2012 Service Research and Innovation Institute Global Conference

Emerging Research Themes in Services-oriented Systems

Luciano Baresi[⊙], Nikolaos Georgantas[◆], Kristof Hamann[■], Valerie Issarny[◆],
Winfried Lamersdorf[■], Andreas Metzger[○], and Barbara Pernici[⊙]

[⊙] Politecnico di Milano, Dipartimento di Elettronica e Informazione, Milano (Italy)
[◆] INRIA Paris-Rocquencourt (France)
[■] University of Hamburg, Department of Informatics, Hamburg (Germany)
[○] University of Duisburg-Essen, Software Systems Engineering, Essen (Germany)

baresi@elet.polimi.it

Service-oriented Architecture (SOA) is one of the most recent trends in IT solutions. From a technical perspective SOA, can be considered a method for designing and developing IT systems where applications are constructed from loosely coupled and autonomous building blocks. For a thorough analysis and discussion of SOA at large, and of the applications it enables, IFIP (International Federation of Information Processing) has recently established a new working group on Services-oriented Systems (WG 2.14/6.12/8.10). The paper presents some first results of the analysis aimed to identify the relevant trends and shape emerging research themes. The goal is to highlight the peculiarities of each area, the main opportunities, and also some possible synergies with the others.

Keywords: services and distributed systems, service engineering, process monitoring and management, service science

I. Introduction

Service-oriented Architecture (SOA) is one of the most recent trends in IT solutions ---and it is also a promising technical idea. Among the many definitions of SOA, this: *"a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains ... in SOA, services are the mechanism by which needs and capabilities are brought together"* [29] distinguishes SOA from previous development approaches. From a technical perspective, SOA can be seen as a method for designing and developing IT systems where applications are constructed from loosely coupled and autonomous building blocks that communicate through well-defined interfaces. From a business-oriented perspective, SOA seems to promise as a new way of solving the well-known alignment problem between business (needs) and IT (capabilities).

As a design technique, the idea of service orientation is far from being new, but the rapid adoption of network technologies has made SOA an interesting alternative to existing development methods. The essence of SOA must be found in the *everything-as-a-service* view of the world. Every problem can be solved by one service or by a collection of services, each with its own peculiar capabilities, provided through well-defined interfaces. The goal is to design a loosely coupled system that use services supplied by independent providers. Services have introduced a prominent and structural change in the software history by providing the basics for the rapid and easy development of distributed and heterogeneous applications. Services-oriented systems promise a world where heterogeneous application components are assembled with almost no effort into compositions that create flexible and dynamic software systems.

Services do not work solely at application level anymore, but they are also used to characterize ---and abstract from unnecessary details--- the actual offer of fully distributed software platforms and computing and communication infrastructures.

For a thorough analysis and discussion of SOA at large, and of the applications it enables, IFIP (the International Federation of Information Processing) has recently established a new working group on *Services-oriented Systems* (WG 2.14/6.12/8.10), as a joint effort of its technical committees TC2 (Software: Theory and Practice), TC6 (Communication Systems), and TC8 (Information Systems). This involvement of different technical committees is motivated by the multi-dimensional nature of services-oriented systems. Indeed, a major goal of the new working group is to organize and promote the exchange of information on both fundamental and practical aspects of SOA. This is to contribute to the so-called service science, management, and engineering (SSME), which addresses the problem from a multi-faceted perspective. Given the many existing initiatives related to services and service-based systems, the working group will try to increase the synergies among them and to structure a research community that comprises both academia and industry; the long term goal is to create an active, permanent, and international forum on SOA.

In this context, one of the first activities of the group is the identification of relevant trends and emerging research themes. To this end, the paper presents some first considerations to identify: (1) the main current research topics addressed by academia and industry, (2) the emerging research topics that will lead to solutions in the next 10 years, and also (3) those topics that are already mature, but will not require further investigations. The findings are organized around four main topics: everything as a service, SOA in mobile settings, services and information systems, and service-based distributed systems.

The rest of the paper is organized around the four main topics. Section II discusses the research challenges as for the "everything as a service" view. Section III introduces the issues related to mobile services. Section IV is about the im-

978-0-7695-4770-1/12 $26.00 © 2012 IEEE
DOI 10.1109/SRII.2012.44

pact of services on information systems, while Section V deals with the relationship between SOA and distributed systems. Section IV concludes the paper.

II. "EVERYTHING" AS A SERVICE

The Internet by and for People, the Internet of Contents, the Internet of Services and the Internet of Things are the pillars of the Future Internet[1], and they all exploit the service paradigm as common abstraction means. Software components, things, contents, and humans are all abstracted as if they were services, and thus they foster the "everything" as a service idea.

Besides the "usual" software services, we must consider that people play a key, significant role in many situations and their contribution can be diverse. We refer to the idea of *humans in the loop* when some experts are supposed to oversee and control a mostly software computation. We use the term *crowdsourcing* when significant parts of the computations are performed by humans, and not by computers. Moreover, people and software components are often embedded in rich environments, where the computing power is spread around a number of (small) devices, and smartphones, sensors, actuators, and tags can be seen as live entities from which the system can acquire fresh data and can provide directives to control the environment. Similarly, very diverse contents (videos, texts, pictures) are often abstracted into the services through which they can be enjoyed.

The problem is not the composition of purely software services anymore. The future (present) is an eco-system of heterogeneous services the users (developers, providers, or final users) can exploit to run their business, live their lives, and create value. Even if the *basic concepts* are always the same, the wider the landscape we consider is, the more sophisticated and articulated proposed solutions must be. In the meanwhile, some (new) problems become prominent, while others are de-emphasized.

A. Current Research Topics

Homogenous and extended service models: More complete and extended models become mandatory when one wants to integrate different types of services and be able to describe their specific features. In these years, there have been different service models (e.g., the one proposed by W3C [7]), but different domains ---or different abstraction levels--- remain separate. Many proposals have been created around the well-known Web services standards, but many of them have simply tried to mimic some concepts in different domains. For example, DPWS (Device Profile for Web Services [16]) is an attempt to see devices as if they were web services, without considering their key characteristics (e.g., the real-time behavior or limited amount of resources).

