (2004), Rao et al. evaluated the Web services composition frame works based on the concept of services composition life-cycle. In Dustdar and Schreiner (2005), Dustdar et al. presented several different composition strategies, based on some currently existing composition platforms and frameworks. This classification represents the first implementations of state-of-the-art technologies. In Bucchiarone and Gnesi (2006), Bucchiarone et al. focused more on Web services composition languages. In Zheng et al. (2010), Zheng et al. proposed an effort-oriented classification matrix for Web services composition, which distinguishes between the context and the technology dimension. The context dimension is aimed at analyzing the environment influence on the effort of Web services composition, while the technology dimension focuses on the technique influence on the effort. In Bartalos and Bielikova (2011), Bartalos and Bielikova provided an overview of several existing approaches dealing with the basic problem of Web services composition. They divided them into three groups: (i) approaches considering only the I/O during services chaining, (ii) approaches considering additional meta-data of services, and (iii) approaches that allow defining also soft constraints to the composition goal or used services. In Portchelvi et al. (2012), Portchelvi et al. classified the approaches into several groups based on various composition development aspects: (i) Industry approaches and (ii) Academia approaches which contain syntactic and semantic approaches. These latter ones are also classified into AI planning and Agent technology.

In this paper, we put an effort in presenting a comprehensive survey of Web services composition. This has been done by identifying various groups of techniques. The proposed classification enabled us to understand the evolution of the Web services composition approaches. We propose that Web services composition should be studied from a software engineering point of view: Academia, Industry, Functional, non-functional, Syntactic, Semantic, and Model of Web service, Model of interaction (Graph Theory, AI-Planning, Workflow, Logic, Matching, etc.). Furthermore, we identified a number of research questions, in Web services, which need to be answered as follows:

- How do you manage and how do you organize a huge number of Web services in UDDI? (Candia, 2004; Wang and Vassileva, 2007; Malik and Bouguettaya, 2009)
- How do you manage the increased competitiveness of providers?
- How do you model the users' context? (Sheng and Benatallah, 2005; Bao et al., 2010; Herv´as et al., 2010)
- How do you adapt discovered Web services to the users' context? (Rajendran et al., 2010; Huang et al., 2009; Mohd et al., 2011)
- How do you ensure security of composite Web services coming from homogeneous to heterogeneous systems? (Nacer-Talantikite and Aissani, 2010; Tabatabaei et al., 2010; Rouached, 2012; Balasubramanian and Ruba, 2012)
- etc.

4. Challenges

Interoperability between different organizations is a complex problem that should be addressed both from business and from technological points of view. Several middlewares have been proposed to address the above challenge, for example, DCE, JAVA-RMI (JAVASOFT, 1998; Juric et al., 2000), CORBA (CORBA, 2013; Emmerich, 2000), DCOM (Seinturier, 2007) and XML Web services. However, one way to deal with the heterogeneity problem may be the use of Semantic Web Services paradigm. This latter one seems to be a suitable framework and makes heterogeneous systems interoperable, and even more challenging, compared to the above middleware.

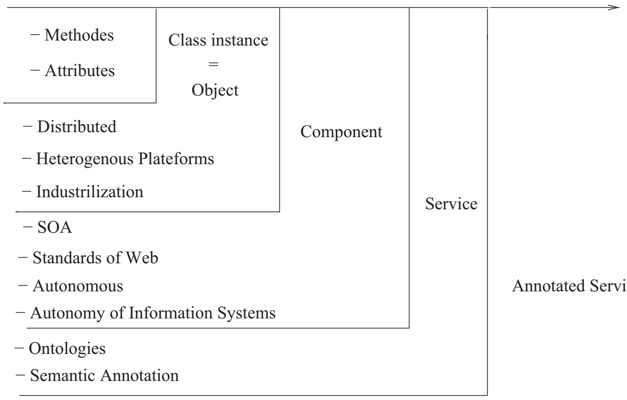

Fig. 5. The concept of distributed objects: in these last decades the concept evolved of an object to a component to a service and today to an annotated service in order to ensure interoperability in large environments for any user.

Moreover, nowadays one of the main problems faced by the designers and the software developers is to make the right choice concerning which middleware to use over the Intranet or over the Internet. The choice includes features such as Performances, Scalability, Maturity, Support for legacy systems and Easiness of development.

In this section, we present the advantages and the disadvantages of each middleware and we present a guideline and a performance comparison according to the following criteria, see Fig. 5 and Tables 9 and 10.

- *Technology*: It specifies the middleware name (DCE, JAVA RMI, etc.).
- *Origin*: It gives the origin of the middleware.
- *Standard*: It defines if the middleware is a standard (Yes or No).
- *Concept*: It points out if the middleware is based on “Distributed Object” concept, or “Component” concept or “Service” concept.
- *OS*: It specifies the used Operating System.
- *Interoperability*: It is the ability of different services to work together. A major reason for the adoption of SOA is the interoperability which allows an easy integration of business functionality over the Internet.
- *Flexibility*: Organizations should be able to adapt to new technologies without any substantial redesign and re-engineering.
- *Integration*: It refers to the possibility to integrate the existing systems with new technologies. The idea is to preserve the main part of the applications, to gradually make evolve and/or to migrate the applications to new technologies.
- *Autonomy*: It permits the separation between the private process of organizations and the public process of coordination.
- *Coupled component* (*Coupled*): In software development, coupling refers to the dependency between software components/ modules (Tightly, Slightly). A tightly coupled system implies that the client and the server logic are closely tied to each other; that means if one interface changes, the other must also be updated.
- *Internet*: It shows that if the middleware is suited for large scale Internet deployments.
- *Complexity*: It relates to the development and to the cost.
- *Communication*: A communication protocol is a set of rules and constraints managing a communication between several entities, such as CORBA IIOP (Internet Inter-ORB Protocol), SOAP, JRMP (JAVA Remote Method Protocol), and ORPC (Object Remote Procedure Call).

Table 9
Comparison between the main known middlewares.

Technology	DCE	JAVA RMI	CORBA
Origin	OSF	SUN	OMG
Standard	Consortium	/	Yes
Concept	/	Distributed object	Distributed object, component
OS	Independence Microsoft's DCOM, ODBC systems	Virtual Machine	Independence
Interoperability	/	JAVA to JAVA to	ORB (Object Request Broken)
Flexibility	/	Yes	Any datatype
Integration	Weak	Strong	/
Autonomy	No	No	With Specification Corba
Coupled	/	Tightly	Tightly
Internet	Failure	Failure	Failure
Complexity	Strong	Simple	Strong
Communication	DCE/RPC	JRMP RMI-IIOP	IIOP
Language	C++	JAVA	Multi-Language
IDL	Socket	JAVA objects	CORBA IDL
Portability	No	/	IDL
Semantic	No	No	No

Table 10
Comparison between the main known middlewares.

Technology	COM/DCOM/COM+	WS	SWS
Origin	Microsoft	W3C	/
Standard	Yes	Yes	No
Concept	Distributed object component	Service	Annotated service
OS	Win32	Independence	Independence
Interoperability	Windows environment COM: Local machine DCOM: Remote server distributed	Any environment	Any environment
Flexibility	No	Yes	Yes
Integration	Average	Strong	Strong
Autonomy	No	Yes	Yes
Coupled	/	Slightly	Slightly
Internet	Failure	Suited	Suited
Complexity	Simple	Simple	Simple
Communication	RPC/ORPC	SOAP	SOAP
Language	Microsoft's Languages	Multi-Languages	Multi-Languages
IDL	Microsoft IDL (MIDL)	WSDL	OWL-S
Portability	No	XML	XML
Semantic	No	No	Yes

- *Language*: It relates to the used programming language(s).
- *IDL*: It provides constructs for specification of types, interfaces, modules and object state. Generally, a mapping from IDL (Interface Definition Language) to standard programming languages is a part of an IDL definition.
- *Portability of code* (*Portability*): It specifies the used inter-operable language.
- *Semantic*: It describes middleware capabilities and their contents in a computer-interpretable language.

