Service-Oriented Middleware for the Future Internet: State of the Art and Research Directions

Valérie Issarny, Nikolaos Georgantas, Sara Hachem, Apostolos Zarras, Panos Vassiliadis, Marco Autili, Marco Aurelio Gerosa, Amira Ben Hamida

HAL Id: inria-00588753
https://inria.hal.science/inria-00588753v2

Submitted on 2 May 2011

Service-Oriented Middleware for the Future Internet: State of the Art and Research Directions

Valérie Issarny · Nikolaos Georgantas · Sara Hachem · Apostolos Zarras · Panos Vassiliadis · Marco Autili · Marco Aurélio Gerosa · Amira Ben Hamida

Abstract Service-oriented computing is now acknowledged as a central paradigm for Internet computing, supported by tremendous research and technology development over the last ten years. However, the evolution of the Internet, and in particular, the latest Future Internet vision, challenges the paradigm. Indeed, service-oriented computing has to face the ultra large scale and heterogeneity of the Future Internet, which are orders of magnitude higher than those of today's service-oriented systems. This article aims at contributing to this objective by identifying the key research directions to be followed in light of the latest state of the art. This article more specifically focuses on research challenges for service-oriented middleware design, therefore investigating service description, discovery, access and composition in the Future Internet of services.

Valérie Issarny, Nikolaos Georgantas and Sara Hachem
INRIA, France
E-mail: first.lastname@inria.fr

Apostolos Zarras and Panos Vassiliadis
University of Ioannina, Greece
E-mail: name@cs.uoi.gr

Marco Autili
Università degli Studi di L'Aquila, Italy
E-mail: marco.autili@di.univaq.it

Marco Aurélio Gerosa
University of Sao Paulo (USP), Brazil
E-mail: gerosa@ime.usp.br

Amira Ben Hamida
PetalsLink, France
E-mail: amira.ben-hamida@petalslink.com

1 Introduction

Service-Oriented Computing (SOC) is now largely accepted as a well-founded reference paradigm for Internet computing [102]. Under SOC, networked devices and their hosted applications are abstracted as autonomous loosely coupled services within a network of interacting service *providers*, *consumers* (aka *clients*) and *registries* according to the *service-oriented interaction pattern* (see Figure 1).

Fig. 1 Service-oriented interaction pattern

Still, despite the remarkable progress of the SOC paradigm and supporting technologies in the last ten years, substantial challenges have been set through the evolution of the Internet. Over the years, Internet has become the most important networking infrastructure, enabling all to create, contribute, share, use, and integrate information and knowledge by all. As a result, the Internet is changing at a fast pace and is called to evolve into the Future Internet, i.e., a federation of service- and self-aware networks that provide built-in and integrated capabilities such as: service support, contextualization, mobility, security, reliability, robustness, and

Constituent	Definition	Reference
Internet of Content	*Content is any type and volume of media. Content may be pre-recorded, cached or live, static or dynamic, monolithic or modular. Content may be combined, mixed or aggregated to generate new content and media. It may vary from a few bits (e.g., the temperature that a sensor has measured) to interactive multi-media sessions and immersive complex and multi-dimensional virtual/real worlds' representations.*	[48]
Internet of Services	*An umbrella term to describe several interacting phenomena that will shape the future of how services are provided and operated on the Internet. The Internet of Services also comprises the various sets of Internet Applications including pervasive/immersive/ambient, industrial/manufacturing, vehicular/logistics, financial/ePayment/eBusiness, power network control/eEnergy, eHealth, and eGovernment applications.*	[101]
Internet of Things	*A global network infrastructure, linking physical and virtual objects through the exploitation of data capture and communication capabilities. This infrastructure includes existing and evolving Internet and network developments. It will offer specific object-identification, sensor and connection capability as the basis for the development of independent cooperative services and applications. These will be characterized by a high degree of autonomous data capture, event transfer, network connectivity and interoperability.*	[33]

Table 1 The Future Internet constituents

self-management of communication resources and services [133].

Practically, the Future Internet vision challenges all the SOC architectural layers, from the bottom to the top: service foundations as formed by the service-oriented middleware realizing the runtime infrastructure, service composition, and service management and monitoring [102]. In this context, the goal of this article is to highlight research directions in the area of service-oriented computing in the Future Internet, based on today's state of the art. However, due to the breadth of the area, the article focuses more specifically on the study of the challenges posed to the middleware layer. Briefly stated, *Service-Oriented Middleware* (SOM) supports the service-oriented interaction pattern through the provision of proper functionalities for deploying, publishing/discovering and accessing services at runtime. SOM commonly also provides support to realize more complex composite services by integrating simpler ones, where it should be acknowledged that this contributes to the upper service composition layer.

In accordance with the above, this article starts by setting the overall challenges and requirements posed by the Future Internet, which in particular relate to its expected ultra large scale, heterogeneity, and mobility. The article is then structured in relation to the essential functionalities of Service-Oriented Middleware, i.e., service description, access, discovery and composition, surveying related state of the art and Future Internet challenges for each one of them. Precisely, in Section 2, we provide our definition of the Future Internet vision and major challenges that come along with it. Then, in Section 3, we survey the description of services that needs to be provided for enabling effective service use in the greatly complex Future Internet environment. In

Section 4, we concentrate on service discovery in the Future Internet, with a special focus on the organization, management and distribution of supporting service registries. In Section 5, we study middleware support for service access, where we highlight the key role now played by the Enterprise Service Bus paradigm as well as the evolution needed to meet Future Internet requirements. In Section 6, we focus on decentralized choreography-based composition in the Future Internet, and associated modeling and runtime support. Finally, the conclusions are presented in Section 7.

2 Future Internet Challenges and Requirements

The Future Internet has become the main focus of several research and development initiatives all over the world, including initiatives in the EU[1], USA[2], China[3], Korea[4], and Japan[5]. However, despite the great interest in the Future Internet, no common definition of it has been adopted yet. Still, considering that the Future Internet will result from the evolution of today's Internet, the Future Internet can be defined as the union and cooperation of the *Internet of Content*, *Internet of Services*, and *Internet of Things*, supported by an expanding network infrastructure foundation. Those core domains, elements of which we find already in today's Internet, are not fully established yet and will emerge with the foreseen evolution of services, content, objects and networks, as summarized in Table 1.

---

[1] http://www.future-internet.eu
[2] http://www.nets-find.net
[3] http://www.cstnet.net.cn/english/cngi/cngi.htm
[4] http://fif.kr
[5] http://akari-project.nict.go.jp/eng/overview.htm

Challenges	Today's Internet	Toward the Future Internet
Scalability	1 billion Personal Computers (2008)[(*)], 647 million smartphones (2010) [44]	1.78 billion Personal Computers (2013), 1.82 billion smartphones (2013)[(+)]
	5 exabytes of data (2005) [52]	990 exabytes of data (end of 2012) [143]
	10^4 services (2007) [4]	Billions of services [133]
	10 billion terminals (2010) [2]	100 billion terminals (2015) [2]
	Consumer Internet traffic of 12.684 exabytes/month (2010) [43]	Consumer Internet traffic of 42.070 exabytes/month (2014) [43]
Heterogeneity	Islands of interconnected objects	Internet-scale connection of highly heterogeneous objects (vehicles, sensors, mobiles devices, home appliances, etc.) [15]
	Emergence of heterogeneous services provided on the Cloud such as Software as a Service (e.g., Google apps) or Infrastructure as a Service (e.g., Storage services at Amazon) [149]	Cloud Computing enabling to provide everything as services, spanning different business and technical domains
	Service/content mashups leading to the provision of new, diverse services by *prosumers*	Global-scale services/content mashups creating new services/content with different types and formats
Mobility	Mostly (mobile) IPv4, which suffers from scalability issues etc.; even IPv6 has issues in mobile situations (e.g., due to the use of home agents/addresses) [101]	Global-scale mobile Internet that requires revisiting communication/routing solutions [4]
	Wide-spread usage of smart mobile devices with limited resources (2 billion users)	Global scale usage of smarter mobile devices with ever-growing resource needs
Awareness & Adaptability	*Ad hoc* solutions to network, content & service adaptation	Large scale content sharing, service provisioning, mobile connectivity that require autonomic adaptation and therefore awareness of content, networks and services [101]
Security, Privacy & Trust	Safety and security requirements still an issue for today's Internet	Integrating real world objects, more users, more information, more services in the Internet intensifies the necessity for safety and security solutions

[(*)] http://www.gartner.com/it/page.jsp?id=703807
[(+)] http://www.gartner.com/it/page.jsp?id=1278413

Table 2 The Future Internet challenges

In general, the Future Internet is setting significant challenges over the computing and networking environments, as it magnifies the features of the already challenging Internet of today (see Table 2). Specifically, key challenges posed by the Future Internet relate to and are amplified by the highly correlated nature of the following requirements:

- *Scalable Internet:* The Internets of Content, Services and Things are confronted with scalability issues due to the increasing number, size and quality of their networked entities, which is further exacerbated by the empowerment of users who are now becoming *"prosumers"* [105,101,128]. For instance, simply considering the Internet of Things, the large amount of new information available through things needs to be comprehensively managed and aggregated to provide useful services [101].
- *Interoperable Internet:* The Future Internet will be heterogeneous in many dimensions, related to physical objects, networks, services and data, which presents a significant challenge for sustaining the

Future Internet vision [101]. In particular, appropriate semantic technologies, shared standards and mediation are required to assure interoperability of heterogeneous entities such as things, sensors, and networks [132].

- *Mobile Internet:* Unlike the current Internet, mobility should be natively integrated in the design of the Future Internet. Indeed, an essential challenge for the Future Internet lies in the explicit design of a protocol for a mobile wireless world given that the majority of the connected entities are now mobile.
- *Aware and adaptive Internet:* Awareness and related adaptability are common requirements for sustaining the Future Internet, be it at the service, content or physical object level. Issues to be addressed include: adapting the Web by and for users, adapting the network to shared media and vice versa, providing personalized content and media to users, providing context-aware and personalized dynamic services [101,128,132].
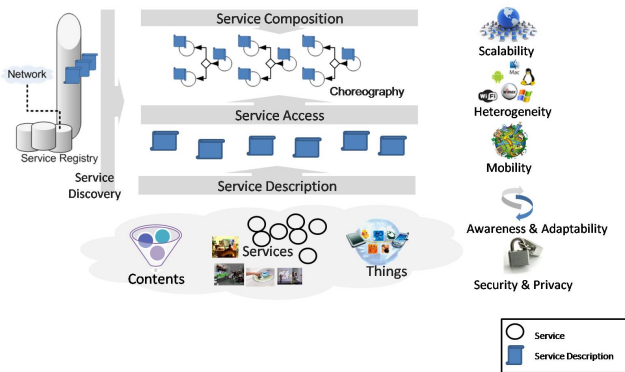- *Safe Internet:* Trust, privacy and security are sensitive cross-domain issues that the current Internet

Fig. 2 Service-oriented computing in the Future Internet

Fig. 3 Service description

is facing and remain critical challenges for the Future Internet. With the global-scale communications and exchange of information, users' mobility and the limited resources their devices may have, as well as the Future Internet's "awareness" of users, their data, and their surroundings, it becomes crucial to find appropriate solutions that will protect users. Indeed, current security mechanisms are unfit in such an open, dynamic and aware setting.

The following sections point out research directions for service-oriented middleware in light of the latest state of the art and the above requirements posed by the Future Internet. The remainder specifically concentrates on the challenges that arise for the base functionalities of service-oriented middleware in the Future Internet (see Figure 2), i.e.: service description, discovery, access, and composition.

3 Service Description

Service description is a fundamental element in SOC, as it determines the information that a service needs to expose to its environment for enabling its unambiguous identification and use. All other information internal to the service is simply out of the scope of SOC.

3.1 State of the Art

Information included in service description varies depending on the complexity and the intended use of the service. Accordingly, a number of service description languages have been proposed to cover different description aspects and are currently in use, some of them having reached the status of standard and other still being the subject of research. Most of the related initiatives focus on Web Services[6] being the dominant
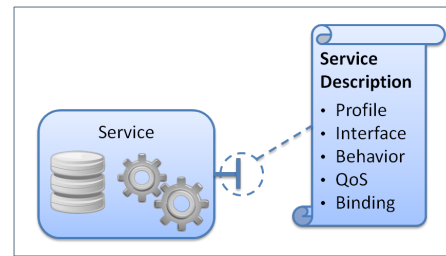
technology for SOC. W3C and OASIS are the two leading standardization bodies in this area. Besides Web Services, the Semantic Web initiative[7] and related technologies have produced a significant change in the way services are perceived and described. In particular, the innovation is to make explicit the business or user-domain semantics of services, so far implied by the syntax of their descriptions. Employing syntactic descriptions either makes service semantics ambiguous or calls for a syntax-level agreement between service developers/providers and service users, which is too restrictive for the inherent loosely coupled character of SOC. Service semantics are made explicit by reference to a structured vocabulary of terms (*ontology*) representing a specific area of knowledge. Ontology languages support formal description and machine reasoning upon ontologies; the Web Ontology Language (OWL)[8] is the standard established by W3C. In the following, we discuss the most common elements of service description (see Figure 3) and survey their coverage by the most widely used service description languages.