If we consider the description of services or of complete systems (often seen as composed services), semantic approaches, like OWL-S [46] and WSMO [41], have been around for a while and now it seems that their market is well identified. Two new initiatives must be mentioned and seem to be very promising for the future. The Unified Service Description Language [11] provides a wide and consistent solution to the homogenous, comprehensive description of (diverse) services, while SoAML (SOA Modeling Language [45]) takes a wider approach and provides interesting means for the description of complete service-based systems.

Service technology foundations: Numerous R&D activities address the technology foundations for services-based systems. These include all facets related to the services technology stack [33]. At service infrastructure level, self-adaptive and federated cloud solutions are being developed to provide guaranteed levels of quality [39]. At service composition level, advanced service composition (orchestration) engines and service component models are being devised to better support QoS aspects and dynamic adaptation [47]. Finally, at business process level, technology is developed to support the near real-time control and management of complex business interactions. More unconventional directions currently include novel computational models for adaptive systems (such as chemical computing [13]), as well as new architectural styles (going beyond SOA and RESTful paradigms) to support long-lived services-oriented systems [1].

Cross-layer adaptation: Adaptation is considered a key capability of service-oriented systems to address the highly dynamic settings in which these systems must live [15]. One important current stream of research in this area is multi-layer monitoring and adaptation, which looks at the monitoring and adaptation mechanisms across the whole service technology stack (see above). Current work involves novel techniques for combining and correlating observations and events from different sources, as well as avoiding potential conflicts that may arise due to uncoordinated adaptations in different layers [1]. In fact, the Future Internet initiatives considers cross layer adaptation one of most relevant research topics, right after context-awareness and human-in-the-loop adaptation [34].

Service and software engineering life-cycle: Researchers are currently developing novel life-cycle models and engineering methods to support the design and operation of adaptive service-oriented systems. One key feature of these models is that adaptation is already considered in the early stages of software and service development [9]. For example, the S-Cube lifecycle model features two dedicated loops: engineering (incl. evolution) and adaptation [33].

We have also started to understand how to include human-provided services in service compositions, how to migrate existing service engineering techniques and processes to incorporate design-for adaptation activities, and how to ensure that the design activities remain agile to allow for a better alignment with the high dynamism embedded in modern SOA systems [33].

B. Convergence of IoS and IoT

Made possible through advancements in information and communication technology, we will see the seamless integration of virtual services (such as financial and telecommunication services) with real-world services (such as transportation or manufacturing). Service-orientation will foster and ease cross-organizational data exchange and integration of IT sys-

[1] http://www.future-internet.eu/

tems. Especially, the Internet of Things (IoT) will lead to an unprecedented access to more data sources (e.g., through the availability of cheap connected sensors) and thus will foster the seamless and effortless access to operational data from everywhere at any time [34]. Such data availability and online access, as well as novel means to interact with the virtual and real-world services, will open up opportunities for innovative ways of monitoring, controlling, adapting and managing business processes and business interactions. However, for this vision to become true, many challenging research issues must be addressed:

- We will see an even stronger decentralization of systems together with the lack of control of such decentralized entities. This requires novel ways to monitor and adapt these systems.
- Due to many different stakeholders and data sources, there is an increased need for understanding the quality of those data. Here, issues such as accuracy and timelines of data from the IoT, as well as trustworthiness of data providers in the IoS must be addressed. On the positive side, the huge number of data sources and data items may allow for data fusion, correlation and consistency checking.
- The high dynamism of SOA systems requires (near) real-time processing of large data streams from distributed sources to observe problems and deviations (e.g., such that adaptations can be executed). This means that current solutions, which often work on post-mortem data, must be significantly enriched to handle such dynamism at runtime.

Many and different kinds of services imply more complex (eco)systems and more diverse and feature-rich technology stacks. We need service infrastructures that are able to scale with respect to both provided services and possible users. For example, the set of services available in a given building can be seen as an ecosystem, but one may also think of the services for a particular event, those provided by a given organization, and also those offered to a particular group of users. Services can belong to different ecosystems at the same time, and enter and leave them dynamically according to some rules. Within an ecosystem, they interact among them and with the users through their capabilities, the context of the interaction (and the profile of the user), their openness, and the perception they have of the others (and the others have created in the ecosystem). All these elements can vary while time elapses, and they can also vary from relation to relation.

The idea of service ecosystem [29] is reality. Its actual enactment however needs special-purpose frameworks to create the ecosystems, oversee and manage the provision of services, and provide users with powerful means to exploit them over a secure layer. The eco-system must consider the degree of privacy each organism wants to keep, the level of trust it offers to the others and they provide to it. A comprehensive solution must also take into account the willingness of the parties to cooperate, the perception the services are willing to provide what promised, and the experience gained in the ecosystem by the different organisms.

C. *Proactive Adaptation*

Adaptation has often been proposed as a solution to allow services-oriented systems to become resilient against failures and changes in third-party services [15]. In this context, we see a recent trend to complement solutions for reactive adaptation (i.e., repairing a system in response to failures that have actually occurred) with proactive capabilities (i.e., modifying the system before a failure actually occurs). Proactive adaptation thus means that the service-oriented system can try to apply countermeasures to prevent the occurrence of a failure, or it can prepare repair mechanisms for the upcoming failure to reduce the time-to-repair [39].

