Enhancing User-Service Interaction Through a Global User-Centric Approach to SOA

Javier Soriano
and David Lizcano
Universidad Politécnica de Madrid
Campus Montegancedo
Boadilla del Monte, Madrid, Spain
{jsoriano, dlizcano} @fi.upm.es

Juan J. Hierro
and Marcos Reyes
Telefónica I+D
Emilio Vargas 6
Madrid, Spain
{jhierro, mru}@tid.es

Christoph Schroth
and Till Janner
SAP Research CEC
St. Gallen, Switzerland
{christoph.schroth,
till.janner} @sap.com

Abstract

Considered as enablers of seamless application-to-application integration both within company boundaries and on a global scale, Web Services-based SOAs have traditionally focused on automating service-to-service collaboration. However, they have never featured a "face" to human users. This gap between human users and services still prevents enterprises from realizing how innovations at the SOA front-end help to make people more productive. And, ultimately, it hinders the emergence of a real Web of Services driven by a global, user-centric SOA. In this paper, we revisit the notion of SOA and analyze its major shortcomings with regard to the emergence of a Web of Services enhancing user-service interaction and increased service usability. We then elaborate on novel, currently emerging technologies that facilitate the establishment of the global mesh of interoperable user-centric services. A novel platform architecture is presented that builds on all the key technical enablers.

1 Introduction

Web Services have attracted a great deal of interest over the last few years as they are expected to act as enablers of seamless application-to-application integration both within company boundaries and on a global scale. However, Web Services-based SOAs mostly exist only within company boundaries at present [18], and the global provision and consumption of services over the Internet is still at an early stage [9]. They have not yet taken on a significant role in realizing cross-organizational collaboration in the coming *Internet of Services (IoS)*. High technical complexity, implementation and maintenance costs, inflexibility and the lack of widely accepted standards for defining service chore-ographies, as well as message semantics, have been repeatedly identified as key factors that have prevented the emergence of a global mesh of interoperable Web Services that could foster enterprise innovation.

Moreover, with Web Services-based SOAs users could be empowered with a truly global mesh of interoperable services [17]. This way enterprises could develop an ecosystem of employees, partners, suppliers, and customers collaborating to develop capabilities by collectively generating, sharing, refining and accessing business knowledge and processes through the service metaphor in a continuous innovation process [4]. This vision could even be further expanded as far as the notion of a *global, user-centric SOA* enabling a real *Web of Services* made up of a global mesh of interoperable *user-centric* services.

However, Web Services-based SOAs remain focused on automating service-to-service (i.e. machine-to-machine) collaboration, and they do not feature a "face" to human users, as they reside on a merely technical layer. One further important shortcoming of existing SOAs is related to this gap between human users and services: the lack of a SOA front-end to humans still prevents enterprises from realizing how innovations at the SOA front-end help to make people more productive. This ultimately hinders the emergence of a *Web of Services* driven by a global, user-centric SOA.

The lack of next-generation **portal platforms** allowing for intuitive human-guided service interaction, reuse and composition thus represents a major stumbling block on the path to a global, user-centric SOA. These platforms will serve many companies as a foundational starting point for SOA, as they represent the SOA front-end and provide the essential framework for organizations to give people instant access to the right content (information and data from multiple sources), applications and processes through a seamless user experience that enables real-time decision making and

execution, as well as ease of interaction and collaboration with other people. They also help businesses to easily extend their SOA environment, foster user-service interaction and improve service usability.

With this in mind, we elaborate in this paper on the synergies the Web 2.0 and the SOA concepts have with regard to the development of a global, user-centric SOA. Web 2.0's focus on the inclusion of human beings and the exploitation of users' collective intelligence is considered a key enrichment of the existing SOA concept and is therefore expected to act as an enabler of a global mesh of interoperable services. The remainder of the paper is structured as follows. First of all we revisit the notion of SOA and analyze its major shortcomings with regard to the emergence of a global user-centric SOA (section 2). Also, we elaborate on novel, currently emerging Web 2.0 technologies that facilitate the establishment of a global mesh of interoperable services and shift the roles and characteristics of resource consumers, providers and intermediaries considerably, and present a generic model of a global user-centric SOA based on the synergies the two concepts (Web 2.0 and SOA) have (section 3). We then present a novel platform architecture that builds on all the key technical enablers and has the potential to drive the development of the above Web of Services (section 4). Existing prototypes of this platform are then shown as a proof of concept (Section 5). Section 6 presents other related work. Finally, a later section concludes this paper and presents a brief outlook on future work.