Service Profile provides a high-level business description of a service, which may include both human-oriented information (e.g., what the service does and service provider information) and machine-oriented elements. The latter, in particular, may range from a simple service name to a precise semantic characterization of the service comprising its provided high-level functionalities as well as its high-level Inputs, Outputs, Preconditions and Effects; these are collectively denoted as IOPEs. Inputs specify the data required by the service for its execution and Outputs specify the data provided by the service as result of its execution. In addition, Preconditions need to be fulfilled before the service may execute, while Effects specify the impact of the service on the state of the world besides its Outputs.

The notion of Service Profile was created or at least made popular by OWL-S. OWL-S[9] is an OWL ontology

---

for describing Web services. In OWL-S, a service description is composed of three parts: the *service profile* is complemented by the *process model* and the *service grounding* (see related paragraphs below). The service profile provides semantic descriptions of the service's *capabilities* (the OWL-S term for service's high-level functionalities) in terms of IOPEs. OWL-S was a candidate for becoming the W3C standard for semantic service description; however, the winner was SAWSDL, discussed in the paragraph on *Service Interface* below.

The expressive power of Service Profile has been widely acknowledged in semantic service matching approaches [100]: matching between a requested and a provided service profile is typically the first step in service discovery and selection (see Section 4). However, while matching between requested and provided inputs and outputs is commonly applied, there is much less use of preconditions and effects. Besides, there is less agreement within the SOC community regarding how PEs should be specified and used. After a number of submissions to the W3C of candidate rule languages for the Semantic Web (suitable for being used in PEs specification and reasoning), the outcome was the creation of a W3C working group studying a *Rule Interchange Format*[10], a future standard for exchanging rules among rule systems, as no single one-fits-all rule language could be identified.

Service Interface specifies the set of observable lower-level (with respect to the functionalities in the Service Profile) atomic operations that a service can perform in coordination with its environment, along with their input/output parameters. Service Interface is the fundamental and mandatory element of service description, as it technically enables the access to a service as a software component (see Section 5). Web Services Description Language (WSDL), now in its 2.0 version[11], is traditionally the language for describing Web service interfaces, and a distinctive element of the Web Services technology. SAWSDL[12] is the W3C Recommendation for adding semantic annotations to WSDL and XML Schema. Such annotations can be expressed in any ontology language, most often in OWL. Annotations can be added to WSDL interfaces, operations, and the XML Schema types of their input/output parameters. Moreover, SAWSDL supports the introduction of two-way transformation mappings between XML Schema types and corresponding semantic concepts. This enables the interoperability between syntactically mismatching input/output parameters upon service invocation. In the

case of OWL-S, a service interface is specified semantically in the process model and syntactically in the service grounding, with appropriate mapping between the two. WSDL is again used in OWL-S service grounding.

In parallel to the development of WS-* technologies for Web Services, REpresentational State Transfer (REST) [55] was introduced as an architectural style and an alternative way (or a return to fundamentals) for enabling services on the Web (RESTful Web services) by using the standard Web mechanism: any entity on the Web is a resource at some URI and can be accessed with the standard HTTP operations. The advantages of REST are its universality and the uniform service interface. Nevertheless, it addresses only basic distributed interaction/coordination [104], leaving open many issues that have been tackled by SOC, such as dealing with service behavior.

Service Behavior specifies the observable supported execution patterns (often called *conversations*) of the service in coordination with its environment that allow the service to produce meaningful results. Such execution patterns involve the operations of the Service Interface and are typically represented as processes.

The OWL-S process model was introduced for describing service conversations that are associated with the realization of service capabilities. However, with the prevalence of SAWSDL over OWL-S, and since SAWSDL does not cover the description of service conversations, a widely accepted business workflow language, the Web Services Business Process Execution Language (WS-BPEL[13]), is often associated with SAWSDL for describing service behavior. WS-BPEL started out as an industrial *de facto* standard and evolved into an OASIS standard. A number of research efforts have elicited formal semantics for WS-BPEL, e.g., based on process algebras [57]. Formal semantics enables the automated reasoning about BPEL processes for service behavior matching as part of service discovery and selection.

Service QoS concerns non-functional properties of the service, such as reliability, performance, security, privacy, trust, which characterize the quality of the results that the service promises to provide to its environment. Depending on the intended use of the service, non-functional properties are often considered less important than functional ones and QoS is omitted in service descriptions. Moreover, in contrast to functional service features, there is less agreement within the SOC community regarding the ways in which QoS should be identified and specified. Nevertheless, as SOC evolves

---

into a major paradigm covering many application domains and quality-critical applications, QoS should be treated accordingly.

Numerous efforts have produced QoS languages and models for general or domain-specific QoS description. Some of these efforts incorporate the expressiveness and reasoning power of ontologies. The Web Service Quality Model (WSQM[14]) is an ongoing standardization effort by OASIS for the specification of Web Services QoS. WSQM is a conceptual model; it defines a well-founded taxonomy of QoS and provides a wide range of QoS properties. Then, accompanying the model, WS-QDL is a XML-based description language for representing Quality of Service by applying WSQM. An extension of WSQM particularly for dynamic SOC environments and an OWL-based semantic transcription of the extended model have been reported in [85]. Models like WSQM, coupled with a Service Level Agreement language and protocol – the WS-Agreement[15] proposition of the Open Grid Forum is the most widely used – allow the establishment and control of runtime QoS contracts between service providers and consumers [97].

Service Binding specifies the underlying communication middleware on which the service is deployed, and hence gives all the information required for accessing the service at middleware protocol level. This information commonly includes the middleware protocol and message format, as well as the service endpoint, e.g., in the form of a URI. Common bindings are SOAP/HTTP/TCP for WS-∗ Web Services and HTTP/TCP for RESTful Web Services. Binding information is included in WSDL.

3.2 Research Challenges

Being a greatly complex environment, the Future Internet places high requirements on service description with respect to the amount and variety of information that needs to be exposed by a service to its environment. In particular, the Future Internet challenges identified in Section 2, i.e., Heterogeneity, Mobility, Awareness & Adaptability, and Security, Privacy & Trust, require the enhancement of service descriptions with relevant information, in order to allow service clients to take into account the rich service features as well as the underlying SOC middleware to manage the complexity of the association between services and their clients. Nevertheless, due to the Future Internet's ultra large scale, this creates a trade-off between the richness of service

description information and its efficiency. In the following, we discuss challenges for service description raised by the specifics of the Future Internet. More particularly, we analyze the impact of Scalability and Heterogeneity on service description, which demonstrates the above trade-off. As for Mobility, Awareness & Adaptability, and Security, Privacy & Trust, we discuss their impact on service access in Section 5; the associated effect on service description is implicit.

Scalability. The ultra large scale of the Future Internet is a determining factor in the trade-off between richness and efficiency of service descriptions. Millions of Resources, People and Things will need to be described and choreographed within complex service compositions in the Future Internet, which raises demanding requirements for the storage, publication, search, access, and reasoning about service descriptions. Such requirements mainly concern service discovery (see Section 4), which upon a service request, has to deal *a priori* with the entire collection of Future Internet services. Still, the service access and composition that may follow service discovery are also dependent on the offering of comprehensive and, at the same time, efficient service descriptions. The identified challenges call for advances in the expressiveness and processing efficiency of XML-based service description languages, as well as in the efficient encoding and reasoning about semantic annotations that are a part of such languages. Initial approaches in this area concern the encoding of ontologies, aiming to accelerate semantic reasoning considerably when executed in dynamic and resource-constrained environments [19].

Heterogeneity. Heterogeneity is a key characteristic of the Future Internet of services that aims to encompass all kinds of resources and present them as services. Such services may be hosted on platforms ranging from resource-rich fixed machines to wireless resource-constrained devices, and further to any physical object enhanced with some networking capacity thus turned into a Thing. Such extreme heterogeneity should be accounted for in service description: the trade-off between richness and efficiency identified above equally applies here, especially when a resource-constrained service hosting platform also needs to manipulate the service description.

Furthermore, besides the common heterogeneity tackled by the service abstraction itself (service internal implementation and hosting platform features), service heterogeneity will be exacerbated by the diversity in both business semantics and communication middleware brought about by the Future Internet. Regard-

---

ing the former issue, dealing with ontologies in delimited SOC environments is already hard, due to the lack of widely accepted ontologies and the fact that the heterogeneity problem that ontologies have aimed to resolve has now moved one abstraction level up, to the ontology level itself. These already demanding matters become even more challenging in the open, unlimited Future Internet context. Establishing reference or global ontologies [21] and tackling the ontology heterogeneity problem [107] are two key requirements in the area. On the other hand, the diversity in communication middleware is due to the Future Internet vision aiming to incorporate all current networking environments into a single, ubiquitous setting. This calls for support for heterogeneous coordination/interaction models, namely message-driven, event-driven, and data-driven models. Different coordination models apply to different needs; for instance, asynchronous, event-based publish/subscribe is more appropriate for highly dynamic environments with frequent disconnections of involved entities. This fact makes the various service bindings accounted for in current SOC service descriptions too stringent, since they comply with a single (client/service) message-based coordination model. Service description should be able to abstract and comprehensively specify the enriched service bindings of the Future Internet. This further implies extending the notion of service and introducing adequate service coordination modeling.

4 Service Discovery

Publishing and discovering descriptions of available services are two core functions of the service-oriented interaction pattern (see Figure 1). As discussed in this section, the existing solutions to service discovery cover various research issues that emerged in the current Internet setting. However, there is still plenty of room for research towards the new challenges introduced in the context of the Future Internet.

4.1 State of the Art

We organize the discussion of service discovery in two parts (see Figure 4): the first part concerns issues related to the way available service descriptions are organized and managed within dedicated registries, whilst the second part focuses on protocols for service retrieval that mainly differ according to the architecture of the registry, from centralized to distributed.

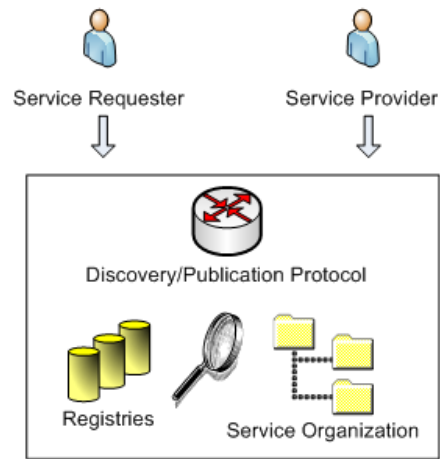For the interested reader who would like to probe further, there are certain excellent surveys worth men-

Fig. 4 Service discovery

tioning. Specifically, in [51], the authors pose criteria for the evaluation of service registries and organize the discussion from two viewpoints: the viewpoint of the system and the viewpoint of humans. A more recent effort [108] emphasizes the degree of distribution of the service registry and the use of semantic information in the service matchmaking process. Further works focus on service discovery protocols for mobile *ad hoc* environments, which are part of the Future Internet [135, 95].

4.1.1 Service Registry

The baseline approach concerning the organization and management of service descriptions within registries, is the data model that has been proposed in the Universal Description Discovery and Integration (UDDI) specification[16]. According to the UDDI standard specification, information is organized as a collection of (a) white pages where business entities advertise the offering of business services, (b) yellow pages where this advertisement is based upon a taxonomy, and (c) green pages, where pointers to Web service descriptions are also provided. For the description of the technical characteristics of Web services, UDDI proposes the use of *tModels* (technical models), along with mappings between tModels and WSDL (see Section 3). The UDDI standard specifies an API that can be used by front-end tools to pose service discovery queries. The API provides mainly keyword (or value) lookup functionalities. Other similar data models include ebXML and WSIL but they have less impact in the state of the art [51].

From the early days of service discovery, a major research issue that emerged was the enhancement of the

---

[16] http://uddi.microsoft.com

simple UDDI data-model with semantically rich metadata. All the efforts in this line of research base their motivation on the fact that both the UDDI data-model and its querying mechanisms support keyword queries over the stored tModels without any semantic information. Generally, this research relies on the proliferation of approaches that annotate information with tags coming from a reference ontology. These approaches come along with tools that enable the exploitation of semantic relationships of the ontology's terms (generalization, synonyms, subsumption, and other relationships of this kind). This idea was adopted in several service discovery approaches that map semantic service descriptions like the ones discussed in Section 3 to the UDDI data-model, in order to support more sophisticated querying (e.g., [99, 123, 83]).