Note: OCF: Open System Foundation (OCF, 2013); OMG: Object Management Group (OMG, 2013).

Most of the existing middlewares are tied too closely to the programming models for which they were originally designed. This closeness limits their ability to interoperate with alternate environments which makes their use in heterogeneous systems unsuitable. However, Semantic Web Services differ from several middlewares in the fact that they use the old "remote service" model similar to DCE. Furthermore, Semantic Web services do not have the concept of an object reference but they are defined simply by an end-point that supports various operations by using ontologies. Semantic Web services can be used in any Web environment without having to worry about firewalls. Though semantic Web services could be used regardless of firewalls and they are supported by various operating systems and languages, they have serious overheads that are caused by XML-creation. First, DCE, in spite of its diffusibility and various services that it offers, suffers deeply from the problem of interoperability which is the main need with distributed applications. Second, JAVA is a programming language, but it also has its own component object model with a distinction between objects and interfaces. JAVA RMI provides another solution to the problem that both COM and CORBA address. The major drawback of JAVA RMI is that it is restricted to JAVA and does not apply to components written in other programming languages. JAVA RMI represents the most suitable and simplest choice to implement distributed applications, for JAVA objects systems, that need to communicate over distributed systems. However, this solution is valid for small specific isolated applications because it does not support the rise in charge and can be hardly integrated in large scale information systems. Third, for COM/DCOM, it is a closed choice to represent a validated solution in the future, but it is the best choice for a Microsoft based environment. COM defines its component object model with a simple binary standard and it cannot be used in most Web environments due to protection policies. Finally, CORBA which actually includes CCM (Component Model) allows applications to communicate with each other independent of their location and their design. The interface is completely independent of the object's localization, the programming language, etc. Providing a common representation is also a central key of CORBA which uses IDL. However, the complexity of CORBA and its high cost are considered as drawbacks by specialists. As a result many projects are widely rejected in practice.

4.1. *Performance comparison of JAVA RMI, CORBA and semantic Web services*

We propose an approach, using discrete event, for simulating distributed middleware (JAVA RMI, CORBA, semantic Web services) through the Intranet and the Internet. To our Knowledge, a large number of comparison and performance studies of Web based applications have been proposed in the literature (Eggen and Eggen, 2001; Gray, 2004, 2005; Juric et al., 2004; Wang et al., 2004; Lee et al., 2006; Hwnag and Jun, 2007; Yu et al., 2008; Li and Cheng, 2011; Artemio et al., 2012; Narang et al., 2014; Khawlani, 2012; Mangal and Mahajan, 2012), but there was a need for a generalized approach for performance analysis and estimation of semantic Web services.

4.1.1. *Performance model*

We use networks of queueing stations to model the above middelwares in order to simulate systems processing and messages communication. $M/M/1/N$ queueing system has been used to model queueing stations in the several networks where the arrival process of the users' requests follows a Poisson distribution (M), and we assumed that the transmission time of users' requests at each server is based on an exponential distribution (M).

Each queueing station in the networks comprises a finite capacity buffer (N) to temporarily hold the arriving messages, and a single server to process the messages. The Poisson distribution is expressed by a single parameter $1/\lambda$ to represent the arrival rate, and the exponential distribution is expressed by μ (mu) to represent the message processing time.

As illustrated in Fig. 6, we assume that the traffic arriving to the different stations in the JAVA architecture is modeled as a poisson process with rate λ (lambda), and the treatment time is exponential. The stations are the following: Translator, Security, RMI Registry, Security, Server and Security, for which the treatments time are the following: $\mu1$, $\mu2$, $\mu3$, $\mu4$, $\mu5$, and $\mu6$.

- *Translator*: It translates a general request to a JAVA request.
- *Security*: A server of security.
- *Register RMI* : The register RMI contains several object-references.
- *Security*: A server of security.
- *Server*: It contains the remote object.
- *Security*: A server of security.

As illustrated in Fig. 7, we assume that the traffic arriving at different stations in the CORBA architecture is modeled as a Poisson process with rate λ and the treatment time is exponential.

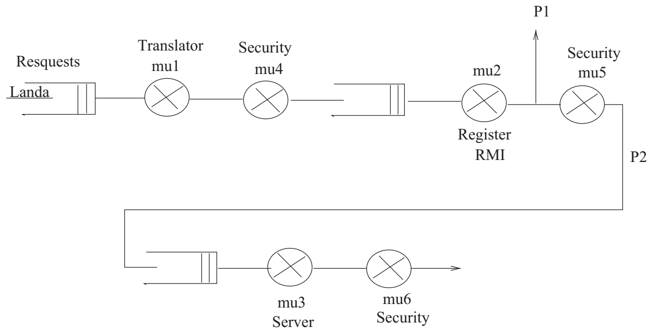

Fig. 6. Modeling JAVA RMI architecture.

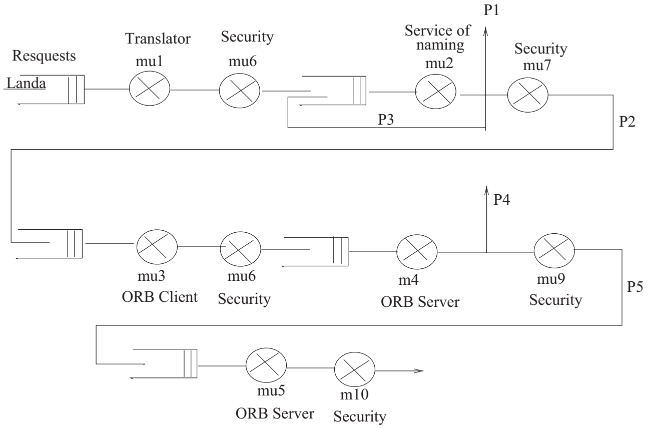

Fig. 7. Modeling CORBA architecture.

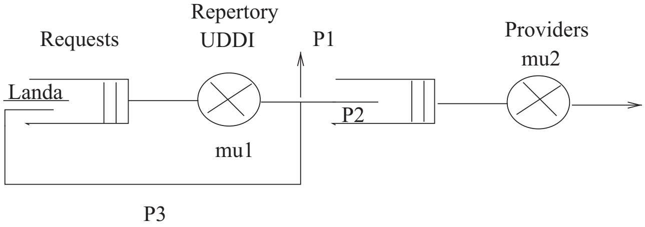

Fig. 8. Modeling Web services architecture.

The stations are the following: Translator, Security, Naming service, Security, ORB Client, Security, ORB Server, Security, ORB Server, and Security, for which the treatments time are the following: $\mu1$, $\mu6$, $\mu2$, $\mu3$, $\mu8$, $\mu4$, $\mu9$, $\mu5$, and $\mu10$. P_1 represents the probability that a request is rejected and P_3 represents the probability that a request is not satisfied and returned to the queuing. P_2 represents the probability that a request is taken into account and it entered the other stations. In this case, the request is satisfied with P_4 or it is redirected with P_5 to another ORB.