Adaptation capabilities will become even more relevant in the extremely dynamic and complex setting of IoS/IoT services-based systems. The opportunities and benefits for proactive adaptation will even increase, but a prerequisite to proactive adaptation is to anticipate ---during run-time--- failures and problems. However, this online quality prediction faces many important research issues:

- Ensuring that online failure prediction is accurate is critical. Otherwise, wrong predictions may lead to the execution of unnecessary adaptations (false positives) or to missing adaptation opportunities (false negatives) [33]. As an example, unnecessary adaptations may introduce severe problems; e.g., when a working service is replaced by a buggy one. Providing accurate failure predictions becomes extremely challenging in the setting of service-oriented systems, especially if they consist of third party IoS and IoT services due to the heterogeneity and dynamism of the entities.
- Traditionally, accuracy of predictions is assessed in a "post-mortem" way to select a matching prediction technique for a specific usage setting. However, in future services-oriented applications, usage settings or contexts will continuously change (cf. Section IV). This means that even if high accuracy is achieved in an initial setting, it may quickly decrease over time. We thus need new ways to assess the accuracy online, and also means to determine at run-time whether predictions should be trusted.
- The challenges towards online quality prediction are further amplified in the presence of noisy and uncertain data (see above). Open issues involve how to reason and predict in the presence of such noisy data and how such uncertainties impact the accuracy of predictions.

D. *Service Selection and Interaction*

The more services become available, the more prominent and attractive this problem becomes. As long as we have a limited amount of services, or we must simulate them, the problem of identifying, filtering, and selecting services is not important, but when the number of alternatives grows, then efficient, precise, and customizable solutions become mandatory.

The first issue one should address is the proper description of available services, and then these descriptions must be collected and offered to interested parties. Unfortunately, service repositories (e.g., UDDI repositories [2]) are almost dead and it seems there is no interest in pursuing research on them and on efficient means to distribute service information. The same applies to selection and filtering, which are not considered to be hot topics anymore. The current solutions, based on UDDI and/or semantic approaches, are not (good) enough. It seems that the main problem was the automatic discovery and selection of available services, while nowadays the efficient distribution of information, along with the heterogeneity of services, would be much more important. Moreover, available services depend on the context of use, and since the context can change frequently, the set of suggested services should be updated accordingly.

Even if semantic approaches have demonstrated their capabilities in well-defined contexts, the fully automated selection and invocation is not applicable in many situations. However, the underlying infrastructure must be capable of suggesting the right services without overburdening the user. This is why the identification of proper (graphical) interfaces that can interact with the user during the discovery process becomes mandatory. The user must identify the services that are available and that meet his/her interests easily and properly. A search engine-based approach would not be applicable since the precision would be too low. We need efficient solutions, maybe based on some kind of subscriptions, where users can only be informed of the services they may be interested in out of the thousands around them.

Orthogonally, there must be ways to interact with selected services. Currently, many user-oriented services come with pre-defined interfaces, but we cannot realistically think of implementing new user interfaces whenever we add a new device or a new operating system. In contrast, we should do better at decoupling the actual services from their interfaces and reason on approaches to associate services with usable and cross-platform GUIs. HTML5 [20] is a first step in this direction, but its development, and the development of similar initiatives (W3C Web apps working group[2]) are only at the beginning.

III. Mobile Services

While mobile services incorporate and apply the fundamental principles of SOA, they present a number of specifics that push certain challenges related to service-oriented systems to their extreme and additionally introduce new unique research challenges. Such specifics relate to: (i) dynamics – open mobile environments are much more volatile than typical service environments with services emerging and disappearing in arbitrary ways without prior notification; (ii) heterogeneity – a direct consequence of ad-hoc mobile environments is that no safe assumption can be made about the technological and business features of the services encountered; (iii) awareness – in most mobile service applications, the business capabilities of services are not the only ones that matter, the multi-faceted context of services is equally important; and (iv) the equation among QoS expectations on services, scalability, and required resources is hard to solve, due to the resource constraints that are typical to mobile environments. Dealing with the identified specifics gets even more complex if we consider both traditional computing services and services attached to the physical world by means of sensors and actuators, i.e., things. In the following, we then discuss the open research challenges in this context.

A. Description and Discovery in ad-hoc Settings

Service description is a fundamental element in SOA, as it determines the information that a service needs to expose to its environment for enabling its unambiguous identification and use. A variety of service description languages have been proposed to cover different aspects and are currently in use. Some of them have also reached the status of standard[3] and others are still research proposals. Such aspects include the profile, that is, a high-level business description of a service, the interface, the behavior, that is, the observable supported execution patterns of the service in coordination with its environment, the QoS properties, and the service binding, that is, the information required for accessing the service at the underlying middleware protocol level. Besides employing an established syntax for describing these aspects, these languages may further make the semantics of the different aspects explicit by referring to a structured vocabulary of terms (ontology), which represents a specific area of knowledge [46]. As already pointed out, mobile environments are typically very dynamic and ad-hoc, which calls for rich service descriptions, covering most of the above aspects, and their automated processing at runtime for enabling a realistic mobile SOA. Both conditions still constitute open research challenges.

In particular, mobile services may be hosted on platforms that range from wireless resource-rich machines to wireless resource-constrained devices, and further to any physical object (thing) enhanced with some networking capacity. Such extreme heterogeneity should be accounted for in service description. Furthermore, service heterogeneity is exacerbated by the diversity in both business semantics and communication middleware. As for the former issue, dealing with ontologies in delimited service environments is already hard, due to the lack of widely accepted ontologies and the fact that the heterogeneity problem, which ontologies had 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 mobile service context. Establishing reference or global ontologies [1] and tackling the ontology heterogeneity problem [40] are two key requirements in the area. On the other hand, the diversity in communication middleware is due to the fact that mobile services may use different networking contexts. This calls for support for heterogeneous interaction models, namely message-driven, event-driven, and data-driven models. Different interaction 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 supported by current service description languages too stringent, since they comply with a single (client/service) message-based interaction model. Service de-

[2] http://www.w3.org/2008/webapps/

[3] http://www.w3.org/TR/wsdl20/

scription should be able to abstract and comprehensively specify the various service bindings of mobile services. This further implies extending the notion of service and introducing adequate service interaction modeling.

At the same time, the need for automated runtime processing of service descriptions creates a trade-off between richness and efficiency given the resource constraints of mobile devices. This trade-off concerns the storage, publication, search, access, and reasoning about service descriptions, as enacted for the purposes of service discovery. In particular, it calls 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 to considerably accelerate semantic reasoning when executed in dynamic and resource-constrained environments [1].