2 Web Services-based SOAs and their short-comings on the road towards a real Web of Services

Nowadays, IT and Internet evolution has stated that services should be more easily deployed by providers, as well as accessed and managed by end users than traditional packaged software [9]. Therefore, **services must be accessible for all users** (not only enterprise stakeholders), because these services should replace a previous wide range of enterprise software [9], [17].

Therefore, e-services should support common daily processes (both business processes carried out by companies and processes conducted by individuals or groups in their daily life) at any time, as flexibly and dynamically as possible.

Business users will see the tools supporting their daily work replaced by composite applications based on Web services, that are not well enough tailored to users and their daily processes. If these users were at liberty to create their own flexible management tools to support their routine operations, this would encourage business innovation. If they were able to create their own tools from friendly interfaces, they would put substantial innate knowledge of the business

into their construction, and this would encourage business innovation and development.

Obviously, SOA, as it was originally conceived [1], represents an architecture focused fundamentally on a B2B context [10] and weak for B2C problems, since it does not offer the best prospects for dealing with user-service interaction. We can tackle its shortcomings from three different perspectives:

1. **SOA's aim**: Conventional SOAs merely aim at facilitating seamless machine-machine collaboration. SOA deployments are very abstract and invisible to users. Its customers of choice are medium-sized or larger corporations instead of normal end users along the long tail [2] of Internet. Therefore, with SOA, normal Internet users with little IT expertise have not been able to easily retrieve and use services because services mostly reside within company boundaries and are only accessed for professional use in a corporate context.

2. **SOA's technology**: Apart from SOA's aims, this architecture relies on a set of complex standards that are not user friendly. As WSDL (as a services description utility), UDDI (as support for the repository of these services), BPEL (as a composition and integration tool) and SOAP (as an access protocol) are the most widespread standards used to set up SOAs, they are not oriented to be readable or understandable for end-users. Because of high SOA's inherent technical complexity, there needs to be one or more expert players within the value chain to build and provide solutions for their customers. In contrast to this one-to-many value chain model of numerous SOA use cases (where one expert serves many clients), new value chains should began to be mostly loosely coupled (many-to-many) networks of self-managed self-sufficient users who can offer and consume resources via the Web.

3. **SOA's government**: Finally, SOAs are subject to clearly defined regulatory frameworks since they mostly exist in the corporate context. The design, provision, maintenance, and coupling of services must be compliant with legal frameworks. Therefore, they do not allow for the flexibility that the described new user-services interaction model appears to need.

3 Emerging Technologies and Design Principles enabling the Web of Services

In the following we elaborate on identified synergies the Web 2.0 and the SOA concepts have with regard to the development of a global, user-centric SOA, and describe the technological paradigms and novel design principles that

we have identified as key enablers for the establishment of a Web of Services. We then present a generic model of a global user-centric SOA.

In general, the Web 2.0 and SOA philosophies serve different user needs and exhibit differences with respect to the design of technologies used in real-world applications.

Web 2.0 refers to a perceived second generation of web-based communities and hosted services (such as social-networking sites, wikis, blogs, resources and folksonomies). They aim to mutually maximize collective intelligence and added value for each participant by formalized and dynamic information sharing and creation [8], facilitating collaboration and sharing between users. The later constitutes the Internet's new core as opposed to old core: data.

Recently, however, numerous novel use cases demonstrate the great potential of combining the technologies and principles of Web 2.0 and SOA [3]. One major example of the convergence of the two philosophies is the emergence of a possible user-centric SOA [18], which could enable a real Web of Services vision as described in the previous sections. The general approach is to consider Web accessible resources that can be registered in platforms and thus discovered, tagged (to collect user ratings and provide for folksonomies), and also mashed up (composed and interlinked with the aim of designing new resources) as per user requirements [14].

Focusing this approach a little, user-centric SOA solutions should deliver a mash-up enabled infrastructure to help businesses share and collaborate with the business ecosystem and partners instantly. In doing so, user-centric SOA ideas lead to a mash-up oriented lightweight programming model as the means for end users to collaborate in solving an immediate, specific business problem by blending externalities with private business content and services [17]. In terms of both the ICT technology and the cultural aspects involved in implementing this enterprise collaboration paradigm shift, the way services are discovered, used and managed by end users is fundamental. In this respect, user-service interaction must embrace a number of principles to ensure the widest acceptance by end users [15]. The most important points we have identified when studying several user-centric SOA cases are:

1. *End users must feel fully empowered* and able to serve themselves from available resources that provide them with access to the content and services that they can use to set up their own personalized operating environment in a highly flexible and dynamic way.

2. *Active user participation must be enabled.* End users must be able to contribute new and improved versions of resources, as well as share further knowledge about these resources, their use, and their interrelationships.