- *Translator*: It translates a general request to a CORBA request.
- *Security*: A server of security.
- *Naming service*: It defines the system level object framework. The basic functions of the ORB with services are naming, event, life-cycle, transaction, relation, etc.
- *Security*: A server of security.
- *ORB Client*: Object request broker is the main component of the architecture. It hides all the details of the communication between the two objects (Client, Server). ORB is the object bus.

As illustrated in Fig. 8, we assume that the traffic arriving to the UDDI of semantic Web services is modeled as a Poisson process with rate λ. The station 2 represents the server where the Web service is developed. We suppose that the arrival process to the station 2 is a Poisson process and the treatment time is exponential. P_1 represents the probability that a request is rejected, P_2 represents the probability that a request is satisfied and P_3 represents the probability that a request is not satisfied and returned to the queuing.

- *UDDI* : A registry of semantic Web services.
- *Server*: It is the server of the provider where the semantic Web service is developed.

There are three types of performance analyses: analytic performance analysis, test based performance analysis and simulation based performance analysis. The first performance analysis is fast but inaccurate for the proposed models because they are complex. The second approach for performance analysis is accurate for a dedicated application. However, the third approach that we used for the proposed models, representing the middleware as a discrete event simulation model, simulates their execution on various conditions, and it is fast, accurate and valid on a large scale.

4.1.2. Experimental results

We realized a simulator using discrete event using Matlab and Windows XP as software, on a machine having the Pentium 4 2.17 GHz and 3 GB RAM as hardware. The simulator reproduces the behavior of the proposed models of queueing stations. The goal of the proposed simulator is to analyze the impact of incoming users' requests on the stationary characteristics of the systems. For this purpose, we considered three situations such as Response time, Number of satisfied requests and Discovery time. For that, we considered several variations of $1/\lambda$ as they are illustrated in Tables 12–14 and μ as it is illustrated in Table 11.

After several executions of the simulator for the three cases considered below, for various values of *Tmax*, we noted that the steady state distribution is reached for $Tmax > = 7,000$ μs simulation time units. As a result, the experiments show an average of 40 repetitions of the calls, $Tmax = 10{,}000$ and $N = 100$.

First situation: This experiment computes a response time of each distributed middleware in the Intranet. In this experiment, we considered only the processing time at each server. The time response represents the average time between the arrival of the request in the system and its exit out of the system. Response time is computed by taking into account different traffic loads. We considered different arrival rates of requests as shown in Table 12. It is observed that the JAVA RMI outperforms CORBA and Web Services in response time. JAVA RMI reduces the response time under different traffic loads.

Second situation: This experiment considers the same middleware as experiment 1. Here, we take into account the number of satisfied requests in the Intranet. We considered different arrival rates of users' requests as shown in Table 13. It is observed that JAVA RMI outperforms CORBA and semantic Web Services in the number of satisfied requests.

Third situation: The aim of this experiment is to check how the above middleware affects the discovery delay of the methods (JAVA RMI, CORBA) or the service (Web Service) through the Intranet. The discovery time is calculated under different traffic loads as shown in Table 14. This experiment shows that JAVA RMI performs better than CORBA and semantic Web services in the case of a larger number of methods or services.

Table 11
The Different values of treatment time μ.

Technology	$\mu1$	$\mu2$	$\mu3$	$\mu4$	$\mu5$	$\mu6$	$\mu7$	$\mu8$	$\mu9$	$\mu10$
CORBA	10^3	10^4	10^6	10^5	10^3	10^4	10^5	10^5	10^4	10^3
JAVA RMI	10^3	10^4	10^6	10^4	10^5	10^4	/	/	/	/
Semantic Web services	10^4	10^6	/	/	/	/	/	/	/	/

Table 12
Measured time response for requests with varying middlewares through the intranet (ms): the response time makes it possible to specify the necessary average duration between the arrival of a request and its final departure. It represents the average duration in the system (queue and the various servers). The difference between these responses time is due to the simplicity of JAVA, the concept of package in semantic Web services and the complexity in CORBA. JAVA RMI performs semantic Web services which perform CORBA.

Technology	$\lambda = 1$	$\lambda = 10^3$	$\lambda = 10^6$	$\lambda = 6*10^7$	$\lambda = 36*10^8$
JAVA RMI	0.11	0.11	0.006	0.006	0.005
CORBA	2.1167	0.5951	0.0885	0.0420	0.0001
Web services	2.8094	0.0026	0.0002	0.0002	0.0003

Table 13
The number of satisfied requests with varying middlewares through the intranet: JAVA is more powerful than the two other middlewares. However, in a certain threshold of the λ, the values of the request' number for JAVA and CORBA converge faster. However, for semantic Web services, the number of satisfied requests does not decrease quickly. This is due to the kind of request in the queue: simple in the two middlewares and by group in semantic Web services.

Technology	$\lambda = 1$	$\lambda = 10^3$	$\lambda = 10^6$	$\lambda = 6*10^7$	$\lambda = 36*10^8$
JAVA RMI	7.0174	0.0095	0.0017	0.0016	0.0016
CORBA	2.08094	0.0026	0.0003	0.0003	0.0002
Semantic Web services	1.95	1.6	1.5500	1.2500	1.2750

Table 14
Discovery time for requests with varying middlewares through the intranet (ms): JAVA performs CORBA and semantic Web services. In a certain threshold of the λ, the values of the discovery time for JAVA, CORBA and Semantic Web services converge faster. However, we can classify the three middlewares concerning the discovery time in RMI register with JAVA RMI, in the naming service with CORBA, and in UDDI with Semantic Web services as follows: JAVA-RMI, CORBA and at the end semantic Web services. This is due to the fact that the capacity of UDDI is bigger than the RMI register.

Technology	$\lambda = 1$	$\lambda = 10^3$	$\lambda = 10^6$	$\lambda = 6*10^7$	$\lambda = 36*10^8$
JAVA RMI	0.8002	0.0899	0.0480	0.0436	0.0510
CORBA	1.0720	0.0015	0.0001	0.0001	0.0001
Semantic Web services	1.6333	0.3037	0.0615	0.0252	0.0001

The simulation results, obtained by the proposed comparison using discrete events, can be checked and validated with theoretical results using an analytical method in a future work on a large scale. To our best Knowledge, few works have studied performance evaluation of Web based applications using discrete events, see the research works proposed by Shen (2000), Gong et al. (2010), Ala et al. (2012), and our previous works in the context of semantic Web services (Nacer et al., 2011, 2012, 2013).

Semantic Web services are evaluated in comparison to the existing middlewares through various simulation experiments over the Intranet. These experiments are conducted by considering various traffic loads. All these experiments show that JAVA RMI appears to be more performant. As illustrated in Tables 12 and 14, the time response is longer and the discovery time of semantic Web services is the worst. One of the reasons is the fact that Web services are not optimized using a larger registry UDDI. However, the discovery service in the CORBA naming service or in the JAVA RMI registry is the fastest. Semantic Web Services lose performance in favor of an easy learning curve. Through the Internet, JAVA RMI and CORBA do not fulfill today's requirements of interoperability; they were more suitable for intra-organizations environments, whereas the technical features and choices of Web services make semantic Web services more reusable, and thus more appropriate for inter-organizations and large environments.