Existing work in the area of service discovery lies in two strands considering the problem from different perspectives. The first strand of work has addressed the issues of heterogeneous service discovery protocols. The challenge here is to ensure interoperability among such protocols both at the level of protocol interaction semantics and at the level of service descriptions conveyed by such protocols [17]. Even if considerable work has been done in this area, the variety and heterogeneity of protocols limit the impact of existing solutions. The second strand considers the problem of matching stored advertisements to requests. The problem here is that commonly available service descriptions cover only a part of the service aspects identified above and that are required for effectively reasoning in an automated way on the composability of mobile services. In particular, information on service behavior and QoS properties is often lacking, while there are no standardized service bindings for interaction models other than message-based. Additionally, the business semantics of services is usually only implicit in the service description syntax. Initial solutions to the lack of required information have employed passive and active machine learning techniques to: (i) extract the business semantics of services from the description of the syntactic interface or from text in natural language provided by developers to describe the purpose of the service or a specific action [4], and (ii) learn the service behavior as well as its non-functional properties given its interface [20]. These approaches are promising even if their applicability in realistic cases has still to be proved. When required service information is available, the matching between service descriptions is based on (variously) syntactic, semantic [1] and behavioral [37] analyses. Despite existing work in the area, this is commonly not adapted to the specifics of mobile services, where resource constraints and user interactivity place inflexible efficiency and performance requirements. Related to this latter issue is also the need for efficient service repositories, where the organization of stored service descriptions and related indexing techniques can considerably accelerate the lookup of services [1]. Additional challenges here are that such repositories should be maintained at runtime and they should also constantly reflect the extremely volatile population of available mobile services.

B. Heterogeneity, Access-awareness, and Composition

As already pointed out, networking contexts of mobile services are characterized by high diversity of the communication middleware infrastructures with respect to employed 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 interaction model, mobile service access and composition is required to support heterogeneous models and to enable interoperability among them.

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 interaction 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 interaction models. Then, some local mapping is performed between the API operations and the models/protocols supported. Wrapping systems behind standard technology interfaces enables the access to these systems via interaction models that are different from their native ones. For instance in [1], 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 protocols. The ESB (Enterprise Service Bus) paradigm is currently the dominant bridging solution for the integration of heterogeneous systems. By employing appropriate ESB adapters, systems with diverse interaction models can be plugged on the bus. The above interoperability solutions are mostly deployed statically. Other efforts have aimed to provide dynamic transparent interoperability among 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 [36]. However, interoperability between different interaction models is not addressed. Despite a number of approaches dealing with interoperability between interaction models, provided solutions are in general ad-hoc and concern specific cases. An overall solution to this issue is required for mobile services, based on appropriate modeling abstractions and transformation mappings between models [18]. Moreover, we also need a precise evaluation of such mappings with respect to the preservation of semantics.

Besides the heterogeneity at the level of communication middleware discussed above, the access to and composability of mobile services must deal at runtime with heterogeneity in the way services provide their business capabilities to their environment. The environment itself may vary significantly in terms of interfaces, data and behaviors, even for services that are functionally compatible, i.e., that provide and require capabilities in a complementary way. As already discussed in the previous section, service discovery may (at least partly) figure out the possibility of composing two (or more) services based on their business semantics, which should be compatible even if there may be differences in the way these semantics are enacted. Then, an important challenge is to be able to automatically synthesize a mediator that can enable the composition of services by seamlessly resolving their differences.

Interoperability and mediation are very popular topics in the literature and have been investigated in several contexts. Indeed, since the early days of networking, many efforts have focused on formal approaches to protocol conversion [25]. Mediation of protocols has received attention in fields such as integration of heterogeneous data sources, software architectures, architectural and design patterns, patterns of connectors, Web services [27], and networked system interoperability [22]. By focusing on mobile service interoperability, we are interested in automated approaches that do not require human intervention in the construction of a mediator. Automated approaches fall into two main categories: generative approaches, which try to synthesize a mediator by inferring it from service specifications, and restrictive approaches, which try to restrict the service behaviors in such a way that mismatches are avoided. As for generative approaches, we distinguish two phases that should be present for a fully automated solution: a common language identification phase and a protocol mapping phase. There are only few existing approaches, which however present certain shortcomings. For example, performance is compromised due to the high cost of exhaustive graph exploration algorithms that prevents their usage in the context of mobile services [32]. Finally, we specifically point out the need for considering in a combined way both service behavior heterogeneity and heterogeneity in the underlying communication middleware as discussed above: this is a challenging research issue since almost all approaches only consider client/service interactions [22].

From another perspective, awareness of the mobile environment is a key factor to be considered in middleware supporting the access to and composition of mobile services. Awareness may be considered from two viewpoints: (i) mobile service middleware should be able to capture the dynamically changing conditions and resource limitations of the underlying networking environment and of the interacting services, and (ii) such middleware should provide services for context-awareness and thus services for personalization and adaptation. These challenges point to research that has been undertaken for quite a few years in the domains of mobile and pervasive middleware as well as middleware for wireless sensor networks.

A number of related research efforts focus on architectural and interaction models for pervasive applications. Such models aim to be modular, flexible and dynamic to enable applications to deal with the uncertainty and dynamics of the mobile environment. Other approaches explicitly adopt the SOA paradigm and provide "aware" solutions to service/resource discovery, access, and composition that are customized to the specifics of pervasive computing. Thus, [10] proposes a solution where the access to services hosted by mobile devices is conditioned by the multi-radio, multi-network character of the communication with pervasive devices. Context is a key element for awareness and adaptability. Middleware facilitating the development and execution of context-aware applications is proposed in [24]. Then, a number of efforts focus on dynamic adaptation, which may be context-aware or QoS-aware, and in particular energy-efficient. 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. Despite these rich research results, mobile services in the Future Internet setting must face unique requirements in terms of openness and hence awareness, which make existing solutions inadequate. For instance, wireless sensor networks still remain mostly closed systems, accessible to the rest of the world through external gateways [1]. On the other hand, with sensors being increasingly pervasive – such as sound, GPS, accelerometer and other sensors embedded in users' smartphones – and the anticipated evolution to the Internet of Things, sensing becomes ubiquitous and "participatory" as each user can be involved [28]. This fact, together with the resulting scale, creates new needs for aware and adaptable programming abstractions for the middleware underlying mobile services.