In general, the semantic annotation of the service information managed by service registries requires extra specification effort. This effort may become an impediment towards the scalability of related approaches. Consequently, another research issue that arose was to incorporate in the service registries means for extracting semantically rich information out of available service descriptions in a (semi)automated way. Specifically, the goal is to find groups of semantically similar services and ontologies/vocabularies that may be used for the semantic annotation/specification of these groups. Typically, the semantic information is extracted by employing well-known clustering [94, 54] and classification [80] techniques that group semantically similar service descriptions with respect to the terms used in these descriptions.

Another interesting research issue that relates with the management of service descriptions by service registries is coping with the potential heterogeneity of these descriptions. As already discussed in Section 3, there are various means for the syntactic or the semantic specification of service-related information. Such heterogeneous service-related information may be published and consequently the service registries should be capable of managing it. To this end, general purpose models are used as pivots for mapping heterogeneous service descriptions [20].

4.1.2 Service Discovery Protocol

Regarding service discovery protocols, the baseline approach is *centralized*, relying on a single registry that organizes and manages service descriptions. This provides consistency and fast local retrieval under normal circumstances.

Unfortunately, the centralized approach has various drawbacks that correspond to research issues that emerged in the early days of service discovery. First, the centralized approach does not scale well with respect to the increasing number of clients that pose service discovery queries [51, 108]. Moreover, in the centralized approach, the single registry constitutes a single point of failure. Therefore, the availability of centralized solutions is questionable. The aforementioned issues become even more important, in the cases where service discovery is supposed to be handled in mobile computing environments that consist solely of mobile nodes [135, 95]. To deal with the scalability, the availability and the mobility issues, several approaches proposed *decentralized solutions*. The decentralized approaches can be divided into two categories, namely purely distributed and hybrid.

The purely distributed approaches enable peers, participating in a network of cooperating sites, to store locally their own service registry. Then, service retrieval is facilitated by distributed querying services that span all these local registries to compile and present answers to users. Naturally, the distributed setting provides a richer set of answers and significant chances of better scalability in terms of both the user load and the available data to index, without the risk of a single point of failure that the centralized solution suffers from. This comes at the cost of supporting a framework that allows each peer to know which peers to contact for serving a user request along with the necessary communication overhead whenever a query to the decentralized virtual registry is posed. To overcome these issues, a possible solution is to employ multicasting instead of broadcasting [96]. Nevertheless, even multicasting is a costly approach that does not scale well [135]. Another possible way out of this problem is to limit the number of search hops [37] and perform selective forwarding [37, 59, 60]. According to this idea, knowledge from prior service discovery queries is used by peers that cannot satisfy a certain service discovery query, to forward the request only towards the correct peers (i.e., the ones that can actually provide answers to the query). An alternative to selective forwarding, which also aims at reducing the overhead of service discovery, is the probabilistic forwarding of queries to peers characterized with a probability that decreases with the number of prior queries that remained unanswered [61]. Finally, efficient broadcasting techniques have also been used [81]. The idea in these techniques is to exploit knowledge about 2-hop peers to calculate a forwarding set that does not include 2-hop peers that are covered by multiple 1-hop peers.

The *hybrid approaches* involve a more elaborate service registry architecture with super-peers that act as yellow pages and/or heads of sub-structures within the P2P network. Super-peers facilitate the efficiency and

effectiveness in the serving of service discovery queries (by achieving a reduction to the messaging overhead in the network and better response time, without sacrificing the richness of the answer). In other terms, these approaches are practically "federations" of many registries in a P2P setting [122, 79, 74, 116]. Focusing on hybrid architectures that are specifically aimed at mobile environments, an important research issue is the choice of the peers that are going to play the role of the registry. To deal with this issue, there have been approaches for the dynamic configuration of a backbone network of peers that serve for hosting registries [76, 116]. Typically, the peers are selected based on their physical mobility, their resources and computing capabilities; while the physical mobility should be low, the available resources and computing capabilities should be relatively high.

Apart from scalability, availability and mobility, another important issue that directly relates to service discovery protocols is heterogeneity. In this line of research, there have been approaches that rely on the idea of providing a higher middleware layer on top of heterogeneous service discovery protocols that takes charge of mapping service discovery queries that conform to a particular protocol into service discovery queries that conform to other protocols (e.g., ReMMoC [62], INDISS [23], MUSDAC [111]). Moreover, the proposed approaches further provide functionalities that allow the forwarding of the mapped queries to available heterogeneous discovery protocols.

Finally, security, privacy and trust are also important issues for service discovery. Typically, decentralized architectures are more vulnerable to security attacks, while privacy and trust cannot be easily established [51]. An interesting classification of privacy attacks along with an approach for dealing with such problems is discussed in [32], while in [47] the authors give a list of requirements that should be provided by service discovery protocols towards achieving security. Interestingly, a comparison of various well-known service discovery protocols (e.g., UPnP, SLP, Jini) discussed in [47] reveals that the protocols only partially satisfy the aforementioned security requirements.

4.2 Research Challenges

We anticipate a hybrid structure for the Future Internet, with computationally powerful devices and lightweight computing devices coexisting together. As will become evident from the discussion that follows, the increasing scale of the Future Internet is a challenge by itself. Scalability also constitutes a driving force that creates further challenges concerning heterogeneity, mobility, awareness, and safety.

Scalability. The ultra large scale of the Future Internet affects service discovery regarding the number of devices that act as service providers or service consumers, together with the number of services that a registry will have to index (e.g., see indicative figures given in Table 2 with respect to the exponential growth of the service base).

A first key challenge is to quantify the number of actors involved, if possible. We envisage that this can be done by systematically studying the growth of the Future Internet via advanced monitoring and Web crawling facilities. Today, the first steps have already been made with certain research prototypes that crawl the Web for available Web services and further analyze the validity of the retrieved data [3, 124]. Subsequently, from an architectural point of view, it is clear that centralized discovery solutions are not adequate. However, choosing between purely distributed and hybrid solutions is not straightforward. There is a trade-off between making the environment as large as possible (to accommodate as much functionality as possible) and keeping the number of registries as small as possible (to receive answers faster). From the point of view of the registry data-model, clearly, the ideal would be a semantically rich schema, as discussed in Section 3, which allows the organization of services that provide similar functional/non-functional properties into classes, each with a clear, but abstract specification of the properties that characterize the represented services. However, the challenging issue is to define such a hierarchical schema. Defining a standardized schema amounts to anticipating the various classes of services that will emerge in the Future Internet. Again, this requires the Future Internet growth to be systematically studied. Today, some initial steps have already been made as part of certain research efforts that mine semantic information from available service descriptions [94, 80, 54] and try to organize available services into classes characterized by service abstractions [8, 9]. Finally, the clients should be able to identify services with (a) the appropriate functionality, and (b) the appropriate QoS characteristics. Service discovery protocols should provide means to accomplish this task, irrespective of whether the client is a programmer/designer [125] or an end-user [141].

Heterogeneity. The primary dimensions of heterogeneity that should be handled in the Future Internet service discovery protocols are:

- Middleware platform heterogeneity, which may emerge due to the availability of various service dis-

covery protocols and data-models that can be exploited towards advertising/discovering services;

- Semantic heterogeneity, which relates to the advertised/discovered descriptions of the services' functional/non-functional properties.

Dealing with middleware platform heterogeneity supposes an agreement on the protocol via which services are advertised and discovered. We anticipate standards to play a key role in the solution. Assuming a set of standard service discovery protocols, there are promising approaches that propose automated mapping facilities from a source discovery protocol to a target one [20,62,23,111]. Nevertheless, these approaches should be further elaborated with advanced extensibility mechanisms that would allow them to adapt easily to the availability of new protocols. On the other hand, semantic heterogeneity may produce huge problems if not properly addressed early in time. Even if we assume that the service discovery protocol and data model employed will be commonly agreed on, the vast space of possible choices of available services will still be posing problems to the precision and recall of the service discovery. A discovery engine will have to resolve problems of (a) naming, (b) structural and (c) value heterogeneity.

Mobility. Mobility is another issue that should be revisited, considering service discovery in the Future Internet. The main dimensions of mobility are:
- Physical mobility, which relates to the movement of mobile entities from one location to another;
- The resources and the computational capabilities of the mobile entities.

Concerning physical mobility, a challenge set by the Future Internet is the trade-off between physical mobility and scalability. As discussed in [135], purely distributed service discovery protocols are more suitable for environments consisting of a small number of devices that are characterized by high physical mobility. On the other hand, hybrid protocols are more suitable for environments consisting of a medium/large number of devices that are characterized by medium/low physical mobility. In the Future Internet, the number of available devices is expected to be ultra large, while the physical mobility may obviously range from low to high. Consequently, the choice of the right service discovery protocol is not straightforward and potentially none of the existing approaches are suitable for dealing with the most extreme cases. Regarding the limited resources and the computational capabilities of the mobile entities, there have been related service discovery approaches [18,19], which should, however, be further elaborated and tested in ultra large scale settings.

Awareness & Adaptability. Awareness and Adaptability can be seen from two different perspectives:

- The service discovery protocol itself, in the sense that the protocol should be ready to be used by service providers/consumers at any given moment in time and consequently it should be aware of the current conditions (e.g., number of consumers/providers, failures) and adapt accordingly;
- The discovered/published services, in the sense that the protocol should be aware of the availability of the services that are indexed (e.g., outdated, or missing references to services) and adapt its content, possibly along with applications that use unavailable services that have been previously discovered.

The expected increased scale of the Future Internet forces us to move from centralized to decentralized service discovery architectures, which further contribute to the availability of the service discovery protocol itself. However, the issue here is that as we move from centralized to decentralized service discovery architectures, it becomes much harder to control the quality of the information managed by the discovery protocol. In other words, in the Future Internet we have to deal with a trade-off between the availability of the discovery protocol itself and the availability of the services, discovered/published, via the protocol.

Security, Privacy & Trust. All the problems of security, privacy and trust in the traditional Internet setting are clearly present in the context of the Future Internet. Unfortunately, even in the current setting, and especially in environments that include mobile devices, the main requirements for security, privacy and trust are not completely covered [135]. In addition, in the Future Internet, the issues become much harder and more challenging due to the vast numbers of devices involved. Regarding security, an attack can involve numbers of hijacked devices that are orders of magnitude larger than the current ones; and the effect of introducing malicious code (i.e., discovered services) in the environment can be orders of magnitude higher. Concerning trust, there may be a huge number of entities that should be evaluated (e.g., registries, service providers, service clients), with respect to an enormous amount of conflicting opinions concerning the reputation of these entities. The difficulty of these issues has to do with the fact that they are orthogonal to all authentication, encryption and anti-virus mechanisms we have, as they are related to scale. Much like the case of the present search engines, another problem has to do with the privacy of client searches. Assuming that registries trace the client searches, a breach of these audit traces will

expose clients' searches and, thus, their privacy. This then requires adequate mechanisms for controlling the disclosure of private user information carried in service discovery requests, as investigated in [31].

5 Service Access

Following a possible client request and subsequent localization of a matching service by service discovery, and based on information about the service externalized in its service description, service access enables actual interaction with the service despite possible heterogeneity of service clients and providers.

5.1 State of the Art

Services come from different sources, and are developed in different ways and, in general, without any coordination among developers, according to the fundamental SOA loose-coupling principle. Regarding service access, and focusing on the underlying middleware aspect, uniformity of employed SOC technology and middleware cannot be assumed in general. Therefore, an integration solution at the middleware level is strongly required. Enterprise Application Integration (EAI) and Enterprise Service Bus (ESB) emerged as integration paradigms targeting the enterprise domain. EAI appeared in the early 1990s followed by ESB in the early 2000s. While EAI proposes a centralized integration of services, ESB comes with a more decentralized and scalable vision.

5.1.1 Service Integration Architecture

The main goal of EAI solutions is to provide integration of heterogeneous applications. Both EAI and ESB share the same philosophy for achieving integration. The difference between them is that ESB is standards-based, distributed and focuses on Web services. The architecture of ESB and EAI middleware is similar. In this section, we particularly detail the ESB architecture as providing a capable integration infrastructure where fundamental Web services and SOA concepts coalesce.