It is interesting to note that there are two responses to address interoperability problems in the literature:

- *Middleware community*: The community proposed standards-based approaches, i.e. common protocols and interface description languages. CORBA, DCOM, and XML Web services are effective examples (Nakazawa et al., 2006; Cortesand et al., 2011; Bromberg et al., 2011);
- *Semantic Web community*: The semantic Web community's responses to the interoperability problem are based on the

principles of reasoning about ontologies and understanding how different systems can work together. The work in semantic Web services demonstrates how ontologies can be used to address interoperability problems at the application level. Specifically, ontologies have been used during discovery to express the capabilities of services, as well as the requests for capabilities; in this case, the proof theory recognizes whether a given capability fits a given request (Gordon et al., 2011; Mokhtar et al., 2008, 2010; Nundloll et al., 2011).

According to Gordon et al. (2011), semantic technologies and interoperability middleware have mostly been developed in isolation by distinct communities. The middleware community made assumptions on common application interfaces and focused on middleware behavior and data heterogeneity. The semantic Web community made the opposite assumption that there exists a common middleware, and the solutions focused on differences in application behavior and data. However, these solutions are not suited to today's highly complex distributed systems that exhibit extreme heterogeneity and dynamic behavior.

In our view, the paradigm of semantic Web services seems to be a good choice for loosely coupled architectures. Its success and its popularity are mainly due on one hand to SOA and SOAP protocols, and on the other hand to semantic annotations as follows:

- *SOA* (*Functional Interoperability*): The SOA principles are realized by Web services standards and technologies based on XML (UDDI, WSDL, and SOAP).
- *SOAP protocol* (*Technical Interoperability*): Web services can be accessed through HTTP and HTTPS (Secure Hyper Text Transfer Protocol) protocols and they utilize XML to exchange data. This implies that Web services are independent of any platform, any programming language and any network infrastructure. Web protocols are usually allowed through a firewall and the associated computational cost may be relatively low, due to the possibility of selective encryption and/or signature of SOAP messages. By using SOAP, different applications can read and send messages over HTTP to each other.
- *Semantic annotation* (*operational interoperability*): It facilitates semantic interoperability of data because they refer to ontologies' describing. Semantically described, services will enable better service discovery and allow easier interoperation and composition. Research in semantic Web has shown that annotation with meta-data can help us to solve the problem of inefficient keyword based search in the current Web. The concept of annotation can be extended to Web services to envision semantic Web services.

5. Conclusion

In recent years the number of middlewares, proposed in the literature, which are based on the same principles and have to solve similar issues, has remarkably increased. More and more, the coupling of XML Web services and semantic Web technologies becomes an important research direction. Most complex tasks on XML Web services still fail to satisfy a software agent, but new providers should be able to offer suitable Web services. Thus, they must annotate and describe semantically their services, and the user must deliver a significant request without ambiguity. Semantic Web services paradigm represents a vision of the future where we will be surrounded by software agents. The obtained results, from a comparison study, provide useful guidelines for the design of distributed systems. Through the Internet, the semantic Web services solution appears to be the best solution over most existing middlewares. However, the new paradigm semantic Web services also have drawbacks through the Intranet. All the simulation experiments through the Intranet show that JAVA RMI solution appears to be better than CORBA and semantic Web services. In this paper, we have outlined research results and practical developments, and we have discussed open research issues of semantic Web services paradigm. This latter one includes relevant topics, such as Competencies's Management, Search Engines, Knowledge Discovery tools, Quality of Services, Companies's Collaboration and Negotiation. Semantic Web services can help users to locate relevant documents and assemble relevant Knowledge for effective-making, and improve continuously their capabilities . However, several research works remain to be conducted in order to cope with the existing obstacles, like heterogeneity, languages complexity, semantic annotations, and adaptation to client's context. Furthermore, users need new tools to help themselves to discover and to compose services into processes for an easier and better quality workflow execution. Semantic Web services paradigm, viewed as a distributed system, may be a promising solution in large environments. Semantic Web services paradigm presents another alternative of distributed computing infrastructure: an alternative that is being strongly promoted over distributed objects such as JAVA RMI or CORBA. Indeed, the use of HTTP and XML text documents supports increased interoperability, and also presents a significant increase in run-time cost for Web Services solutions compared to JAVA RMI or CORBA solutions.

As a conclusion of this work, we can address some suggestions that can be considered as a solution to realize the interoperability over the Web.

1. *First*: Representing any Knowledge by semantic Web services in order to achieve Functional Interoperability. SOA is an interoperability mechanism. SOA has characteristics that make effective interoperability easier, such as loose coupling, published interfaces, and a standard communication model (http://www.w3.org; Oaks and HOfstede, 2007; Dong et al., 2013; Gayathridevi and Manikandan, 2013).
2. *Second*: Using standards of XML Web services technology for communication and publication in order to achieve Technical Interoperability. The most common form of SOA is XML Web services. The goal of XML Web services architecture is to allow heterogeneous software applications to smoothly work together by using the standards (UDDI, WSDL, SOAP) (http://www.w3.org/TR/wsdl; http://www.WebServices.org; http://www.ws-i.org; Kadima and Monfort, 2004; Akkiraju et al., 2005; Burstein et al., 2005; Willy, 2013; Wang et al., 2004).
3. *Third*: Adding semantic annotations for description Knowledge in order to achieve Semantic interoperability. Semantic interoperability depends on how the interfaces to a service are described and how the meaning of the information is shared with relevant clients of the service. Ontologies can facilitate semantic interoperability by aligning different terms that might be used in different applications (http://www.w3.org/Submission/OWL-S; http://www.w3.org/TR/2007/REC-sawsdl-20070828; http://www.daml.org/ontologies; http://www.wsmo.org; Fensel et al., 2002; Akkiraju et al., 2005; Kiryakov et al., 2005; Burstein et al., 2005; Nacer-Talantikite et al., 2009; Mohebbi et al., 2012; Dong et al., 2013).

XML Web services need to be improved in order to achieve the discovery and composition of Web services required by interoperability through the Internet (Benna et al., 2008; Nacer-Talantikite et al., 2009, 2011).

The goal of using semantic Web services is to describe Knowledge in a computer interpretable language including semantic Web services interrogation, discovery, selection and composition by any user. Semantic Web services can dynamically integrate different platforms and solve the communication problem: It can let different platforms co-operate smoothly. To assess the results, there is a direction in which we will focus our future work: we will try to develop a real distributed application using the above middelwares. Distributed applications may gain high levels of interoperability, flexibility and reuse by relying on semantic Web services paradigm.

References

Akkiraju R, Farell J, Miller J-A, Nagarajan M, Sheth A, Verma k. Web service semantics-WSDL-S. UGA-IBM Technical Note, 2005 File URL ⟨http://www.w3.org/2005/04/FSWS/Submissions/17/WSDL-S.htm⟩.

Ala F-M, Mohan S, Fowler JW, Gopalakrishnan M. A discrete event simulation tool for performance management of web-based application systems. J Simul 2012;6:21–32.

Alrifai M, Risse T, Dolog P, Nejdl W. A scalable approach for QoS-based web service selection. In: Proceedings of the service-oriented computing; 2009. p. 190–9.

AMICLARE, Tool for textual extraction of information. File URL ⟨http://nlp.shef.ac.uk/amiclare⟩, 2013.

Annotizer O. File URL ⟨http://annotation.semanticWeb.org/ontomat/index.html⟩, 2013.

Arpinar IB, Zhang R, Aleman-Meza B, Maduko A. Ontology-driven web services composition platform. Inf Syst E Bus Manag J 2005;3(2):175–99.

Artemio S, Leonardo B, Hugo J, Carlos M-J. Evaluation of CORBA and web services in distributed applications. In: Proceedings 22nd international conference on electrical communications and computers, Mexico; 2012. p. 97–100.