IV. Services and Information systems

Services in the information systems (IS) domain have been studied under different perspectives. On one hand, researchers in the IS area have been focusing the on the design of service-oriented systems by considering services as enablers of cooperation among information systems in different organizations. On the other hand, the theme of service science has been investigated: it provides a broader scope to research since it considers the technological, organizational, economic, and social aspects related to services. According to [48], these aspects are the pillars of service systems connected through a value proposition. The aim is a dynamic co-creation of value through the participation of customers and of external/internal service systems.

When considering the cooperation among diverse systems, the focus has been on providing homogenous interfaces to access the services of the different organizations. Under this perspective, we can consider BPEL and WSDL as starting points for defining the processes and their interacting services: [33] provides a wide perspective on technological solutions. Among the many topics considered to conceive advanced service-based systems, the following ones are currently touched by state-of-the-art research:

- ***Quality of service***: the representation of quality dimensions and their association with services is critical in systems in which service providers are dynamically selected on the basis of their characteristics;
- ***Service selection***: the ability to select services that provide required functionality and quality of service has been studied to optimize the selection and to take into account the semantics of the services being provided;
- ***Standardization of interfaces and their meaning***: many research efforts are in the direction of defining the functionality provided by services in a precise and unambiguous way. Ontologies and annotations are the basis for providing such information, but also the description of

the behavioral aspects of services and their underlying processes play a fundamental role;

- ***Flexibility and adaptability***: service-based systems are often characterized by the ability of reconfiguring themselves dynamically to cope with variability in the execution environment, changes in used services, and variations in requirements;
- ***Compliance***: recent work is focusing on the ability of guaranteeing service quality both by means of proactive approaches based on adaptability and through the formal verification of service properties.

Cooperation is also between service-based systems and their final users. The *experience* brought to the users is a key issue to be addressed. The IS community has traditionally developed a body of knowledge around the topics of HCI that can be extended to include elements regarding the service experience: personalization and profiling of services, multi-modal interfaces depending on the media, space and time location, and context-awareness in service provisioning.

As a design technique, the idea of SOA is interesting in the adopted service-centered view of the world where all real-life effects can be modeled as one or a collection of services: each service has its own capabilities provided through well-defined interfaces. The idea of separating data and processing makes it possible to assemble and reuse services to develop or maintain IT systems. The goal is to design a loosely coupled system that allows for the use of independent service providers.

Methods for the design of such systems are still under investigation. In particular they focus on the definition of services with an appropriate level of *granularity* to balance the need for reuse with the ability of retrieving and selecting services that have a significant size from the user's perspective. They also touch *exception management* and recovery ---in particular for transactional services---, and the ability to link services to *requirements*, goals, and their representation, specifically within inter-organizational processes that involve several partners.

These last topics lead to the second main area of interest for the IS community: services are not only considered under a mainly technological perspective, but they are framed in broader perspective. Under the term *service science* [48], services are considered in the broader context of the organizations that use them and of the humans that perspective them. They also consider the social aspects of providing a service, and the way it is created, provided, consumed, and withdrawn.

As for the *design and management lifecycle*, the community concentrated on phases such as service management, adaptation, and evolution, and it also studied the conformance to process reference schemas. In particular, the management of services has been considered both within an organization and across different organizations.

From a business perspective, a central concept is the alignment of IT services and business needs, where *alignment* is defined as the degree to which the IT mission, objectives, and plans support the business mission, objectives and plans [40]. The business-IT alignment problem has been intensively studied since the initial work of [19]. However, most of the literature focused on the macro level, that is, whether strategies are aligned to the information systems, to the structure of the IT organization, to the overall IT strategy, and to the IT architecture. But few studies have looked at alignment at the micro level: it is exactly at this level that SOA is primarily meant to make the difference.

SOA also promotes business agility. In a recent special issue on agility [31], it is argued that: "the agile organizations respond quickly, they are resourceful, and they are able to adapt to their environment". A service-oriented approach may offer a solution to the agility problem, since the technology is meant to embrace change, and allows companies to adapt to a dynamic business context: "SOA enables an IT portfolio which is also adaptable to the varied needs … is also more agile and responsive … [and] SOA can also provide a solid foundation for business agility and adaptability" [29].

SOA offers a promising solution to two major problems, alignment and agility, with which many companies have been struggling for years. It is therefore no surprise that more and more organizations are engaging in SOA. However, the importance of alignment between business and IT and the need for business agility are putting a lot of pressure on IT departments to respond to changes quickly and in an effective way. Yet resources are scarce and eventually it becomes a matter of prioritizing the initiatives to pursue. Typically this prioritization process uses cost-benefit analyses or business cases as inputs, but solid estimates of the costs of projects are needed.

If one considers the *economic* aspects, a service is seen as an activity or as a series of activities "executed" in an exchange between a supplier and a customer, where the object of the transaction is intangible. Services have the inherent (but only potential) goal of producing value, or, better, of making the supplier and the customer to co-produce value prior and during the above mentioned exchange. This means that the concept of value is central in service science, where value can mean value in exchange, value in use, and social/public value, according to the view of providers, public administrations, and users, respectively.

As for *skills and competences*, there is also the problem of training the engineers and architects to conceive the architecture of innovative service-based systems. This covers aspects related to creativity, entrepreneurship, management of trans-disciplinary teams, and management of the governance framework. These experts must also be able to evaluate the feasibility of a single service, or of an entire system, from a technological perspective, according to a business model, and given a value proposition. They must also be able to check that the final solution complies with regulations and environmental constraints.