The ESB technology is based on an open-standard message backbone that enables the implementation, deployment and management of services. Typical ESBs are able to support large numbers of services and high distribution via scalable integration infrastructures, which act as mediators between service providers and consumers (see Figure 5). Specifically, the ESB technology is based on the following key mechanisms:
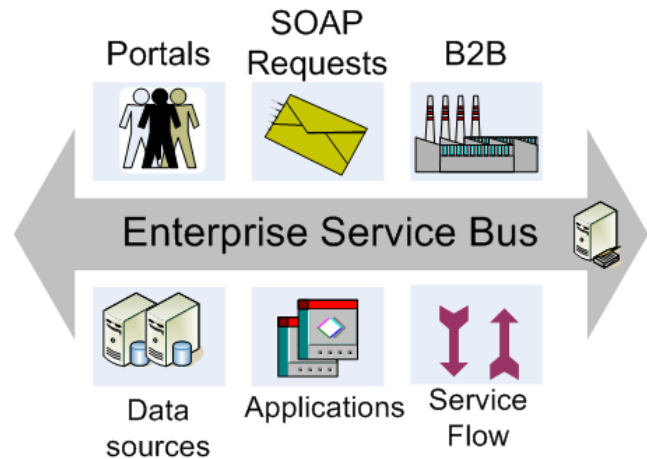

Fig. 5 Enterprise Service Bus

- Services are discovered dynamically thanks to a common registry where semantic service descriptions are stored and retrieved (see Section 4). The registry of the bus stores the physical addresses of the services, along with meta-data that relate to the service providers and the service requesters that use it.
- Business processes and services are choreographed and orchestrated using a powerful orchestration engine (see Section 6). The engine is the cornerstone mechanism of the typical ESB solutions.
- Communication between services and applications is realized through XML-based messages, which are stored in a queue until their consumption by service clients.
- Mediation patterns are realized for routing, transformation, encoding and mapping of messages. These patterns serve for the manipulation of messages issued from an application to another, hence overcoming mismatches arising from application heterogeneity.

In his book, Chappell [39] presents a list of ESB key features. We present the most relevant ones with regard to the Future Internet requirements:

- *Pervasiveness and scalability.* The ESB technology allows several ESBs to be connected in a large network, achieving a pervasive integration, which is strongly required in wide and dynamic systems. It can provide a core for a pervasive grid where applications can be plugged to the bus and where the visibility of other applications and services is ensured.
- *Autonomous and federated environments.* In the Future Internet, ultra large scale systems shall consist of heterogeneous applications, services and re-
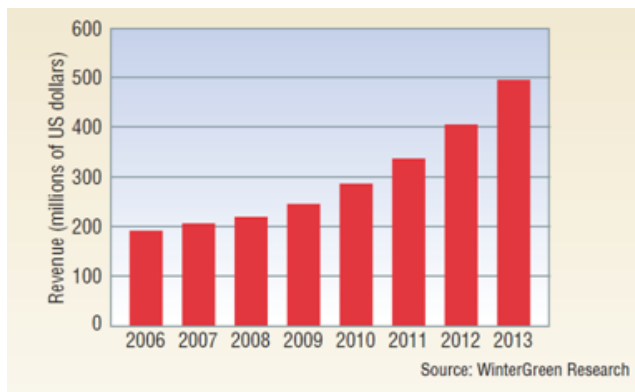
Fig. 6 Worldwide ESB Revenue from 2006 till 2013 [98]

sources. The ESB technology supports the integration of applications that are independently designed and developed. However, related local business and IT units usually need to keep control over their own resources and applications. This is achieved in ESB by allowing local messages, components and connectors to be installed, configured, and managed locally. Thanks to this deployment model, the ESB ensures the autonomy of each single environment within a large federation.

– *Operation awareness.* Thanks to the use of semi-structured data, the ESB technology can have real-time awareness of the data flowing through the enterprise. In large systems where an important number of partners and applications are involved, data flows are important and it is very difficult to achieve individual monitoring of the system. Based on the relevant indicators and track information that the ESB technology provides, it is possible to monitor the health of the system.

In addition to the aforementioned features, remote configuration and management of distributed ESB applications and services are also provided. The next subsection presents state-of-the-art solutions to service integration, while concentrating on industrial products, given the maturity of the technology. The industrial success of the ESB technology is in particular highlighted in Figure 6, which depicts the evolution of the ESB revenue in industry. In fact, according to a research undergone by the WinterGreen incorporation[17], the worldwide ESB market will grow, steadily doubling from 2006 till 2013 [98].

5.1.2 Service Integration Solutions

Existing proprietary and open source ESB solutions provide powerful integration and orchestration solu-

tions for distributed services. For instance, the BizTalk server[18] is a proprietary Microsoft product, which offers a toolkit providing a collection of tools and libraries supporting a loosely coupled and dynamic messaging architecture. It works as a middleware that provides tools for mediation of services and their consumers. The Open ESB[19] is a java-based open source tool for application integration. It supports open standards such as SOAP, WS-*, XML, and a NetBeans-based Integrated Development Environment (IDE). Both BizTalk and Open ESB rely on centralized architectures. Meanwhile, even though this generation of ESBs provides a powerful means of integration and service orchestration, it is still not suitable for highly distributed environments. Therefore, a more flexible and dynamic vision is needed to cope with the scalability requirement of Future Internet services, which is supported by distributed ESBs.

The Fuse ESB[20] is an open source OSGi-based[21] distributed ESB that is based on the Apache Service Mix ESB[22]. Fuse supports BPEL processing, and both OSGi and JBI[23] deployment and runtime. In a sense, it is characterized as being distributed, because it offers a remote console to control the runtime bus. The Fiorano ESB[24] is a proprietary product that has a different and more scalable vision of distribution. It is built upon a hybrid architecture, relying on a hub and a spoke management layer and a peer-to-peer system. This enables ESB peer distribution over a wide network. Fiorano supports the main ESB features, i.e., application integration, service orchestration, event management, etc. The Sonic ESB[25] is another example of a proprietary, distributed ESB. It achieves service integration and distributed operation management. Moreover, Sonic focuses on security in distributed domains and achieves BPEL orchestration. As a final example, the Petals ESB[26] is an open source JBI-based distributed ESB. Petals combines both distributed service registry and multi-site architecture. A multi-node platform is provided leading to a moderate scalable architecture. Petals handles BPEL process orchestration, business process monitoring and management, as well as heterogeneous application integration through several protocol connectors. Around the bus, a set of open source tools providing process design and configuration, service management and monitoring, are offered. Distri-

---

[17] http://www.wintergreenresearch.com/

[18] http://www.microsoft.com/biztalk/
[19] https://open-esb.dev.java.net/
[20] http://fusesource.com/
[21] http://www.osgi.org
[22] http:servicemix.apache.org
[23] http://jcp.org/aboutJava/communityprocess/
[24] http://www.fiorano.com
[25] http://web.progress.com/en/sonic/sonic-esb.html
[26] http://petals.ow2.org/

bution is also addressed by leveraging the bus service registry. Although the current ESB approaches tackle the distribution and integration issues required by the Future Internet, a larger vision is required, as discussed next.

5.2 Research Challenges

The unique combination of features and the high diversity of the Future Internet raise numerous challenges for service access. While heterogeneity – as discussed in detail in Section 5.1 – remains a principal challenge and becomes an even harder one for service access in the Future Internet, there are additional challenges in the Future Internet calling for advanced solutions to be provided by SOC communication middleware. Research results from different domains like pervasive computing, Grid and Cloud computing can offer the base for such solutions, nevertheless, significant progress is required in order to cope with the Future Internet issues. In the following, we discuss challenges and identify research directions towards supporting service access in the Future Internet.

Scalability. With respect to service access, the ultra large scale of the Future Internet translates into ultra high number of interacting entities, as in parallel single service accesses or coordinated service accesses within a service choreography, and ultra high service load, i.e., number of concurrent user accesses, for certain services.

From the standpoint of scalable interaction, and leaving aside issues that concern the Future Internet network infrastructure, such as communication bandwidth and IP routing, solutions for SOC communication middleware need to be devised and underpinned by wide-area and highly decentralized networking paradigms. Taking the example of the ESB paradigm extensively discussed in Section 5.1, which provides a multi-faceted SOC middleware, we can identify the potential of distributed ESBs to provide the base for wide-area decentralized networking. Nevertheless, they need to be extended in order to deal with the demanding scalability requirements of the Future Internet. In this direction, work in [16] concerns the evolution of the Petals distributed ESB towards an ESB federation in order to support large-scale SOA. This extension includes scaling up message routers and service registries to the level of federations. Besides ESBs, other networking paradigms that have been applied to wide area and decoupled settings should be considered, such as peer-to-peer systems and publish/subscribe systems [146] or tuple space systems [5].

With respect to ensuring scalability in dealing with user load, paradigms for high-performance computing and computing based on resources retrieved on demand, such as Grid and Cloud computing, seem to be very promising to be employed by SOC middleware. Hence, there has been research for many years on Grid middleware solutions, e.g., OurGrid [42], InteGrade [120], to enable the execution of computationally-intensive applications on sets of geographically distributed clusters of machines. Originally targeting the scientific community with their large data processing needs, Grids can provide solutions to any computing-intensive application. On the other hand, Cloud computing [148] is a more recent paradigm providing virtualization mechanisms for supporting elastic and on-demand provision of remote networked resources at different levels, such as infrastructural resources (Infrastructure as a Service - IaaS), higher-level service components for building applications (Platform as a Service - PaaS), or even complete applications (Software as a Service - SaaS). This enables the support of everyday Web applications that are used by hundreds of millions of users. There exists a number of commercial (Amazon EC2[27], Google App Engine[28]) and research (OpenCirrus [25]) Cloud computing solutions.

While some of the aforementioned paradigms and technologies – i.e., Grids and Clouds – have proved their ability to scale, the Future Internet introduces such unique scalability requirements that new SOC middleware solutions need to be devised. For instance, besides employing the pool of Grid and Cloud computational resources for covering demanding service loads, some of the high processing needs of a global-scale ESB middleware could be also served in the same way.

Heterogeneity. Among the heterogeneity dimensions of the Future Internet of services discussed in Section 3.2 (resources, business semantics, communication middleware), we focus herein on the latter, while we discuss resource constraints in the paragraph on *Mobility, Awareness & Adaptability* below. As already pointed out, the Future Internet networking context will be characterized by high diversity in terms of communication middleware, particularly with respect to employed coordination/interaction models, e.g., remote procedure call (RPC), message-based, shared memory, event-based models. Hence, as entities interacting in *ad hoc* settings cannot be assumed to share the same coordination model, service access in the Future Internet is required to support heterogeneous models and to enable interoperability among them.

---

27 http://aws.amazon.com/ec2
28 http://code.google.com/appengine

Distributed system interoperability approaches at the middleware level are typically based on bridging communication protocols, wrapping systems behind standard technology interfaces, or providing common API abstractions. Most of these efforts focus on a single coordination model, which is already a difficult problem. Nevertheless, a number of approaches attempt to combine diverse models. Common API abstractions enable the development of applications that are agnostic to the underlying coordination models. Then, some local mapping is performed between the API operations and the models/protocols supported. In this category, ReMMoC [63] is an adaptive middleware for mobile systems, enabling clients to interact with both RPC-based servers and publish/subscribe systems via a common programming interface. Wrapping systems behind standard technology interfaces enables the access to these systems via coordination models different from their native ones. In [11], a gateway allows high-level access to the data and operations of a wireless sensor network via Web service interfaces.

Bridging is about interworking between heterogeneous interaction protocol stacks. The ESB paradigm, extensively discussed in Section 5.1, is currently the dominant bridging solution for the integration of heterogeneous systems. By employing appropriate ESB adapters, systems with diverse coordination models can be plugged on the bus. For instance in [16], an external tuple space is connected through adapters to a distributed ESB topology and is accessible via the bus messaging-based interface. Acknowledging the flexibility of the shared memory model, certain efforts (both academic and industrial) introduce extended tuple spaces as an alternative solution to the realization of the ESB paradigm ([29], [92]). Some of these ESBs offer various coordination semantics (by emulating different coordination models) and related APIs, such as RPC and event-based in addition to shared memory.

All the above interoperability solutions are deployed statically, with the exception of ReMMoC, which enables the dynamic change of the client protocol based on the detected server protocol. Other efforts have aimed at providing dynamic transparent interoperability between legacy systems. Such solutions are based on the runtime configuration and deployment of bridging mechanisms in response to the detection of systems that seek to interact via incompatible protocols (INDISS [23], uMiddle [93]). However, interoperability between different coordination models is not addressed.

Despite a number of approaches dealing with interoperability between coordination models, provided solutions are in general *ad hoc* and concern specific cases.

In the Future Internet setting, an overall solution to this issue is required, based on appropriate modeling abstractions and transformation mappings between models. Moreover, a precise evaluation of such mappings with respect to the preservation of semantics is needed. Deployment of the solution in the rich Future Internet context should then be addressed, in particular taking into account composition patterns of Future Internet entities, such as service choreographies, as well as the scale and dynamics of such compositions. Dynamics of the Future Internet are discussed in more detail in the following.

Mobility, Awareness & Adaptability. The wireless and – to a large extent – dynamic Future Internet setting encompassing the Internet of Things creates new user behaviors and expectations as well as challenges for service access in terms of Mobility, Awareness & Adaptability. Mobility – if we leave aside issues that concern the Future Internet network infrastructure, such as seamless mobile addressing, routing and communication – implies users moving freely and employing their resource-constrained handhelds to access local or remote Future Internet resources. These resources may themselves be wireless or mobile, such as other users' handhelds, sensors and actuators, or simply networked Things. In this setting, communication patterns are unpredictable with, most probably, frequent disconnections. Awareness & Adaptability are more general notions that include Mobility, and may be considered from two viewpoints:

1. SOC middleware for the Future Internet should be able to capture the dynamically changing conditions and resource limitations of the underlying networking environment and the interacting networked entities and to adapt accordingly; and
2. SOC middleware should provide services with support for user context-awareness and related service personalization and adaptation.