Aversano L, Canfora G, Ciampi A. An algorithm for web service discovery through their composition. In: Proceedings of the IEEE international conference on web services (ICWS), CA, USA; 2004. p. 332–41.

Baget J-F, Canaud E, Euzenat J, Saïd-Hacid M. Les langages du web sémantique (in french). Technical report, INRIA Rhône-Alpes and LIRIS FRE 2672 CNRS University of Claude Bernard Lyon 1; 2003.

Balasubramanian N, Ruba A. Security: a major threat for web services. In: Proceedings of the IEEE international conference on advanced communication control and computing technologies (ICACCCT), Tamilnadu, India; 2012. p. 104–9.

Bao J, Tao J, Deborah L, Context representation for the semantic web. In: Proceedings of the web science conference, North Carolina, USA, 2010. p. 206–12.

Bartalos P, Bieliková M. Semantic web service composition framework based on parallel processing. In: Proceedings on E-commerce technology, IEEE CS; 2009. p. 495–8.

Bartalos P, Bieliková M. QoS aware semantic web service composition approach considering pre/postconditions. In: Proceedings on web services, IEEE CS; 2010. p. 345–452.

Bartalos P, Bielikova M. Automatic dynamic web services composition: a survey and problem formalization. Comput Inf J 2011;30:793–827.

Beau S, Taouil F-T, Hassanaly P. Collaborate to co-elaborate knowledge between necessity and opportunity. In: Proceedings of the third international conference of information systems and economic intelligence (SIIE), Sousse, Tunisia; 2010. p. 317–30.

Belhajjame K, Embury S-M, Paton N-W, Stevens R, Goble C-A. Automatic annotation of web services based on workflow definitions. ACM Trans Web 2008, http://dx.doi.org/10.1145/1346237.1346239.

Benatallah B, Sheng Q, Dumas M. The self-serv environment for web services composition. IEEE Internet Comput 2003;7(1):40–8.

Benna A, Boudjlida N, Nacer-talantikite H. SAWSDL, mediation and XQUERY for web services. In: Proceedings of the eighth international conference on new technologies in distributed systems (NOTER), vol. 1. Lyon, France: ACM; 2008. p. 23–37.

Berardi D, Automatic service composition, models, techniques and tools [Ph.D. thesis]. Department of Data Processing, University la Sapienza, Rome; 2005.

Bouchaib F, Boudjlida N, Nacer-talantikite H. Generic model of knowledge and competence of domains. In: Proceedings of the third international conference on web and information technologies, Marrakech, Marocco; 2010. p. 209–20.

Boudries A, Boudjlida N, Nacer-talantikite H. Annotation semantiques et web services (in french). In: Proceedings of the first international conference of information systems and economic intelligence (SIIE), Hammamet, Tunisia; 2008. p. 589–601.

Bromberg Y, Grace P, Reveillere L. Starlink: runtime intereoperability between heterogeneous middleware protocols. In: Proceedings of the 31st IEEE international conference on distributed computing systems, Minneapolis, USA; 2011.

Bucchiarone A, Gnesi S. A survey on services composition languages and models. In: Proceedings of international workshop on web services modeling and testing; 2006. p. 51–63.

Bultan T, Fu X, Hull R, Su J. A new approach to design and analysis of e-service composition. In: Proceedings of the 12nd international World Wide Web conference. NY, USA: ACM; 2003. p. 403–10.

Burstein M, Bussler C, Zaremba M, Finn T, Huhns M. A semantic web services architecture. IEEE Internet Comput 2005;9(5):72–81.

Candia T. Benefits of federated identity to government, White paper, Liberty Alliance Project; 2004.

Canfora G, Di Penta M, Esposito R, Villani M. An approach for QoS-aware service composition based on genetic algorithms. In: Proceedings of the international conference on genetic and evolutionary computation (GECCO), Washington, USA; 2005. p. 1069–75.

Cardoso J, Sheth A. Introduction to semantic web services and web process composition in semantic web process: powering next generation of processes with semantic and web services, Lecture notes in computer science; 2005.

Carman M-J, Knoblock C-A. Learning semantic definitions of online information sources. J Artif Intell Res 2007;30:1–50.

Casati F, Ilnicki S, Jin L-J, Shan M-C. An open flexible and configurable system for service composition, HPL 2000-41; 2000.

Casati F, Sayal M, Shan M-C. Developing e-composing e-services. In: Proceedings of 13th international conference on advanced information systems engineering (CAiSE), Interlaken, Switzerland. Lecture notes in computer science, vol. 2068; 2001. p. 171–86.

Chakraborty D, Joshi A. Dynamic service composition: state of the art and research directions, Technical report, CS, Department of Computer Science and Electrical Engineering, University of Maryland; December 2001.

Chappell D, Jewell T. JAVA Web services. o'reilly Edition, Sebastopol, California, USA; 2002.

Charlet J, Laublet P, Reynaud C. Action web sémantique spécifique (in french). Technical report, V3, 32 CNRS/STIC; 2003.

Chein M, Mugnier M-L. Conceptual graphs: fundamental notions. Intell Artif 1992;6:365–406.

Cherifi C, Labatut V, Santucci J. On flexible web services composition networks. In: Proceedings of the international conference on digital information and communication technology and its applications (DICTAP), Dijon, France; 2011. p. 45–59.

Cherifi C, Rivierre Y, Santucci J. A community based algorithm for large scale web service composition. J Converg Inf Technol 2013;8(4):148–57.

COHSE, Conceptual open hypermedia services environment. File URL ⟨http://www.aktors.org/technologies/cohse⟩, 2013.

Condack J, Schwabe D. Swell annotating and searching semantic web services. In: Proceedings of the third Latin American Web Congress LA-WEB, Buenos Aires, Argentina; 2005. p. 102–5.

CORBA, By example. File URL ⟨http://corba.developpez.com/presentation/exemple/LI⟩.

Cortesand C, Grace P, Blair G. Sedim: A middleware framework for interoperable service discovery in heterogeneous networks. ACM Trans Auton Adapt Syst 2011;6(1):1–8.

Cruz I, Xiao H. Using a layered approach for interoperability on the semantic web. In: Proceedings of the 4th international conference on web information systems engineering, 2003. p. 221–32.

Davenport T, Prusak L. Working knowledge: how organizations manage what they know; 1998.

Denoue L. Yet, another Web annotation system. File URL ⟨http://www.fxpal.com/people.denoue/yawas⟩, 2013.

Dodani M-H. From objects to services: a journey in search of component reuse nirvana. J Object Technol 2004;3(8):49–54.

Domingue J, Cabral L, Hakimpour F, Sell D, Motta E. IRS-III: a platform and infrastructure for creating WSMO- based semantic web services. In: Proceedings of the workshop on WSMO implementations (WIW), vol. 113, Frankfurt, Germany; 2004. p. 29–30.

Dong H, Hussain F-K, Chang E. Semantic web service matchmakers: state of the art and challenges. Concurr Comput Practic Exp 2013;25(7):961–88.

Durcik Z. Automated web service composition with knowledge approach. Inf Sci Technol Bull ACM Slovak 2010;2:35–42.

Dustdar S, Schreiner W. A survey on web services composition. Int J Web Grid Serv 2005;1:1–30.

Eggen R, Eggen M. Efficiency of distributed parallel processing using java rmi, sockets and corba 2001. p. 888–93.

Elmaghraoui H, Zaoui I, Chiadmi D, Benhlima L. Graph based e-government web service composition. J Comput Sci Issues 2011;8(1):103–10.

Emmerich W. Engineering distributed objects. John Wiley and Sons Edition, Chichester, UK; 2000.