V. DISTRIBUTED SERVICE-ORIENTED SERVICES

Advanced service-oriented systems in distributed environments can, in general, be characterized by the following properties: (i) services are *composed of* various service components and hence constitute service-based processes, (ii) such components may reside at many *different locations* − also dynamically changing over time, and (iii) such location as well as contextual changes have to be constantly *moni-*

tored to be able to (iv) *manage* complex process or service executions so that they can (v) *adapt* dynamically to such changes during execution(s).

Closely related to these characteristics, the following research questions arise: How can one detect appropriate services in dynamically changing distributed environments? How can one relate services as well as processes in such environments to dynamically changing contexts? How can one guarantee service-level agreements in new and (constantly and dynamically) changing environments? How can one enable processes to be executed (fully or in parts) in such environments? How can one migrate process fragments to environments where they can continue to be executed despite device limitations and/or dynamic contextual changes, e.g. on mobile devices? How can one monitor service as well as process contexts and migrations? And finally: How can one manage distributed migrating services in dynamically changing contexts such that they can adapt to all changes dynamically whenever they occur?

In summary, important challenges for flexible execution of distributed processes in service-oriented systems arise from service as well as process distribution, and the fact that distribution contexts may change unexpectedly. Respective research issues focus, e.g., on process instance migration, context modeling and processing, adaptation strategies as well as service and process monitoring, management, and adaptation in dynamically changing environments [33].

Finally, research and software development in distributed service-oriented systems has to provide appropriate infrastructure ("middleware") components for executing distributed processes in service-oriented environments.

A. Current Research Topics

In service-based systems, distribution appears at both the service infrastructure level and at the process execution level. At the infrastructure level, resources are distributed as services on networks like the Internet or the ubiquitous "internet of things". In such heterogeneous distributed environments, services can appear and disappear at any time as well as change their functional and/or non-functional properties in dynamically changing contexts. Therefore, the first step in realizing distributed service-based applications is to bind to all necessary components: The respective dynamic service binding comprises service discovery, the "process of finding services that match the requirements of the service requestor" [39], and service selection where specific services out of a set of suitable services are chosen dynamically. In this area, most research efforts so far have addressed providing advanced semantics for service bindings based on *functional* service properties. Nevertheless, also *non-functional* properties, such as costs, reputation and trust of a service, have become increasingly important as well, especially with respect to market-based service selection mechanisms [50] and cloud-based environments.

However, not only services but also *processes* (resp. process executions) can be spread over different locations and various contexts of a distributed environment. *Distributed process execution* can be advantageous, e.g., to maintain the autonomy of business partners in cross-organizational processes, for legal causes, due to technological reasons such as incompatibilities or mobility issues (e.g. temporary unavailability of nodes or connections) or non-functional requirements (as, e.g., distrust in or high costs of nodes). Also distributing (parts of) process executions can be suitable for fast and flexible reaction to changing requirements and dynamic markets; therefore *process instance migration* has to be supported as well. This allows for transferring process instances to other execution engines – which then continue to execute them starting from their respective state at the time of transfer [52]. Here, the selection of respective process participants can be performed ad-hoc at runtime by taking the actual process execution context into consideration.

To prepare for *context adaptation*, contexts have to be *modeled* adequately first. A respective context model includes relevant objects in the environment that are used to characterize the situation of the process, e.g., its physical location, time, available services and infrastructure. Representations of context models can be classified as *key-value models*, *mark-up scheme models*, *graphical models*, *object oriented models*, *logic based models* or *ontology based models* [47]. In addition (whenever possible), *predicting a future context* is another important goal for enabling devices and applications to *proactively* adapt to their (dynamically changing) environment or to enable automatic execution of tasks even in such environments [29]. Currently, the development of context-aware applications can be facilitated by reusable components such as generic context management systems and middleware support [1], so that context-aware applications can be built on the basis of generic frameworks. As an extension to such systems, *structured context prediction* offers reusable system support for the adaptation even to future contexts in order to support context-aware applications that not only consider their current context but that can derive and use information about future situations as well. On this basis, prediction support for different application areas can be realized to facilitate the development of *future context-aware* applications [53].

Other important issues in distributed process execution concern the specification of *process goals* and *service-level agreements* (SLAs) between process resp. service users and providers ---covering both functional and non-functional properties. Service-level agreements traditionally stipulate the functionality to be provided and respective guarantees on non-functional aspects, such as performance, availability, costs, security, or other attributes. Considering dynamic markets, SLAs should preferably be negotiated automatically at runtime with respect to predefined business goals [13]. Organizations can then bid for process execution at different levels of quality. Corresponding agreements have to be monitored and evaluated at runtime in order to examine their compliance with predefined business goals.

Therefore, *monitoring* services and processes becomes also a crucial matter for ensuring and verifying the desired progress of (automatically running) processes. Even more challenging *distributed monitoring* approaches are required for cross-organizational processes that span multiple, heterogeneous infrastructures consisting of various autonomous systems. Existing infrastructures, e.g. an *enterprise service* *bus* (ESB), can be used here as a first step. However, dynamic interactions between autonomic organizations cannot always rely on the existence and compatibility of such an ESB. Specific monitoring agreements are hence required to share process-relevant events between organizations and can therefore extend the respective SLAs [51].

Future emerging service-based applications may also use monitoring functions for *explaining* causes of SLA violations which can then be used to improve processes a posteriori. Here, small errors at the infrastructure level can already cause considerably larger problems at the business level. Respective correlations of low-level events to high-level business events are therefore required. Moreover, advanced distributed process systems shall be able to predict future SLA violations based on such monitored information [25]. And this information can, finally, be used to initiate appropriate reactions and help prevent such processes from failing.

B. Managing Business Processes In Distributed Systems

The management of cross-organizational business processes in distributed systems is one of the most challenging emerging research topics. It addresses the aforementioned challenges in order to provide support for flexible, adaptive and reliable service-based applications. Based on monitored (past and current) and predicted information about the process, its context and corresponding service-level agreements, a holistic approach for a fully autonomous process management at all levels of the service-based application can be imagined.