The above identified challenges point to research that has been undertaken, for quite a few years or more recently, in the domains of mobile and pervasive middleware as well as middleware for wireless sensor networks.

A number of research efforts focus on architectural models [112] as well as interaction models [87] for pervasive applications. Such models aim to be modular, flexible and dynamic in order to enable applications to deal with the uncertainty and dynamics of the mobile/pervasive environment. Other approaches explicitly adopt the SOC paradigm and provide "aware" solutions to service/resource discovery, access and composition that are customized to the specifics of pervasive computing. Thus, in [28], access to services hosted

---

by mobile devices is conditioned by the multi-radio, multi-network character of pervasive device communication. Similarly, efforts in [17,71,84] take into account the rich semantics and, at the same time, the resource constraints of pervasive nodes, thus employing efficient mechanisms for service composition. Context is a key element for awareness and adaptability. In [10], a solution to dynamic management of context sources is proposed, while middleware facilitating the development and execution of context-aware applications is devised in [64] and [70]. Then, a number of efforts focus on dynamic adaptation, which may be context-aware [140] or QoS-aware [41], and in particular energy-efficient [40]. Finally, while pervasive computing and wireless sensor networks started out as separate fields, there is a strong convergence towards middleware-based solutions that combine the two paradigms. Thus, numerous research approaches propose programming abstractions and middleware for wireless sensor networks [46,45,86], or focus more particularly on adaptation in sensor networks [68,130].

Despite these rich research results new work can be based upon, the Future Internet setting introduces unique requirements in terms of openness and hence awareness and adaptability, which make existing solutions fall short. For instance, wireless sensor networks still remain mostly closed systems, accessible to the rest of the world through external gateways [11]. On the other hand, with sensors being increasingly pervasive – such as sound, GPS, accelerometer and other sensors attached to users' smartphones – and the anticipated evolution to the Internet of Things, sensing becomes ubiquitous and "participatory" as each user can be involved [82,53]. This fact, together with the resulting scale, creates new needs for aware and adaptable programming abstractions and underlying middleware.

Security, Privacy & Trust. Being a key issue in the current Internet, the problem of ensuring Security, Privacy & Trust will be exacerbated, particularly in the mobile and aware Future Internet, due to two main reasons:

1. In open mobile and wireless settings, interaction is often *ad hoc*, among entities that have no prior knowledge of each other, and reliance on infrastructure that can provide trust guarantees for interacting entities cannot in general be assumed.
2. Awareness and personalization imply revealing and collecting plenty of information about the user, which is a direct compromise of the user's privacy.

Hence, SOC middleware should incorporate mechanisms for supporting Security, Privacy & Trust in open, dynamic, *ad hoc* and aware environments. Besides, the trade-off between exposing user-related information and preserving user's privacy should be a central consideration.

In this direction, ongoing research focusing on Security, Privacy & Trust in pervasive computing environments is very relevant for the Future Internet. A principal consideration is that security, privacy & trust should be closely intermingled in such environments. Hence, a number of authentication frameworks and related access control schemes have been introduced, based on trust relations and privacy preservation among interacting entities [113,131,1]. Then, to cope with open, *ad hoc* interactions, proposed trust models and frameworks are based on context [114,137,134] or reputation [91]. In other cases, trust establishment needs to be performed through trust negotiation, which presents a threat to privacy due to the associated disclosure of sensitive information. Then, work in [126] introduces a solution to privacy-preserving trust negotiations. Finally, other privacy concerns are tackled in [73], where users can control external access to private sensor data by enforcing their privacy preferences.

In accordance with these research results, it should be pointed out that security deployment in the open and aware Future Internet should be flexible and dynamic, conditioned by privacy needs and trust requirements. Still, as already stated, the openness, diversity, scale, and extreme awareness of the Future Internet create new unique conditions, calling for new solutions. For instance, the increased proliferation of sensors often acting out of the user's access or attention, although enabling advanced context-aware personalized services, makes it extremely difficult for the user to control the amount of their personal information that is indirectly revealed, collected and used.

6 Service Composition

If no single service satisfies the specific request of a user, it may be necessary to compose existing services [110]. Resulting composite services can be used as basic services in further hierarchical service compositions or offered as complete applications to service clients [102].

Briefly stated, the process of (automated) service composition is as follows. The service requester defines the requirements (in a specification language) and the service composition engine (based on the requirements, the services available and their current state) generates the composition, which is deployed in the execution engine of the middleware (see Figure 7). The middleware may also monitor the execution of the composition and the state of the services in order to enable the redesign
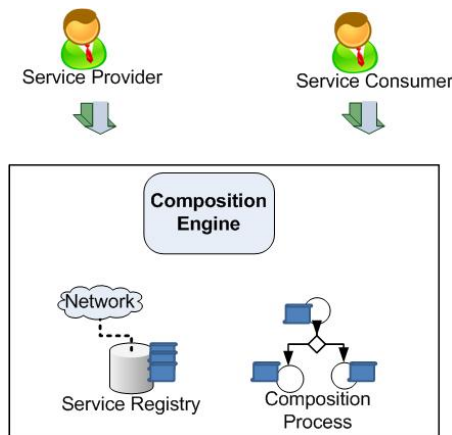
Fig. 7 Service Composition

of the composition when the QoS or the requirements are not fulfilled anymore. This process is known as dynamic adaptation or self-healing composition.

Service composition is a very complex and challenging task, which has received a lot of attention in the literature, from modeling to runtime concerns, as surveyed below. Still, the Future Internet raises additional challenges for service composition, especially with respect to handling the scale and heterogeneity of the target networking environment although, as discussed in the previous sections, related issues are partly covered by the underlying middleware functions for service discovery and access.

6.1 State of the Art

In the last few years, building upon the wide acceptance of the service-oriented architecture paradigm, there is a growing interest in choreography as a key concept in forming complex service-oriented systems. Choreography is put forward as a generic abstraction of any possible collaboration among multiple services, and integrates previously established views on service composition, among which service orchestration. More specifically, we find in the literature three distinct but overlapping viewpoints (often denoted with varying terms) [50, 13]:

1. *Choreography* captures collaborative processes involving multiple services and especially their interactions seen from a global perspective;
2. *Behavioral interface* captures the behavior of a single service that participates in choreography; and
3. *Orchestration* deals with the description of the interactions in which a given service can engage with other services, as well as the internal steps between these interactions.

Based on these three viewpoints, one finds in the literature different styles of choreography. Interaction-oriented models describe choreography as a set of interactions between participants. Process-oriented models describe choreography as a parallel composition of the participants' business processes [49, 77]. Activity-based models focus on the interactions between the parties and their ordering, whereas the state of the interaction is not explicitly modeled or only partly modeled using variables. State-based models model the states of the choreography as first-class entities, and the interactions as transitions between states [136].

Choreography Modeling. A number of conceptual models and corresponding languages have been proposed for modeling/describing choreographies at different conceptual levels, which originate from the Web Services and Electronic Business initiatives. Proposed languages include W3C's Web Services Choreography Description Language (WS-CDL)[30], which has been a W3C candidate recommendation since 2005. Still, Web Services Business Process Execution Language BPEL, which is the major widely accepted industry standard for modeling service composition, overtaken by OASIS, can be conveniently exploited in the context of choreography modeling to specify abstract processes for behavioral interfaces. BPEL4Chor [49] extends BPEL for defining choreographies, by introducing an interconnection layer on top of abstract BPEL processes, thus leading to interconnected behavioral interface descriptions. As BPEL itself is used unchanged, the BPEL4Chor extensions facilitate a seamless integration between service choreographies and orchestrations. Let's Dance [142] is a visual choreography language derived from workflow and architecture description languages and targeted at business analysts; this language is not linked to any imperative programming constructs. Acknowledging the need to give unambiguous semantics to proposed choreography models/languages, there have been several research approaches on defining the formal semantics of choreographies with a special focus on WS-CDL [78, 38, 147, 138], although other languages deserved attention, such as WSCI [67].

Formal models and languages have further been introduced with the aim of verifying conformance between global choreography and local behavioral interfaces; thus, the ability of services to take part in a given choreography can be verified. Related research approaches have addressed: checking consistency between ebXML BPSS choreography and BPEL orchestration based on CSP [139]; reducing the complexity of conformance verification based on an extension to the pi-

---

[30] http://www.w3.org/TR/ws-cdl-10/

calculus for enabling resource-constrained mobile devices to participate in a choreography [118]; verifying a WS-CDL choreography against service behaviors expressed in BPEL based on Finite State Process and Labeled Transition Systems [58]. From the methodological standpoint, OASIS' and UN/CEFACT's Modeling Methodology (UMM) [66] is based on UML (defined as a UML profile) and addresses the analysis of the business environment, the requirements of each partner, and the requirements for an inter-organizational collaboration. The OMG's Business Process Management Initiative (BPMI) further considers the transition from the business process design (supported by the OMG's Business Process Modeling Notation BPMN[31]) to the business process implementation. Thus, the mapping from BPMN to XML-based executable business processes (e.g., BPEL) is part of the standard. As part of the related OMG's continuous standardization process, BPMN 2.0[32], the next version of BPMN, has been recently released. This update enriches BPMN with a comprehensive embedded meta-model and related graphical notation and interchange format. This will improve the capability for business analysts to develop, communicate, and understand business process models, and also related tool development.

Last but not least, transformations between models and languages have been proposed for enabling top-down model-driven generation of behavioral interfaces from choreographies. In [65], UMM is projected to a local orchestration model developed as a UML profile; then, this model is transformed into an executable orchestration expressed in BPEL. Kang *et al.* [115] integrate QoS aspects in a top-down choreography development process, annotating WS-CDL choreographies with SLAs and further generating a BPEL orchestration for each partner along with WS-QoS policies that can be enforced during the orchestration execution.

Choreography Execution. Enacting choreographies raises the issue of choreography execution. While execution has been well addressed for business orchestrations based on BPEL with a multitude of execution engines, often free and open source (e.g., Apache ODE[33], Orchestra[34], ActiveBPEL[35], Oracle BPEL[36]), choreography is mostly considered as a design artifact rather than an implementation artifact. Nevertheless, reference [72] proposes a (probably unique) WS-CDL execution engine aimed at enabling the testing and

evaluation of the properties of WS-CDL with practical use cases. The engine enables the simulation of choreographies on a single machine, which facilitates testing and debugging WS-CDL documents. This work additionally extends WS-CDL to WS-CDL+ to resolve some usability weaknesses. Regarding the handling of composing heterogeneous services, as discussed in Section 5, ESB platforms that come along with a BPEL engine constitute the widely accepted emerging solution.

Considering the increasing dynamics and openness of the networking environment, dynamic service composition has now become a key concern. Many approaches have been proposed in the literature aiming at automatically composing services by means of BPEL-, WSCI-, or the latest W3C choreography pattern candidate, WS-CDL -choreographers [22, 24, 88, 89, 17, 103, 117, 129]. The common idea underlying these approaches is to assume a high-level specification of the requirements that the choreography has to fulfill and a behavioral specification of the services participating in the choreography. From these two assumptions, by applying data- and control-flow analysis, the BPEL-, WSCI- or WSCDL- description of a centralized choreographer specification is automatically derived. This description is derived in order to satisfy the specified choreography requirements. In particular, Su *et al.* [129] propose an approach to derive service implementations automatically from a choreography specification. The authors of [56] and [121] present different approaches to semi-automatic service composition based on abstract functional blocks and semantic service descriptions, respectively. Ponnekanti and Fox [106] propose an automatic approach for service composition, using AI planning algorithms. Salaun [117] strives for the same goal, however assuming that some services are reused. The proposed approach exploits wrappers to make the reused services match the specified choreography. Some studies [12, 109, 35, 150] investigate dependency management in a dynamic service composition scenario. Employed techniques include service and middleware instrumentation, use of self-healing rules, and establishment of a dependency-aware service-oriented architecture.