Esmaeilsabzali S. An interface approach to discovery and composition of web services [Master's thesis]. University of Waterloo, Waterloo, Canada; July 2004.

Fensel D, Bussler C, Maedche A. Semantic web enabled web services. ACM SIGMOD Rec 2002;31(4):1–2.

Gardarin G. XML des bases de données aux services Web (in French). dunod Edition, Paris, France; 2002.

Gasmi B, Boudjlida N, Nacer-Talantikite H. Conceptual graphs for competence. In: Proceedings of the third international conference of information systems and economic intelligence (SIIE), Sousse, Tunisia; 2010. p. 505–14.

GATE, General architecture for text engineering. File URL ⟨http://gate.ac.uc⟩, 2013.

Gayathridevi M, Manikandan R. Semantic web services—a survey. Int J Sci Eng Res 2013;4(6).

Geetha T, Sathyavathy P. Semantic and QoS based web service selection using a multi agent system. In: Proceedings of the second Indian international conference on artificial intelligence (IICAI), Pune, India; 2005. p. 2521–33.

Gehlot V, Edupuganti K. Use of colored petri nets to model, analyze, and evaluate service composition and orchestration. In: Proceedings of the 42nd Hawaii international conference on system sciences (HICSS), Hawaii; 2009. p. 1–8.

Georgakopoulos D, Schuster H, Cichocki A, Baker D. Process-based e-service composition for modelling and automating zero latency supply chains. Inf. Syst Front 2002;4.

Gerber MKA, Schmidt M. Semantic web service composition planning with OWLs-XPlan. In: Proceedings of the AAAI fall symposium on semantic web and agents, Arlington, VA, USA; 2005. p. 55–62.

Gong L, Feng L-G, Zhong L, Ru-Kui A. An opened model with web service in discrete event simulation. In: Proceedings of the second international conference on future computer and communication (ICFCC); 2010. p. 383–6.

Gordon B-S, Bennaceur A, Georgantas N, Grace P, Issarny V, Numdloll V et al. The role of ontologies in emergent middleware: supporting interoperability in complex distributed systems. In: Proceedings of the 12th international middleware conference, Lisbon, Portugal; 2011. p. 410–30.

Gray NA, Comparison of web services, JAVA RMI, and CORBA service implementation. In: Proceedings of the 5th Australasian workshop on software and system accitecture, 2004.

Gray N-A. Performance of JAVA middleware JAVA RMI, JAXRPC and Corba. In: Proceedings of the sixth Australasian workshop on software and system architectures, Australia; 2005. p. 31–9.

Group OM. The common object request broker: architecture and specification. Chichester: Wiley; 1998.

Guangjun G, Fei Y. A method for semantic web service selection based on ontology. J Comput 2011;6:377–86.

Hamadi R, Benatallah B. A petri net-based model for web service composition. In: Proceedings of the 14th Australasian database conference (ADC), vol. 17, Australia, 2003. p. 191–200.

Han W, Shi X, Chen R. Process-context aware matchmaking for web service composition. J Netw Comput Appl 2008;31:559–76.

Harold E-R, Means W-S. XML in a nutshell. o'reilly Edition, Sebastopol, California, USA; 2004.

Hashemian S-V, Mavaddat F. A graph-based approach to web services composition. In: Proceedings of IEEE-IPSJ international symposium on applications and the internet (SAINT), Trento, Italy; 2005. p. 183–9.

Heflin J, Hendler J. Semantic interoperability on the web. In: Proceedings of extreme markup languages; 2000. p. 111–20.

Henni F, Atmani B. Dynamic web service composition: use of case based reasoning and ai planning. In: Proceedings of the fourth international conference on web and information technologies (ICWIT), Sidi Bel Abbes, Algeria; 2012. p. 22–29.

Herv´as R, Bravo J, Fontecha J. A context model based on ontological languages: a proposal for information visualization. J Univers Comput Sci 2010;16:1539–55.

Hess A, Johnston E, Kushmerick N. Assam: a tool for semi-automatically annotating semantic web services. In: Proceedings of international semantic web conference, Hiroshima, Japan; 2004. p. 320–35.

Huang F, Lan C-W, Stephen Y. An optimal QoS-based web service selection scheme. Inf Sci 2009;179(19):3309–22.

Hull R, Su J. Tools for composite web services. ACM SIGMOD Rec 2005;34(2):86–95.

Hwnag Y, Jun S. The performance evaluations and enhancements of GIS web services. In: Proceedings of the international conference on multimedia and ubiqutious engineering; 2007.

Jaeger M-C, Muhl G. Soft real-time aspects for service-oriented architectures. In: Proceeding of the IEEE joint conference on E-commerce technology and enterprise computing, E-commerce and E-services (CEC/EEE), San Francisco, USA; 2006. p. 5.

Jasper R, Uschold M. A framework for understanding and classifying ontology applications. In: Proceedings of the IJCAI99 workshop on ontologies and problem-solving methods; 1999.

JAVASOFT, JAVA remote. method invocation specification, revision. 1st ed.; 1998.

Jiang W, Zhang C, Huang Z, Chen M, Hu S, Liu Z. A tool for QoS-aware automatic service composition. In: Proceedings on web services, IEEE CS; 2010. p. 42–9.

Juric M-B, Rozman I, Hericko M. Performance comparison of Corba and RMI. Inf Softw Technol 2000;42:915–33.

Juric M-B, Kezmah B, Hericko M. Java RMI, RMI tunneling and web services comparison and performance analysis. J ACM SIGPLAN Not 2004;39(5):58–65.

Kadima H, Monfort V. Les services Web, techniques, démarches et outils XML, WSDL, SOAP, UDDI, ROSETTANet, UML (in French). dunod Edition, Paris, France; 2004.

Kagal L, Paolucci M, Srinivasan N, Denker G, Finin T, Sycara K. Authorization and privacy for semantic web services. IEEE Intell Syst 2004;19(4):50–6.

Kahan J, Koivunen M-R, Hommeaux P, Swick R.R. Annotea: an open RDF infrastructure for shared web annotations. In: Proceedings of the 10th international World Wide Web conference, Hong Kong, China; 2001. p. 623–32.

Karimi O. Security model for service-oriented architecture. Adv Comput: Int J 2011;2(4), http://dx.doi.org/10.5121/acij.2011.2405.

Khawlani M-MA. Web services: a bridge between Corba and DCOM. J Sci Technol 2012:1–2.

Kiryakov A, Popov B, Ognyanoff D, Manov D, Kirilov A, Goranov M. Semantic annotation, indexing and retrieval. Web Semant J 2005;2(1):1–39.

Lacot X. Introduction à OWL, Un langage XML d'ontologies Web, enjeux, objectifs et mise en oeuvre, File URL ⟨http://www.lacot.org/public/owl⟩, 2013.

Lammermann S. Runtime service composition via logic-based program synthesis [Ph.D. thesis]. Department of Micro Electronics and Information Technology, Royal Institute of Technology; June 2002.

Lee J. Matching algorithms for composing business process solutions with web services. In: Proceedings of the fourth international conference on E-commerce and web technologies (ECWEB 03), Prague, Czechoslovakia; 2003. p. 393–02.

Lee T-B, Hendle J, Lassila O. The semantic web. Sci Am 2001;5(284):34–43.

Lee S, Lee KW, Dong K, Deok J, Verma D, Jong R et al. Deployment-time binding selection to improve the performance of distributed applications. IBM Research report, Computer Science 3861; 2006.

Lee C-L, Liu A, Huang H. Using planning and case-based reasoning for service composition. Adv Comput Intell Intell Inf 2010;14:540–8.