The overall goal is to enable such processes to *adapt* to all possible changes in distributed and dynamically changing environments (as, e.g., loss of nodes, unavailability of connections, unreliable servers, changing contexts, etc.). Specific adaptation tasks of a respective process and service management system include, e.g., rebinding of services to improve non-functional attributes, migrating process instance to better suited execution engines, or adjusting operational aspects of the process (e.g. the control flow) in order to align the structure of the service-based application to new situations dynamically. Another possible adaptation strategy could rely on *automatic renegotiation of SLAs* that are predicted to fail in the future [25], in case involved partners allow for it.

All these adaptations have always to be performed “correctly”, i.e., with respect to high level SLAs or *process goals* that should preferably be expressed in abstract, non-procedural, and implementation independent ways. To specify such goals, emerging approaches use, for example, rule-based, artifact-centric, or goal-oriented [13] modeling techniques. Appropriate *agreements* between multiple participants of a cross-organizational process should also include specifications of service-level agreements, key performance indicators, respective monitoring agreements, thresholds, and penalties or options in case of violations. A *holistic management solution* shall handle such business goals and agreements, monitor the execution of service-based processes, self-adapt the process, if needed, and self-optimize the service-based system, whenever possible.

VI. Conclusions and Future Work

The paper presents a first set of ongoing and future research directions in the broad field of SOA and service-based systems. This summary comes from the view of the authors, and from some informal interviews with key figures in the research field. The aim is to continue the work by involving further researchers, and thus by collecting further opinions on the future of service-oriented systems and technologies to shape a complete and detailed research agenda for the working group, but also for the community at large.

Acknowledgment

The authors would like to thank Jan Pries-Heje (TC 8 chair), Eric Dubois, and Carlo Batini for their valuable comments and contributions. Moreover, the research leading to these results has been partially supported by the 7th Framework Program of the European Community, under grant agreements 215483 (S-Cube) and 285598 (FInest).

References

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

[2] T. Bellwood et al.: "UDDI version 3.0.2", OASIS http://uddi.org/pubs/uddi_v3.htm, 2004

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

[4] A. Bennaceur, R. Johansson, A. Moschitti, R. Spalazzese, D. Sykes, R. Saadi, and V. Issarny. Inferring affordances using learning techniques. In International Workshop on Eternal Systems (EternalS'11), 2011.

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

[6] D. Bonetta and C. Pautasso: “Towards liquid service oriented architectures”. WWW (Companion Volume) 2011: 337-342.

[7] D. Booth et al, W3C Web Services Architecture, http://www.w3.org/TR/ws-arch/, 2004

[8] A. Bucchiarone, C. Cappiello, E. Di Nitto, S. Gorlatch, D. Mailänder, and A. Metzger: “Design for self-adaptation in service-oriented systems in the cloud,” in European Research Activities in Cloud Computing, Cambridge Scholars Publishing, 2012.

[9] A. Bucchiarone, C. Cappiello, E. Di Nitto, R. Kazhamiakin, V. Mazza, and M. Pistore: “Design for adaptation of service-based applications: Main issues and requirements”. In: Engineering Service-Oriented Applications: Supporting Software Service Development Lifecycles (WESOA) co-located with ICSOC/ServiceWave, 2009.

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

[11] J. Cardoso, A. Barros, N. May, Norman, and U. Kylau: "Towards a Unified Service Description Language for the Internet of Services: Requirements and First Developments", IEEE International Conference on Services Computing, IEEE CS, 2010.

[12] G. Chen and D. Kotz : “A survey of context-aware mobile computing research. *Technical Report TR 2000-381*, Dartmouth College 2000

[13] C. Di Napoli, M. Giordano, Z. Németh: Alchemy of Services. ERCIM News 2011(85): 37 (2011).

[14] E. Di Nitto, M. Di Penta, A. Gambi, G. Ripa, and M.L. Villani: “Negotiation of Service Level Agreements: An Architecture and a

Search-Based Approach". *Proceedings of the 5th International Conference on Service-Oriented Computing* , 295–306, 2007.

[15] E. Di Nitto, C. Ghezzi, A. Metzger, M. P. Papazoglou, and K. Pohl: "A journey to highly dynamic, self-adaptive service-based applications," Autom. Softw. Eng., vol. 15, no. 3-4, pp. 313–341, 2008.

[16] D. Driscoll and A. Mensch: "Devices Profile for Web Services (DPWS)" v.1.1, OASIS Standard, http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01, 2009

[17] C.A. Flores-Cortés, P. Grace, and G.S. Blair. Sedim: A middleware framework for interoperable service discovery in heterogeneous networks. TAAS, 6(1), 2011.

[18] N. Georgantas, M. A. Rahaman, H. Ameziani, A. Pathak, V. Issarny. A Coordination Middleware for Orchestrating Heterogeneous Distributed Systems, in "International Conference on Grid and Pervasive Computing, GPC 2011", 2011, p. 221-232.

[19] J.C. Henderson and N. Venkatraman, "Strategic alignment: a model for organizational transformation through information technology", Oxford University Press, New York, 1992.

[20] I. Hickson: "HTML5: Edition for Web Authors", W3C http://dev.w3.org/html5/spec-author-view/, 2012

[21] F. Howar, B. Steffen, and M. Merten. Automata learning with automated alphabet abstraction refinement. In VMCAI, 2011.

[22] V. Issarny, A. Bennaceur, and Y.D. Bromberg. Middleware-layer connector synthesis: Beyond state of the art in middleware interoperability. In Formal Methods for Eternal Networked Software Systems (SFM'11). 2011.

[23] K. Jander, L. Braubach, A. Pokahr, W. Lamersdorf, K. Wack: "Goal-oriented Processes with GPMN", *International Journal on Artificial Intelligence Tools* (IJAIT), vol. 20, no. 6, World Scientific Publishing, 2011, 1021-1041.