In addition, concerning the increasing dynamics and openness of the networking environment, there is a particular line of research that specifically focuses on the dynamic service composition with respect to QoS requirements. In the typical composition approaches, a system is considered as an abstract orchestration, where each particular task may be performed by a given set of alternative services that can serve as substitutes for each other. The quality of different alternative services

---

[31] http://www.omg.org/spec/BPMN/1.2/

[32] http://www.omg.org/spec/BPMN/2.0/

[33] http://ode.apache.org/

[34] http://orchestra.ow2.org

[35] http://www.activebpel.org

[36] http://www.oracle.com/appserver/bpel_home.html

is characterized in terms of different types of quality attributes (e.g. reliability, availability, reputation, cost, performance). The quality attributes are associated with corresponding metrics, the values of which are usually continuous and normalized, using typical normalization techniques. The overall quality of the system is measured with a global score function, whose values are calculated, with respect to the values of the quality attributes of the orchestrated services. Then, the objective of the composition approaches is to calculate an optimal configuration for the orchestration that satisfies the system's requirements, while maximizing the value of the global quality function. Obviously, all of the approaches in this line of research may be used multiple times at runtime towards dynamically adapting the configuration of a composition to deal with changes in the QoS requirements, or changes in the quality attributes of the composed services. A pioneer approach in this line of research was proposed in [144]. The global QoS optimization problem is solved using an integer programming optimization technique. In [7, 6] the authors further consider cases where the same service should be assigned to different dependent tasks. Differently from the aforementioned approaches, in [27] the authors employ a genetic algorithm towards solving the global QoS optimization problem involved in the composition of a given service orchestration. Moreover, in [145, 84] the authors propose methods that find sub-optimal solutions to the global optimization problem. The primary motivation for the proposed methods is that the complexity of the methods that find optimal solutions is very high. Going one step further, in [29, 30] the proposed approach solves the global QoS optimization problem for sets of independent orchestrations. Finally, in [34] the authors consider choreographies of services. Specifically, they propose a method for the composition of systems that consist of tiles. A tile corresponds to a particular service that may require using other services. The goal is to find the optimal composition of tiles, based on a global quality score function. Similarly to approaches that focus on orchestrations, the proposed approach also exhibits poor performance, as the number of tiles scales up.

6.2 Research Challenges

The industry has high performance and availability requirements on services when they are deployed in a production environment. They want the services to be scalable, robust, measurable (so that service usage can be charged), secure, and verifiable [127]. The challenges for service composition may be seen from two different perspectives: design and execution. The former is

important because of the necessity of automatic dynamic adaptation that may occur during the composition execution. The latter suffers impact from the scalability, heterogeneity, mobility, awareness, and safety issues present in the Future Internet scenario, as discussed below.

Scalability. The decentralization and pervasiveness of adaptable, QoS-aware highly-scalable compositions demand new service composition paradigms. In an ultra large scale scenario, it is still not clear how to visualize and manipulate the huge compositions that are formed. To cope with the challenges of the Future Internet, the composition languages and the execution platform have to deal with this scalability issue.

Although extensively investigated, the problem of designing loosely coupled compositions consisting of ultra large numbers of participating services [69] is still an open challenge when dealing with the Future Internet context. It is not yet clear how to combine the need to aggregate several services, maintain the QoS, and keep the whole composition coupling level as low as possible. Software Evolution research has already discussed the benefits of low coupling [90]. When there are fewer dependencies, the system becomes more flexible and modifications or faults have fewer consequences on others systems. It is still complicated and time consuming to implement, test, and debug low-coupled service-oriented systems with current strategies and tools.

With the emergence of the Cloud Computing paradigm, the investigation on how to distribute an ultra large number of services in the global Internet and delegate heavy computational tasks in a semi-automatic fashion becomes a necessity. The distribution needs to minimize costs financially or in terms of resource usage. It is also necessary to maintain and negotiate QoS to cope with the users' requirements.

Heterogeneity. In the Future Internet context, heterogeneity is particularly high and mechanisms to provide interoperability are necessary. Services are developed by different organizations, which use different concept models and technologies. The service compositions should be realized in a manner independent of programming languages, vendors, operating systems, data models, etc. As in the case of service description, discovery and access discussed in the previous sections, more research is still needed to cope with these issues and the exploitation of ontology technologies may be a promising research direction [119].

Mobility. In the Future Internet, the execution of ultra large scale self-adapting compositions may involve

devices with limited resources and computational capabilities. Thus, the algorithms for designing and dynamically adapting the compositions need to be conceived to be efficient. Existing strategies used for dynamic composition and adaptation should be revisited and further investigated in this context. In an ultra large scale scenario, where nodes may be mobile and intermittent, new approaches are necessary for coordinating service compositions. Coordination needs to be adaptable to topology changes of the mobile environment and handle mobility, network disconnections and other types of node or service failures [36]. The coordination needs to consider mobility patterns, platform battery lifetime, fault tolerance, and reliability. Broadcast-based messages, for example, should be avoided since they generally impose a high network load. Mobile devices often have a short switched-on time and often are unable to process remote requests due to system overload or network level disconnections. The composition and the coordination protocols should be tolerant of such failures and degrade gracefully with resource unavailability.

The context within which service compositions are enacted also offers research opportunities, in order to achieve customized and personalized behavior [75]. The mobile users' preferences, behavior and context vary among users and their situations. Mechanisms for the dynamic composition of services are necessary to provide tailored services on demand to users [119]. As it deals with dynamic adaptation, the middleware needs to map the high level user requirements to the low level hardware resource efficiently.

Awareness & Adaptability. In an ultra large scale scenario where service compositions may consist of a vast number of participating services, being aware of and dealing with failures becomes extremely challenging and difficult. Specifically, it is not clear how to automate the testing of service compositions [26]. When the compositions are viewed as atomic services, the testing approaches are derived from the classical unit testing techniques [14]. On the other hand, such approaches are not directly applicable in services participating in a composition. To scale up existing efforts, several issues must be resolved. First, it is necessary to deal with the lack of observability; since some services export only their interfaces, this prevents white-box testing in some cases. Some inherent characteristics of service compositions such as dynamics, adaptiveness, third-party rules, and governance issues must also be solved to automate the integration tests. Finally, some issues such as the decentralized flow of information, multiple party communication, and parallelism must be adequately con-

SOM functionalities	State of the art
Description	Web services, Semantic Web, OWL
Discovery	Service registries and distributed, hybrid service discovery protocols
Access	ESB paradigm for heterogeneous SOC technology/middleware integration
Composition	Orchestration/Choreography-based composition of services and related BPEL engines, Dynamic composition and adaptation

Table 3 SOM State of the Art

sidered for the automated testing of ultra large scale service compositions.

Security, Privacy & Trust. As already discussed, achieving security, privacy and trust in the Future Internet becomes much harder and challenging. Dealing with these issues mainly concerns the underlying middleware facilities that enable access to the services that participate in service compositions (see Section 5). However, the interesting issue is that these facilities should be aware of and take into consideration specific characteristics of the compositions themselves, which is not straightforward in the Future Internet ultra large scale setting. For instance, if a service composition consists of an ultra large number of choreographed services, then the participating services cannot be aware of the actual sources of information that they receive. The previous may be a feature that could be exploited to achieve privacy. On the other hand, the same characteristic may be a problem from the perspective of security and trust.

In summary, a service-oriented middleware for the Future Internet should deal with new tools, strategies, and languages for modeling, executing, and automatically composing service orchestration and choreographies, dealing with scalability, heterogeneity, mobility, awareness, adaptability, and safety issues. In particular, some research is expected to improve huge compositions visualization (when dealing with modeling or semi-automatic adaptation), orchestrations and choreographies evolution, integration with Cloud computing technology, ontology use in context-aware service compositions, coordination strategies, automated testing, and privacy and dissemination of information.

7 Conclusion

Our survey of Service-Oriented Middleware, some highlights of which are depicted in Table 3, showed a ma-

SOM functionalities	Research directions
Description	Investigate trade-off between rich service descriptions and related processing complexity.
Discovery	Investigate service discovery protocols for the ultra large scale, heterogeneous and mobile Future Internet, while controlling the quality of the information and providing security, privacy, trust guarantees.
Access	Exploit high-performance, resource-on-demand computing technologies to cope with scale. Handle heterogeneity, mobility, security, privacy, trust in open, dynamic and aware settings.
Composition	Enable scalable and adaptive choreography modeling and execution for the highly heterogeneous and mobile Future Internet, while guaranteeing security and privacy properties.

Table 4 Research Directions for SOM in the Future Internet

tured paradigm, well anchored in the present Internet. On the other hand, we have pointed out that the Future Internet, perceived as the evolution of the current Internet, has to face existing challenges (i.e., related to scalability, heterogeneity, mobility, awareness & adaptability, and security, privacy & trust), while pushed to the extreme, as summarized in Table 2. Indeed, the Internet of Services and Things is becoming a reality with a population of services, including service-enabled things, bound to evolve at a very fast pace. Then, assisting the developers in leveraging such a plethora of services to provide new applications raises the need to carefully revisit the service-oriented middleware solutions developed for today's Internet (see Table 4).

Overall, the middleware must cope with the trade-off between the increasing scale and heterogeneity of the Future Internet reflected on the information exposed by service descriptions and the complexity of processing this information, to enable service publication, discovery, composition and access. With respect to service publication and discovery, the correct architectural choices must be made for the corresponding publication and discovery protocols, to balance the trade-off introduced among keeping the complexity of the protocols low, handling the different dimensions of the Future Internet scale/heterogeneity/mobility, controlling the quality of the information managed by the protocols, and providing reasonable security/privacy/trust guarantees. From the standpoint of service access, the middleware must benefit as much as possible from high-performance computing paradigms and

resource-on-demand computing technologies, so as to cope with the different dimensions of the Future Internet scale. The challenges related to heterogeneity, mobility, security, privacy and trust in service access are intensified by the open, dynamic and aware character of the Future Internet. For instance, the middleware should be able to support unanticipated interaction patterns and *ad hoc* security/privacy/trust needs. Finally, the middleware must successfully handle the inherent complexity of the dynamic service composition and adaptation strategies, exacerbated by the scale of choreographies, the mobility of participants, and their highly varying characteristics and requirements, e.g., in terms of security, privacy and trust.

References

1. Abi-Char, P., Mokhtari, M., Mhamed, A., El-Hassan, B.: A dynamic trust-based context-aware authentication framework with privacy preserving. International Journal of Computer and Network Security **2**(2) (2010)
2. AKARI Architecture Design Project: New generation network architecture: AKARI conceptual design (ver1.1). Tech. rep. (2008). Available at http://nag.nict.go.jp/topics/AKARI_fulltext_e_translated_version_1_1.pdf
3. Al-Masri, E., Mahmoud, Q.H.: Investigating Web services on the World Wide Web. In: 17th International Conference on World Wide Web (WWW) (2008)
4. Andersen, F.U., Berndt, H., Abramowicz, H., Tafazolli, R.: Future internet from mobile and wireless requirements perspective. http://www.emobility.eu.org/ (2007)
5. Arango, M., Kaponig, B.: Ultra-scalable architectures for telecommunications and Web 2.0 services. In: 13th International Conference on Intelligence in Next Generation Networks, (ICIN) (2009)
6. Ardagna, D., Comuzzi, M., Mussi, E., Pernici, B., Plebani, P.: PAWS: A framework for executing adaptive web-service processes. IEEE Software **24**(6) (2007)
7. Ardagna, D., Pernici, B.: Adaptive service composition in flexible processes. IEEE Transactions on Software Engineering **33**(6) (2007)
8. Athanasopoulos, D., Zarras, A., Issarny, V.: Service substitution revisited. In: 24th IEEE/ACM International Conference on Automated Software Engineering, (ASE) (2009)

9. Athanasopoulos, D., Zarras, A., Issarny, V.: Towards the maintenance of service oriented software. In: 3rd CSMR Workshop on Software Quality and Maintenance, (SQM) (2009)

10. Athanasopoulos, D., Zarras, A.V., Issarny, V., Pitoura, E., Vassiliadis, P.: Cowsami: Interface-aware context gathering in ambient intelligence environments. Pervasive and Mobile Computing **4**(3) (2008)

11. Avilés-López, E., García-Macías, J.: TinySOA: a service-oriented architecture for wireless sensor networks. Service Oriented Computing and Applications **3**(2) (2009)

12. Badr, N., Taleb-Bendiab, A., Randles, M., Reilly, D.: A Deliberative Model for Self-Adaptation Middleware Using Architectural Dependency. In: 15th International Workshop on Database and Expert Systems Applications, (DEXA) (2004)

13. Barros, A., Dumas, M., Oaks, P.: Standards for Web Service Choreography and Orchestration: Status and Perspectives. In: Business Process Management Workshops (2006)

14. Bartolini, C., Bertolino, A., Elbaum, S.G., Marchetti, E.: Whitening SOA testing. In: 7th The joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE) (2009)

15. Bassi, A., Europe, H., Horn: Internet of Things in 2020: Roadmap for the Future. Tech. rep., Internet of Things workshop report (2008). Available at http://www.iot-visitthefuture.eu/fileadmin/documents/researchforeurope/270808_IoT_in_2020_Workshop_Report_V1-1.pdf

16. Baude, F., Filali, I., Huet, F., Legrand, V., Mathias, E., Merle, P., Ruz, C., Krummenacher, R., Simperl, E., Hammerling, C., Lorre, J.P.: ESB federation for large-scale SOA. In: ACM Symposium on Applied Computing (SAC) (2010)