Li C, Cheng B. A web service performance evaluation approach based on user experience. In: Proceedings of international conference on web services; 2011.

Li L, Wei J, Huang T., High performance approach for multi-QoS constrained web services selection. In: Proceedings of the fifth international conference on service-oriented computing; 2007. p. 283–94.

Lin Y, Ding H. Ontology based semantic annotation for semantic interoperability of process models. In: Proceedings of international conference on computational intelligence for modelling, control and automation; 2005. p. 162–7.

Liu Y, Ngu H, Liangzhao Z. QoS computation and policing in dynamic web service selection. In: WWW (Alternate Track Papers & Posters); 2004. p. 66–73.

Liu W, Du Y, Guo B, Yan C, Xu Q. A fast algorithm for web service composition based on dynamic description logic. Inf Technol J 2010;9(6):1150–7.

LOOM, A concept language. File URL ⟨http://www.isi.edu/isd/LOOM⟩ 2013.

Malik Z, Bouguettaya A. Rateweb: reputation assessment for trust establishment among web services. VLDB J 2009;18(4):885–911.

Mangal D, Mahajan R-P. A novel approach for performance estimation of soap-based web services. Int J Emerg Technol Adv Eng 2012;2(2).

Martin D, Ankolekar A, Burstein M. OWL-S release. File URL ⟨http://www.daml.org/services/owl-s/⟩, 2004.

McIlraith S-A, Son T-C, Zeng H. Semantic web services. IEEE Intell Syst 2001; 16(2):46–53.

Medjahed B. Semantic web enabled composition of web services [Ph.D. thesis]. Computer Science and Applications Faculty of Virginia Polytechnic Institute, USA; 2004.

Medjahed B, Bouguettaya A, Elmagarmid AK. Semantic web enabled composition of web services. VLDB J 2003;12(4):333–51.

Michael C, Gero M. QoS-based selection of services: the implementation of a genetic algorithm. In: Proceeding of the international conference on service oriented architectures and service oriented computing (SOA/SOC), Bern, Switzerland; 2007. p. 1–12.

Mohd F, Fudzee M, Abawajy H. QoS-based adaptation service selection broker. Future Gener Comput Syst 2011;27:256–64.

Mohebbi K, Ibrahim S, Idris N-B. Comtemporary semantic web service frameworks: an overview and comparisons. Int J Web Serv Comput 2012;3(3):961–88.

Mokhtar SB, Preuveneers D, Georgantas N, Issarny V, Berbers Y. Easy: efficient semantic service discovery in pervasive computing environments with QoS and context support. J Syst Softw 2008;8(5):785–808.

Mokhtar S, Raverdy P-G, Urbieta A, Cardoso SR. Sedim: a interoperable semantic and syntactic service discovery for ambient computing environments. Int J Ambient Comput Intell 2010;2(4):13–32.

Nacer H, Aissani D, Adel K. Performance evaluation in virtual web services. In: Proceedings of the 15th Austrian–French–German conference on optimization (AFG), Toulouse, France, Sudwestdeutscher Verlag, Saarbrucken, Germany; 2011.

Nacer H, Aissani D, Adel K. Performance analysis in semantic web services composition. In: Proceedings of the third international conference on industrial engineering and operations management (IEOM), Istanbul, Turkey; 2012.

Nacer H, Adel K, Aissani D. The simulation design and analysis of distributed systems: Java-rmi, corba and web services. In: Proceedings of the EURO/INFORMS 26th conference, Italy, Roma; 2013.

Nacer-Talantikite H, Aissani D. A security model for virtual web services. In: Proceedings of strategic advantage of computing information systems in enterprise management, Athens, Greece; 2010. p. 273–86.

Nacer-talantikite H, Aissani D, Boudjlida N. Semantic annotations for web services discovery and composition. Comput Stand Interfaces J 2009;31(6):1108–17.

Nacer-Talantikite H, Aissani D, Boudjlida N. Les web services complexes (in French). Européenne Edition, Sudwestdeutscher Verlag, Saarbrucken, Germany; 2011.

Nakazawa J, Edwards W-K, Tokuda H, Ramachandran U. A bridging framework for universal interoperability in pervasive systems. In: Proceedings of 26th IEEE international conference on distributed computing systems, Lisbon, Portuga; 2006.

Narang V, Pulusani P, Thakkar H. Comparison of communication protocols in distributed environment. http://sites.google.com/site/distributedwarriors. Project webpage for Distributed Systems / RIT 4005-730, 2014.

Nester C, Philippsen M, Haumacher B. A more efficient RMI for JAVA. In: Proceedings of the ACM conference on JAVA Grande, New York, NY, USA; 1999. p. 152–9.

Niu W, Li G, Zhao Z, Tang H, Shi Z. Multi-granularity context model for dynamic web service composition. J Netw Comput Appl 2011;34:312–26.

Nonaka I, Takeuchi H. The knowledge creating company. Oxford University Press Edition, New York: Oxford University Press; 1995.

Nundloll V, Grace P, Blair G. The role of ontology in enabling dynamic interoperability. In: Proceedings of the 11th IFIP international conference on distributed applications and interoperable systems, Reykjavik, Iceland; 2011.

Oaks P, HOfstede A. Guided interaction: a mechanism to enable ad hoc service interaction. Inf Syst Front 2007;9(1):29–51.

OCF, Open system foundation. File URL ⟨http://www.osf.org⟩, 2013.

OMG, Object management group. File URL ⟨http://www.omg.org⟩, 2013.

Orfali R, Harkey D. Client/Server programming with JAVA and CORBA. Chichester: Wiley, New York: Oxford University Press; 1997.

Papazoglou M-P, Georgakopoulos D. Service-Oriented Computing. In: Communications of the ACM 2003;46(10):25–8.

Papazoglou M, Aiello M, Pistore M, J. Yang. XSRL: an xml web-service request language. Technical report, DIT-02-0079. University of Trento; 2002.

Patil A, Oundhakar S, Sheth A, Verma K. Meteor-s web service annotation framework. In: Proceedings of the 13th international World Wide Web conference, NY, USA; 2004. p. 553–62.

Peer J. A PDDL based tool for automatic web services composition. In: Proceedings of the second workshop on principles and practice of semantic web reasoning (PPSWR) at the 20th international conference on logic programming, Berlin, Germany; 2004. p. 149–63.

Pérez A-G, Fernandez-Lopez M, Corcho O (eds.), Ontological engineering: with examples from the areas of knowledge management, e-Commerce and the semantic web. London: Springer-Verlag; 2004.

Popov B, Kiryakov A, Kirilov A, Monov D, Ognyanoff D, Goranov M. Kim, semantic annotation platform. In: Proceedings of the second international semantic web conference (ISWC), vol. 2870, FL, USA; 2003. p. 484–99.

Portchelvi V, Venkatesan VP, Undaram GS. Achieving web services composition—a survey. Softw Eng 2012;2(5):195–202.

Prashant D, Goodwin R, Akkiraju R, Verma K. Dynamic workflow composition using Markov decision process. Web Serv Res J 2005;2(1):1–17.

Rajendran T, Balasubramanie P, Cherian R. An efficient WS-QoS broker based architecture for web services selection. Int J Comput Appl 2010;1:75–80.

Rao J, Su X. A survey of automated web service composition methods. In: Proceedings of the first international workshop on semantic web services and web process composition (SWSWPC), San Diego, USA. Lecture notes in computer science, vol. 3387; 2004. p. 43–54.