[24] C. Julien and G.C. Roman: Egospaces: facilitating rapid development of context-aware mobile applications. IEEE Transactions on Software Engineering 32(5) (2006)

[25] S.S. Lam. Protocol conversion. IEEE Transaction Software Engineering, 1988.

[26] P. Leitner, A. Michlmayr, F. Rosenberg, and S. Dustdar: "Monitoring, Prediction and Prevention of SLA Violations in Composite Services". *Proceedings of the IEEE International Conference on Web Services (ICWS)*, 369–376, IEEE Computer Society, 2010.

[27] X. Li, Y. Fan, J. Wang, L. Wang, and F. Jiang. A pattern-based approach to development of service mediators for protocol mediation. In proceedings of WICSA '08, pages 137–146. IEEE Computer Society, 2008.

[28] H. Lu, W. Pan, N.D. Lane, T. Choudhury, and A.T. Campbell: SoundSense: scalable sound sensing for people centric applications on mobile phones. In: MobiSys (2009)

[29] C.M. MacKenzie and others, Reference model for service oriented architecture, V.1.0, 2006.

[30] R. Mayrhofer: "Context prediction based on context histories: Expected benefits, issues and current state-of-the-art", *Proceedings ECHISE '05*, *Extending the Growing Neural Gas Classifier for Context Recognition*, Lecture Notes in Computer Science, vol. 4739, Springer-Verlag, 2007, 920-927.

[31] L. Mathiassen and J. Pries-Heje: "Business agility and diffusion of information technology", European Journal of Information Systems, vol. 15(6), pp. 116-119, 2006.

[32] J.A. Martìn and E. Pimentel. Automatic generation of adaptation contracts. In Proceedings of FOCLASA, 2008.

[33] A. Metzger, "Towards accurate failure prediction for the proactive adaptation of service-oriented systems (invited)," in Proceedings Workshop on Assurances for Self-Adaptive Systems (ASAS), collocated with ESEC 2011, 2011.

[34] A. Metzger and C. Cassales Marquezan, "Future Internet Apps: The next wave of adaptive service-oriented systems?" in ServiceWave 2011, Springer, 2011.

[35] A. Metzger and E. Di Nitto, "Addressing highly dynamic changes in service-oriented systems: Towards agile evolution and adaptation," in Agile and Lean Service-Oriented Development: Foundations, Theory and Practice, IGI Global, 2012 (to be published).

[36] J. Nakazawa, H. Tokuda, W. Edwards, and U. Ramachandran: A bridging framework for universal interoperability in pervasive systems. In: IEEE International Conference on Distributed Computing Systems (ICDCS) (2006)

[37] O. Oanea, J. Sürmeli, and K. Wolf. Service discovery using communication fingerprints. In ICSOC, pages 612-618, 2010.

[38] M. Papazoglou, K. Pohl, M. Parkin, and A. Metzger (Ed.s): "Service Research Challenges and Solutions for the Future Internet: S-Cube – Towards Engineering, Managing and Adapting Service-Based Systems", *Lecture Notes in Computer Science*, vol. 6500, Springer, 2010.

[39] D. Petcu and J. Vásquez-Poletti, Eds. *European Research Activities in Cloud Computing*, Cambridge Scholars Publishing, 2012.

[40] Y. Qu, W. Hu, and G. Cheng: "Constructing virtual documents for ontology matching". In: 15th international conference on World Wide Web (WWW) (2006)

[41] D. Roman, J. de Bruijn, A. Mocan, H. Lausen, J. Domingue, C. Bussler, D. Fensel: "WWW: WSMO, WSML, and WSMX in a Nutshell", First Asian Semantic Web Conference, Springer, Lecture Notes in Computer Science, 4185, 516-522, 2006.

[42] B.H. Reich and I. Benbasat, Factors that influence the social dimension of alignment between business and information technology objectives, MIS Quarterly, Vol. 24, No. 1, Mar. 2000

[43] F. Salfner, M. Lenk, and M. Malek. A survey of online failure prediction methods. ACM Comput. Surv., 42(3), 2010.

[44] S-Cube EU Network of Excellence, 2011, http://www.s-cube-network.eu/km

[45] SoaML, OMG standard, http://www.omg.org/spec/SoaML/1.0/

[46] M. Solanki, D. Martin, M. Paolucci, S. McIlraith, M. Burstein, D. McDermott, D. McGuinness, B. Parsia, T. Payne, M. Sabou, N. Srinivasan, and K. Sycara, "Bringing Semantics to Web Services: The OWL-S Approach", First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004) July, 2004.

[47] M. Sonntag and D. Karastoyanova: Compensation of Adapted Service Orchestration Logic in BPEL'n'Aspects. BPM 2011: 413-428.

[48] J. Spohrer, P.P. Maglio, J. Bailey, D. Gruhl, Steps Toward a Science of Service Systems, Computer, 40(1), pp. 71-77, Jan. 2007

[49] T. Strang and C. Popien: "A context modeling survey". *Workshop on Advanced Context Modelling, Reasoning and Management, Proceedings of the Sixth International Conference on Ubiquitous Computing*, September, 2004

[50] A. Vilenica, K. Hamann, W. Lamersdorf, J. Sudeikat, W. Renz: "An Extensible Framework for Dynamic Market-Based Service Selection and Business Process Execution". *11th IFIP International Conference on Distributed Applications and Interoperable Systems*, 2011

[51] B. Wetzstein, D. Karastoyanova, O. Kopp, F. Leymann, and D. Zwink: "Cross-organizational process monitoring based on service choreographies". *SAC*, 2485–2490. ACM, 2010.

[52] S. Zaplata, K. Hamann, K. Kottke, W. Lamersdorf: "Flexible Execution of Distributed Business Processes based on Process Instance Migration". *Journal of Systems Integration (JSI)*, vol. 1, no. 3, 2010.

[53] S. Zaplata, M. Meiners, and W. Lamersdorf: "Designing Future-Context-Aware Dynamic Applications with Structured Context Prediction, *Software: Practice & Experience* (SPE), John Wiley, October, 2011