17. Ben Mokhtar, S., Georgantas, N., Issarny, V.: CO-COA: Conversation-based service composition in pervasive computing environments with QoS support. Journal of Systems and Software **80** (2007)

18. Ben Mokhtar, S., Kaul, A., Georgantas, N., Issarny, V.: Efficient semantic service discovery in pervasive computing environments. In: ACM/IFIP/USENIX International Conference on Middleware (2006)

19. Ben Mokhtar, S., Preuveneers, D., Georgantas, N., Issarny, V., Berbers, Y.: EASY: Efficient semantic service discovery in pervasive computing environments with QoS and context support. Journal of Systems and Software **81** (2008)

20. Ben Mokhtar, S., Raverdy, P.G., Urbieta, A., Cardoso, R.S.: Interoperable semantic and syntactic service discovery for ambient computing environments. International Journal of Ambient Computing and Intelligence **2**(4) (2010)

21. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data - The Story So Far. International Journal on Semantic Web and Information Systems, (IJSWIS) (2009)

22. Brogi, A., Popescu, R.: Automated generation of BPEL adapters. In: International Conference on Service Oriented Computing, (ICSOC) (2006)

23. Bromberg, Y.D., Issarny, V.: INDISS: Interoperable discovery system for networked services. In: ACM/IFIP/USENIX International Conference on Middleware (2005)

24. Calvanese, D., Giacomo, G.D., Lenzerini, M., Mecella, M., Patrizi, F.: Automatic Service Composition and Synthesis: the Roman Model. IEEE Data Engineering Bulletin **31**(3) (2008)

25. Campbell, R., Gupta, I., Heath, M., Ko, S.Y., Kozuch, M., Kunze, M., Kwan, T., Lai, K., Lee, H.Y., Lyons, M., Milojicic, D., O'Hallaron, D., Soh, Y.C.: Open CirrusTMcloud computing testbed: federated data centers for open source systems and services research. In: Conference on Hot topics in cloud computing, (HotCloud) (2009)

26. Canfora, G., Penta, M.D.: Service-oriented architectures testing: A survey. Software Engineering (2009)

27. Canfora, G., Penta, M.D., Esposito, R., Villani, M.L.: A framework for QoS-aware binding and re-binding of composite Web services. Journal of Systems and Software **81**(10) (2008)

28. Caporuscio, M., Raverdy, P.G., Issarny, V.: ubiSOAP: A service oriented middleware for ubiquitous networking. IEEE Transactions on Services Computing (PrePrints) (2010)

29. Cardellini, V., Casalicchio, E., Grassi, V., Presti, F.L., Mirandola, R.: QoS-driven runtime adaptation of service oriented architectures. In: 7th joint European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE) (2009)

30. Cardellini, V., Iannucci, S.: Designing a broker for QoS-driven runtime adaptation of SOA applications. In: IEEE International Conference on Web Services (ICWS) (2010)

31. Cardoso, R., Raverdy, P.G., Issarny, V.: A privacy-aware service discovery middleware for pervasive environments. In: IFIP Conference on Trust Management, (IFIPTM), *LNCS*, vol. 238 (2007)

32. Cardoso, R.S., Issarny, V.: Architecting pervasive computing systems for privacy: A survey. In: 6th Working IEEE / IFIP Conference on Software Architecture, (WICSA) (2007)

33. CASAGRAS: RFID and the inclusive model for the internet of things. http://www.rfidglobal.eu (2009)

34. Cavallaro, L., Nitto, E.D., Furia, C.A., Pradella, M.: A tile-based approach for self-assembling service compositions. In: IEEE International Conference on Engineering of Complex Computer Systems, (ICECCS) (2010)

35. Cervantes, H., Hall, R.: Automating service dependency management in a service-oriented component model. In: 6th ICSE Component-Based Software Engineering Workshop (CBSE) (2003)

36. Chakraborty, D., Joshi, A., Finin, T., Yesha, Y.: Service composition for mobile environments. Mobile Networks and Applications **10** (2005)

37. Chakraborty, D., Joshi, A., Yesha, Y., Finin, T.: Toward distributed service discovery in pervasive computing environments. IEEE Transactions on Mobile Computing **5**(2) (2006)

38. Chao, C., Zongyan, Q.: An approach to check choreography with channel passing in WS-CDL. In: IEEE International Conference on Web Services (ICWS) (2008)

39. Chappell, D.: Enterprise Service Bus. O'Reilly Media, Inc. (2004)

40. Charlet, D., Issarny, V., Chibout, R.: Energy-efficient middleware-layer multi-radio networking: An assessment in the area of service discovery. Computer Networks **52** (2008)

41. Chuang, S.N., Chan, A.: Dynamic QoS adaptation for mobile middleware. IEEE Transactions on Software Engineering **34**(6) (2008)

42. Cirne, W., Brasileiro, F., Andrade, N., Costa, L., Andrade, A., Novaes, R., Mowbray, M.: Labs of the world, unite!!! Journal of Grid Computing **4** (2006)
43. Cisco: Cisco Visual Networking Index: Forecast and Methodology, 2009–2014. Tech. rep. (2010). Available at http://www.cisco.com/en/US/solutions/collateral/-ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf
44. Cisco: Cisco visual networking index: Global mobile data traffic forecast update, 2010–2015. Tech. rep. (2011). Available at http://www.cisco.com/en/US/-solutions/collateral/ns341/ns525/ns537/ns705/ns827/-white_paper_c11-520862.html
45. Costa, P., Coulson, G., Gold, R., Lad, M., Mascolo, C., Mottola, L., Picco, G.P., Sivaharan, T., Weerasinghe, N., Zachariadis, S.: The RUNES middleware for networked embedded systems and its application in a disaster management scenario. In: 5th IEEE International Conference on Pervasive Computing and Communications, (PerCom) (2007)
46. Costa, P., Mottola, L., Murphy, A.L., Picco, G.P.: Programming wireless sensor networks with the teenylime middleware. In: ACM/IFIP/USENIX 2007 International Conference on Middleware (2007)
47. Cotroneo, D., Graziano, A., Russo, S.: Security requirements in service oriented architectures for ubiquitous computing. In: 2nd Workshop on Middleware for Pervasive and Ad-Hoc Computing, (MPAC) (2004)
48. Daras, P., Williams, D., Guerrero, C., Kegel, I., Laso, I., Bouwen, J., Meunier, J., Niebert, N., Zahariadis, T.: Why do we need a content-centric future Internet? proposals towards content-centric Internet architectures. Information Society and Media Journal (2009)
49. Decker, G., Kopp, O., Leymann, F., Weske, M.: BPEL4Chor: Extending BPEL for Modeling Choreographies. In: International Conference on Web Services (ICWS) (2007)
50. Dijkman, R., Dumas, M.: Service-oriented design: A multi-viewpoint approach. International Journal of Co-operative Information Systems **13**(4) (2004)
51. Dustdar, S., Treiber, M.: A view based analysis on Web service registries. Distributed and Parallel Databases **18**(2) (2005)
52. Ec FIArch Group: Fundamental limitations of current Internet and the path to future Internet. Tech. rep. (2011). Available at http://ec.europa.eu/information_society/activities/foi/library/docs/fiarch-current-internet-limitations-march2011.pdf
53. Eisenman, S.B., Miluzzo, E., Lane, N.D., Peterson, R.A., Ahn, G.S., Campbell, A.T.: The bikenet mobile sensing system for cyclist experience mapping. In: 5th International Conference on Embedded Networked Sensor Systems (SenSys) (2007)
54. Elgazzar, K., Hassan, A.E., Martin, P.: Clustering WSDL documents to bootstrap the discovery of Web services. In: IEEE International Conference on Web Services (ICWS) (2010)
55. Fielding, R.T., Taylor, R.N.: Principled design of the modern Web architecture. ACM Transactions on Internet Technology **2** (2002)
56. Fluegge, M., Tourtchaninova, D.: Ontology-derived activity components for composing travel Web services. In: The International Workshop on Semantic Web Technologies in Electronic Business, (SWEB) (2004)
57. Foster, H., Uchitel, S., Magee, J., Kramer, J.: LTSA-WS: a tool for model-based verification of Web service compositions and choreography. In: 28th International Conference on Software Engineering, (ICSE) (2006)
58. Foster, H., Uchitel, S., Magee, J., Kramer, J.: Model-based analysis of obligations in Web service choreography. In: Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services, (AICT-ICIW) (2006)
59. Gao, Z., Wang, L., Yang, M., Yang, X.: CNPGSDP: an efficient group-based service discovery protocol for MANETs. Computer Networks **50**(16) (2006)
60. Gao, Z., Wang, L., Yang, X., Wen, D.: PCPGSD: an enhanced GSD service discovery protocol for MANETs. Computer Communications **29** (2006)
61. Gao, Z., Yang, X., Ma, T., Cai, S.: RICFFP: An efficient service discovery protocol for MANETs. In: Embedded and Ubiquitous Computing, *LNCS*, vol. 3207 (2004)
62. Grace, P., Blair, G., Samuel, S.: ReMMoC: A reflective middleware to support mobile client interoperability. In: On The Move to Meaningful Internet Systems, *LNCS*, vol. 2888 (2003)
63. Grace, P., Blair, G.S., Samuel, S.: A reflective framework for discovery and interaction in heterogeneous mobile environments. SIGMOBILE Mobile Computing and Communications Review **9**(1) (2005)
64. Gu, T., Pung, H.K., Zhang, D.Q.: A service-oriented middleware for building context-aware services. Journal of Network and Computer Applications **28** (2005)
65. Hofreiter, B., Huemer, C.: A model-driven top-down approach to inter-organizational systems: From global choreography models to executable BPEL. In: IEEE Joint Conference on E-Commerce Technology (CEC) and Enterprise Computing, E-Commerce, and E-Services (EEE) (2008)
66. Hofreiter, B., Huemer, C., Liegl, P., Schuster, R., Zapletal., M.: UN/CEFACT'S Modeling Methodology (UMM): A UML profile for B2B e-commerce. In: Advances in Conceptual Modeling - Theory and Practice ER Workshops, LNCS 4231 (2006)
67. Huang, Y., Xu, C., Wang, H., Xia, Y., Zhu, J., Zhu, C.: Formalizing Web service choreography interface. In: 21st International Conference on Advanced Information Networking and Applications Workshops, (AINAW) (2007)
68. Ingram, D.: Reconfigurable middleware for high availability sensor systems. In: 3rd ACM International Conference on Distributed Event-Based Systems, (DEBS) (2009)
69. Josuttis, N.: SOA in Practice: The Art of Distributed System Design (2007)
70. Julien, C., Roman, G.C.: Egospaces: facilitating rapid development of context-aware mobile applications. IEEE Transactions on Software Engineering **32**(5) (2006)
71. Kalasapur, S., Kumar, M., Shirazi, B.: Dynamic service composition in pervasive computing. IEEE Transactions on Parallel and Distributed Systems **18**(7) (2007)
72. Kang, Z., Wang, H., Hung, P.: WS-CDL+ for Web service collaboration. Information Systems Frontiers **9**(4) (2007)
73. Kapadia, A., Henderson, T., Fielding, J.J., Kotz, D.: Virtual walls: protecting digital privacy in pervasive environments. In: PERVASIVE (2007)
74. Kassim, A., Esfandiari, B., Majumdar, S., Serghi, L.: A flexible hybrid architecture for management of distributed Web service registries. In: 5th Annual Conference on Communication Networks and Services Research (CNSR) (2007)