Ren K, Chen J, Chen T, Song J, Xiao N. Grid-based semantic web service discovery model with QoS constraints. In: Proceedings of the third international conference on semantics, knowledge and grid; 2007. p. 479–82.

Rouached M. Security analysis for web services compositions. Int J Sci Eng Res 2012;3(5).

Samuel SJ, Sasipraba T. An approach for graph based planning and quality driven composition of web services. Indian J Comput Sci Eng 2011;2(5):672–9.

Seinturier L. Distributed component object model DCOM/COM+. Technical report. University Pierre and Marie Curie; 2007. File URL ⟨http://arcad.essi.fr/arcad/livrables/proceedings-JC2002.pdf⟩.

Shen C-C. Discrete event simulation on the internet and the web. Future Gener Comput Syst 2000;17(2):187–96.

Sheng QZ, Benatallah B. A UML based modeling language for model-driven development of context-aware web services. In: Proceedings of the international conference on mobile business, Washington, USA; 2005. p. 206–12.

Shin D, Lee K. An automated composition of information web services based on functional semantics. In: Proceedings of the IEEE congress on services; 2007. p. 300–7.

SHOE, Knowledge annotator. File URL ⟨http://www.cs.umd.edu/projects/plus/SHOE/KnowledgeAnnotator.html⟩, 2013.

Sinderen V. Editorial: challenges and solutions in enterprise computing. Enterp Inf Syst 2008;2(4):341–6.

Sirin E, Hendler J, Parsia B. Semi-automatic composition of web services using semantic descriptions. In: Proceedings of workshop of web services: modelling, architecture and infrastructure (ICEIS), Angers; 2003. p. 17–24.

Sirin E, Parsia B, Wu D, Hendler J, Nau D. Htn, planning for web service composition using shop2. Web Semant J 2004;1(4):377–96.

SMORE, Tool for textual extraction of information. File URL ⟨http://annotation.semanticWeb.org/tools/⟩, 2013.

Soley R-M, Stone C. Object management group. Object management architecture guide. Chichester: Wiley; 1995.

Story H, Harbulot B, Jacobi I, Jones M. FOAF+SSL: restful authentication for distributed social networks. In: Proceedings of the first international workshop on trust and privacy on the social and semantic web (SPOT); 2009.

Syncar K-P, Widoff S, Klusch M, LU J. Larks: dynamic matchmaking among heterogeneous software agents in cyberspace. Auton Agents Multi-Agent Syst 2002;5(2):173–203.

Tabatabaei S, Dastjerdi A, Kadir W, Ibrahim S, Sarafian E. Security conscious ai-planning-based composition of semantic web services. Int J Web Inf Syst 2010;6(3).

Taher L, Khatib HE, Basha R. A framework and QoS matchmaking algorithm for dynamic web services selection. In: Proceedings of the second international conference on innovations in information technology (IIT), UAE, Dubai; 2005.

Tao Y, Kwei-Jay L. Service selection algorithms for web services with end-to-end QoS constraints. In: Proceedings of the IEEE international conference on E-commerce technology (CEC), Washington, USA; 2004. p. 129–36.

Tao Y, Kwei-Jay L. Service selection algorithms for composing complex services with multiple qos constraints. In: Proceedings of the third international conference on service oriented computing (ICSOC), Berlin, Heidelberg; 2005. p. 130–43.

Tao Y, Yue Z, Kwei-Jay L. Efficient algorithms for web services selection with end-to-end QoS constraints. ACM Trans Web 1. http://dx.doi.org/10.1145/1232722.1232728, in press.

Tari A, Elgedawy I, Dahmani A. A dual-layered model for web services representation and composition. J Intell Inf Syst 2009;32:237–65.

Tsai W. Ontology-based dynamic process collaboration in service-oriented architecture. In: Proceedings of IEEE international conference on service-oriented computing and applications; 2007. p. 39–46.

Vadivelou G, Ilavarasan E, Prasanna S. Algorithm for web service composition using multi-agents. Int J Comput Appl 2011;13(8):40–5.

Vera M-V, Motta E, Domingue J, Lanzoni M, Stutt A, Ciravegna F. MnM: Ontology driven, semi-automatic and automatic support for semantic markup. In: Proceedings of the 13th international conference on knowledge engineering and management (EKAW), Lecture notes in computer science, vol. 2473; 2002. p. 379–91.

W3C, The World Wide Web consortium. File URL ⟨http://www.WebServices.org/⟩.

W3C, The World Wide Web consortium. File URL ⟨http://www.w3.org/⟩.

W3C, Web services description language (WSDL). File URL ⟨http://www.w3.org/TR/wsdl⟩.

W3C, Semantic annotations for WSDL and XML schema. File URL ⟨http://www.w3.org/TR/2007/REC-sawsdl-20070828⟩.

Wache H. A survey of existing approaches. In: Proceedings of the workshop on ontologies and information sharing, Seattle, USA; 2001. p. 108–17.

Waldinger R. Web agents cooperating deductively. In: Proceedings of the first international workshop formal approaches to agent-based systems (FAABS), Greenbelt, USA. Lecture notes in computer science, vol. 1871; 2001. p. 250–62.

Wang Y, Vassileva J. Toward trust and reputation based web service selection: a survey. Int Trans Syst Sci Appl J 2007;3(2):118–32.

Wang H, Huangc J, Qub Y, Xie J. Web services: problems and future directions. J Web Semant 2004;1(3):309–20.

Wang X, Vitvar T, Kerrigan M, Toma I. A QoS-aware selection model for semantic web services. In: Proceeding of the fourth international conference on service oriented computing (ICSOC), Berlin, Heidelberg; 2006. p. 390–401.

Web-Site, Common object request broker architecture. File URL ⟨http://www.omg.org⟩.

Web-Site, Remote method invocation. File URL ⟨http://JAVA.sun.com/j2se/1.4.2/docs/guide/rmi/index.html⟩.

Web-Site, Web services interoperability organization. File URL ⟨http://www.ws-i.org/⟩.

Web-Site, DAML ontology library. File URL ⟨http://www.daml.org/ontologies/⟩.

Web-Site, The OWL services coalition, OWL-S: semantic murkup for web services. File URL ⟨http://www.w3.org/Submission/OWL-S/⟩.

Web-Site, Web service modelling language. File URL ⟨http://www.wsmo.org/wsml/⟩.

Web-Site, Web services modelling ontology. File URL ⟨http://www.wsmo.org/⟩.

Web-Site, Description logic. File URL ⟨http://dl.kr.org/⟩.

Willy C. Technology SOA and web services. The performance paradox. File URL ⟨http://www.ca.com/us/whitepapers/collateral.aspx?cid=147947⟩, 2013.

Yue K, Yue1 M, Liu W, Li X. A graph-based approach for type matching in web service composition. J Comput Inf Syst 2010;6(7):2141–9.

Yu Q, Liu X, Bouguettaya A, Medjahed B. Deploying and managing web services: issues, solutions, and directions. VLDB J 2008;17(3):537–72.

Zeng L, Benatallah B. Qos-aware middleware for web services composition. IEEE Trans Softw Eng 2004;30:311–27.

Zheng L, O'Brien L, Keung J, Xiwei X. Effort-oriented classification matrix of web service composition. In: Proceedings of the fifth international conference on internet and web application and services (ICIW), Barcelona, Spain; 2010. p. 357–62.

Zhou Z, Tianruo L, Bhiri S, Shu L, Xiong N, Hauswirth M. Verifying mediated service interactions considering expected behaviours. J Netw Comput Appl 2011;34(4):1043–53.