75. Keidl, M., Kemper, A.: Towards context-aware adaptable Web services. In: 13th international World Wide Web Conference (WWW) (2004)
76. Kozat, U.C., Tassiulas, L.: Service discovery in mobile ad hoc networks: An overall perspective on architectural choices and network layer support issues. Ad Hoc Networks **2**(1) (2004)
77. Lanese, I., Guidi, C., Montesi, F., Zavattaro, G.: Bridging the gap between interaction- and process-oriented choreographies. In: 6th IEEE international Conference on Software Engineering and Formal Methods (SEFM) (2008)
78. Li, J., He, J., Zhu, H., Pu, G.: Modeling and verifying Web services choreography using process algebra. In: 31st IEEE Software Engineering Workshop (SEW) (2007)
79. Li, R., Zhang, Z., Song, W., Ke, F., Lu, Z.: Service publishing and discovering model in a Web services oriented peer-to-peer system. In: International Conference on Web Engineering (ICWE), LNCS 3579 (2005)
80. Liang, Q.A., Lam, H.: Web service matching by ontology instance categorization. In: IEEE International Conference on Services Computing, (SCC) (2008)
81. Lim, H., Kim, C.: Flooding in wireless ad hoc networks. Computer Communications **24**(3-4) (2001)
82. Lu, H., Pan, W., Lane, N.D., Choudhury, T., Campbell, A.T.: SoundSense: scalable sound sensing for people-centric applications on mobile phones. In: MobiSys (2009)
83. Luo, J., Montrose, B.E., Kim, A., Khashnobish, A., Kang, M.H.: Adding OWL-S support to the existing UDDI infrastructure. In: IEEE International Conference on Web Services (ICWS) (2006)
84. Mabrouk, N.B., Beauche, S., Kuznetsova, E., Georgantas, N., Issarny, V.: QoS-aware service composition in dynamic service oriented environments. In: ACM/IFIP/USENIX International Conference on Middleware (2009)
85. Mabrouk, N.B., Georgantas, N., Issarny, V.: A semantic end-to-end QoS model for dynamic service oriented environments. In: ICSE Workshop on Principles of Engineering Service Oriented Systems (PESOS) (2009)
86. Madden, S.R., Franklin, M.J., Hellerstein, J.M., Hong, W.: TinyDB: an acquisitional query processing system for sensor networks. ACM Transactions on Database Systems **30** (2005)
87. Mamei, M., Zambonelli, F.: Programming pervasive and mobile computing applications: The TOTA approach. ACM Transactions on Software Engineering and Methodologies **18** (2009)
88. Marconi, A., Pistore, M., Traverso, P.: Automated Composition of Web Services: the ASTRO Approach. IEEE Data Engineering Bulletin **31**(3) (2008)
89. Melliti, T., Poizat, P., Mokhtar, S.B.: Distributed behavioural adaptation for the automatic composition of semantic services. In: Fundamental Approaches in Software Engineering (FASE), LNCS 4961 (2008)
90. Mens, T., Demeyer, S.: Software Evolution. Springer (2008)
91. Mohan, A., Blough, D.M.: AttributeTrust: A framework for evaluating trust in aggregated attributes via a reputation system. In: 6th Annual Conference on Privacy, Security and Trust (PST) (2008)
92. Mordinyi, R., Kühn, E., Schatten, A.: Space-based architectures as abstraction layer for distributed business applications. In: International Conference on Complex, Intelligent and Software Intensive Systems (CISIS) (2010)
93. Nakazawa, J., Tokuda, H., Edwards, W., Ramachandran, U.: A bridging framework for universal interoperability in pervasive systems. In: IEEE International Conference on Distributed Computing Systems (ICDCS) (2006)
94. Nayak, R., Lee, B.: Web service discovery with additional semantics and clustering. In: IEEE / WIC / ACM International Conference on Web Intelligence, (WI) (2007)
95. Niam, A.N., Baldoni, R., Beraldi, R.: A survey of service discovery protocols in multihop mobile ad hoc environments. IEEE Pervasive Computing **8**(1) (2009)
96. Nidd, M.: Service discovery in DEAPspace. IEEE Personal Communications **8**(4) (2001)
97. Oldham, N., Verma, K., Sheth, A., Hakimpour, F.: Semantic WS-agreement partner selection. In: 15th international conference on World Wide Web (WWW) (2006)
98. Ortiz Jr., S.: Getting on board the enterprise service bus. IEEE Computer **40** (2007)
99. Paolucci, M., Kawamura, T., Payne, T.R., Sycara, K.P.: Importing the semantic web in UDDI. In: Revised Papers for International Workshop on Web Services, E-Business, and the Semantic Web (WES 2002), in conj. with CAiSE 2002, LNCS 2512 (2002)
100. Paolucci, M., Kawamura, T., Payne, T.R., Sycara, K.P.: Semantic matching of Web services capabilities. In: 1st International Semantic Web Conference on The Semantic Web (ISWC) (2002)
101. Papadimitriou, D.: Future Internet–The Cross-ETP Vision Document. European Technology Platform, Alcatel Lucent **8** (2009)
102. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: Service-oriented computing: State of the art and research challenges. IEEE Computer **40**(11) (2007)
103. Pathak, J., Basu, S., Lutz, R., Honavar, V.: MOSCOE: an approach for composing Web services through iterative reformulation of functional specifications. International Journal on Artificial Intelligence Tools **17**(1) (2008)
104. Pautasso, C., Zimmermann, O., Leymann, F.: Restful Web services vs. "big"' Web services: making the right architectural decision. In: 17th international conference on World Wide Web (WWW) (2008)
105. Pereira, J.: From autonomous to cooperative distributed monitoring and control: Towards the Internet of smart things. In: ERCIM Workshop on eMobility (2008)
106. Ponnekanti, S., Fox, A.: SWORD: A Developer Toolkit for Web Service Composition. In: 11th International World Wide Web Conference (WWW) (2002)
107. Qu, Y., Hu, W., Cheng, G.: Constructing virtual documents for ontology matching. In: 15th international conference on World Wide Web (WWW) (2006)
108. Rambold, M., Kasinger, H., Lautenbacher, F., Bauer, B.: Towards autonomic service discovery: A survey and comparison. In: IEEE International Conference on Services Computing (SCC) (2009)
109. Randic, M., Blaskovic, B., Knezevic, P.: Modeling service dependencies in ad hoc collaborative systems. In: EUROCON (2005)
110. Rao, J., Su, X.: A survey of automated Web service composition methods. In: Semantic Web Services and Web Process Composition, *LNCS*, vol. 3387 (2005)
111. Raverdy, P.G., Issarny, V., Chibout, R., de La Chapelle, A.: A multi-protocol approach to service discovery and access in pervasive environments. In: 3rd Annual International Conference on Networking Services, Mobile and Ubiquitous Systems (2006)

112. Rellermeyer, J.S., Riva, O., Alonso, G.: AlfredO: an architecture for flexible interaction with electronic devices. In: 9th ACM/IFIP/USENIX International Conference on Middleware (2008)

113. Ren, K., Lou, W., Kim, K., Deng, R.: A novel privacy preserving authentication and access control scheme for pervasive computing environments. IEEE Transactions on Vehicular Technology **55**(4) (2006)

114. Ries, S.: Certain trust: a trust model for users and agents. In: ACM Symposium on Applied Computing (SAC) (2007)

115. Rosenberg, F., Enzi, C., Michlmayr, A., Platzer, C., Dustdar, S.: Integrating quality of service aspects in top-down business process development using WS-CDL and WS-BPEL. In: 11th IEEE international Enterprise Distributed Object Computing Conference (EDOC) (2007)

116. Sailhan, F., Issarny, V.: Scalable service discovery for MANET. In: 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom) (2005)

117. Salaun, G.: Generation of Service Wrapper Protocols from Choreography Specifications. In: 6th IEEE International Conference on Software Engineering and Formal Methods (SEFM) (2008)

118. van Seghbroeck, G., de Turck, F., Dhoedt, B., Demeester, P.: Web service choreography conformance verification in M2M systems through the piX-model. In: IEEE International Conference on Pervasive Services (ICPS) (2007)

119. Goncalves da Silva, E., Ferreira Pires, L., van Sinderen, M.: Dynamic composition of services: Why, where and how. In: 2nd International Workshop on Enterprise Systems and Technology (I-WEST) (2008)

120. da Silva e Silva, F.J., Kon, F., Goldman, A., Finger, M., de Camargo, R.Y., Filho, F.C., Costa, F.M.: Application execution management on the InteGrade opportunistic grid middleware. Journal of Parallel and Distributed Computing **70** (2010)

121. Sirin, E., Hendler, J., Parsia, B.: Semi-automatic composition of Web services using semantic descriptions. In: Web Services: Modeling, Architecture and Infrastructure workshop in conjunction with ICEIS (WSMAI) (2003)

122. Sivashanmugam, K., Verma, K., Sheth, A.: Discovery of Web services in a federated registry environment. In: IEEE International Conference on Web Services (ICWS) (2004)

123. Sivashanmugam, K., Verma, K., Sheth, A.P., Miller, J.A.: Adding semantics to Web services standards. In: IEEE International Conference on Web Services (ICWS) (2003)

124. Song, H., Cheng, D., Messer, A., Kalasapur, S.: Web service discovery using general-purpose search engines. In: IEEE International Conference on Web Services (ICWS) (2007)

125. Spanoudakis, G., Sisman, A.: Discovering services during service-based system design using UML. IEEE Transactions on Software Engineering **36**(3) (2010)

126. Squicciarini, A., Bertino, E., Ferrari, E., Paci, F., Thuraisingham, B.: PP-trust-X: A system for privacy preserving trust negotiations. ACM Transactions on Information and System Security **10** (2007)

127. Srivastava, B., Koehler, J.: Web Service Composition - Current Solutions and Open Problems. International Conference on Automated Planning and Scheduling, (ICAPS) (2003)

128. Stuckmann, P., Zimmermann, R.: European research on future Internet design. IEEE Wireless Communications **16**(5) (2009)

129. Su, J., Bultan, T., Fu, X., Zhao, X.: Towards a theory of Web service choreographies. In: 4th International Conference on Web Services and Formal Methods (SEFM) (2007)

130. Taherkordi, A., Le-Trung, Q., Rouvoy, R., Eliassen, F.: WiSeKit: A distributed middleware to support application-level adaptation in sensor networks. In: Distributed Applications and Interoperable Systems, *LNCS*, vol. 5523 (2009)

131. Takabi, H., Amini, M., Jalili, R.: Enhancing role-based access control model through fuzzy relations. In: 3rd International Symposium on Information Assurance and Security (IAS) (2007)

132. Toma, I., Simperl, E., Filipowska, A., Hench, G., Domingue, J.: Semantics-driven interoperability on the Future Internet. In: IEEE International Conference on Semantic Computing (ICSC) (2009)

133. Tselentis, G., Galis, A., Gavras, A., Krco, S., Lotz, V., Simperl, E., Stiller, B., Zahariadis, T.: Towards the Future Internet - emerging trends from european research. IOS Press (2010)

134. Uddin, M.G., Zulkernine, M., Ahamed, S.I.: CAT: a context-aware trust model for open and dynamic systems. In: ACM Symposium on Applied Computing (SAC) (2008)

135. Ververidis, C.N., Polyzos, G.C.: Service discovery for mobile ad hoc networks: a survey of issues and techniques. IEEE Communications Surveys **10**(3) (2008)

136. Wieczorek, S., Roth, A., Stefanescu, A., Charfi, A.: Precise steps for choreography modeling for SOA validation and verification. In: 2008 IEEE International Symposium on Service-Oriented System Engineering (SOSE) (2008)

137. Xu, W., Xin, Y., Lu, G.: A trust framework for pervasive computing environments. In: International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM) (2007)

138. Yang, H., Zhao, X., Cai, C., Qiu, Z.: Model-checking of Web services choreography. In: IEEE International Symposium on Service-Oriented System Engineering (SOSE) (2008)

139. Yeung, W.: A formal basis for cross-checking ebXML BPSS Choreography and Web service orchestration. In: IEEE Asia-Pacific Services Computing Conference (APSCC) (2008)

140. Zachariadis, S., Mascolo, C., Emmerich, W.: The SATIN component system: A metamodel for engineering adaptable mobile systems. IEEE Transactions on Software Engineering **32** (2006)

141. Zachos, K., Maiden, N.: Inventing requirements from software: An empirical investigation with Web services. In: 16th IEEE International Requirements Engineering Conference (RE) (2008)

142. Zaha, J.M., Barros, A., Dumas, M., terHofstede, A.: Let's dance: A language for service behavior modeling. In: 14th International Conference on Cooperative Information Systems (CoopIS) (2006)

143. Zahariadis, T., Pau, G., Celetto, L., Daras, P.: Seamless content delivery in the future mobile internet. IEEE Wireless Communications (2009)

144. Zeng, L., Benatallah, B., Ngu, A., Dumas, M., Kalagnanam, J., Chang, H.: Qos-aware middleware for Web services composition. IEEE Transactions on Software Engineering **30**(5) (2004)

145. Zhai, Y., Zhang, J., Lin, K.J.: SOA middleware support for service process reconfiguration with end-to-end QoS constraints. In: IEEE International Conference on Web Services (ICWS) (2009)
146. Zhang, C., Krishnamurthy, A., Wang, R.Y., Singh, J.P.: Combining flexibility and scalability in a peer-to-peer publish/subscribe system. In: ACM/IFIP/USENIX International Conference on Middleware (2005)
147. Zhang, P., Li, B., Muccini, H., Zhou, Y., Sun, M.: Data-enriched modeling and verification of WS-CDL based on UML models. In: IEEE International Conference on Web Services (ICWS) (2008)
148. Zhang, Q., Cheng, L., Boutaba, R.: Cloud computing: state of the art and research challenges. Journal of Internet Services and Applications **1**(1) (2010)
149. Zhang, S., Zhang, S., Chen, X., Huo, X.: Cloud computing research and development trend. In: Second International Conference on Future Networks (2010)
150. Zhou, J., Pakkala, D., Perala, J., Niemela, E., Riekki, J., Ylianttila, M.: Dependency-aware service oriented architecture and service composition. In: IEEE International Conference on Web Services (ICWS) (2007)