3. *Community-based collaborations need to be fostered.* The introduction of a *share, reuse and assembly culture of collaboration* will boost and speed up this process thanks to a network effect.

In addition to these principles, the creation of a sustainable business ecosystem for all actors (including each enterprise end user) in the service value chain also becomes a fundamental principle to comply with. This requires defining the ICT technologies that will effectively support the concept of a marketplace where trustworthy, innovative and flexible business models can be implemented involving both back-end providers, who publish resources (content or application services), as well as brokers/aggregators, who finally make these resources available to end users/consumers.

As a consequence of these principles, users get involved in software development and foster its innovation processes. Business users will benefit from the availability of tailor-made ad-hoc applications that enhance their daily processes [4].

Bearing these principles in mind, we now present a generic model of a user-centric SOA. First, we investigate the new roles of **providers** and **consumers** of Web-based resources (e.g. Web services) and **intermediaries** (referred to as registries in the traditional SOA context) that are expected to prevail in such a next-generation *Web of Services.* Second, we elaborate on the **underlying key technological paradigms** that will effectively support the concepts of a user-centric SOA.

3.1 New stakeholders, a new kind of intermediaries

New kinds of stakeholders will be crucial for the successful emergence and widespread acceptance of the envisioned global user-centric SOA realizing the idea of an IoS, as a huge inter-enterprise service-bus. Rich functionality and information services rather than mere information on isolated Websites are expected to be available. The readily existing Web Services stack foresees publicly available service registries [11].

Based on the UDDI standard, platform-independent, XML-based registries can be set up to allow resource providers worldwide to list the services they provide via the Web.

However, the UDDI standard targets technical experts and only specifies very limited functionality (organized as white, yellow and green pages). Merely UDDI-based registries are inadequate to serve the many ordinary Internet users who require intuitive, human-readable interfaces to identify, rate and finally integrate existing services into their applications. Novel forms of intermediaries [5] are now about to emerge that offer richer resource registry functionality and extend the role of the traditional UDDI-based im-

plementations. They do not only contain references to services, but also offer statistics about resource performance or information about ratings by previous users. The typical resource discovery and utilization process is therefore different from the one common in UDDI-based deployments [13]. The new catalogue sets out the knowledge available within the company for composing resources (including composed services and mash-ups) in a graphical and usable fashion.

Potential consumers may then access the intermediary's Website and search for adequate resources receiving ubiquitous recommendations.

In a future *Web of Services*, Web-based resources will be consumed by both the long tail of private users and enterprises who consume business services. Especially in the professional context, the Internet as of today is mostly only used to gather information, but not to seek and electronically consume mission-critical business services. Due to advancements in the underlying technology and the emergence of novel applications (which we will introduce below), companies will be able to interconnect their business functionality and automate information exchange with suppliers and customers in order to improve their productivity, the quality of their products and to allow for flexibility with regard to changing external requirements.

3.2 Key technological principles

3.2.1 Unified Access to Services through RESTful resources

As a first technological enabler of a user-centric SOA, access to services shall be implemented through *resources* that comply with the Representational State Transfer (REST) architectural style [6]. A resource can be considered as a simple component that owns an URI (Uniform Resource Identifier). The URI that identifies the content or service it represents and responds to the basic http verbs (i.e. get, post, put, and delete), thus wrapping the access to that content or service via a **uniform interface**.

Embracing the REST paradigm implies incorporating the principles of complexity-hiding and uniformity, since it significantly reduces programming efforts and forces both clients and resources to adhere to a common set of supported operations and interface descriptions. In a REST-based SOA, resources and clients exchange resource representations (pre-negotiated data formats) in self-descriptive messages, as part of their interactions through a uniform interface. Because REST targets distributed hypermedia systems, representations also normally contain identifiers for other resources. This way applications can use the identifiers to navigate among related resources (within a service), thus changing (transferring) state. This uniform interface constraint is highly advantageous compared to traditional

SOA. For a client to correctly interact with a conventional Web Service, he or she must understand the specifics of that service's interface contract not only for its operations but also for the data exchanged as part of operation invocations. But for a user to invoke a REST service, all he or she needs to understand is the service's specific data contract, because the operations are uniform for all services.

Some resources are designed to deal with presentation functions, i.e. in response to an http request, they produce output using a data format that can be directly integrated at the user interface level (e.g., they can produce data in a markup language that can be processed and rendered by browsers, using XUL). Others are designed to just produce data that can be processed by other resources, and enable their remix and assembly.

Ultimately, this means that resources are a standard means to offer a uniform front that end users can "see and touch". The conception of contents and services as resources adds a lot to the idea underlying traditional mash-ups (hybrid Web applications or services composed of contents and services from a range of disperse sources), as it provides uniform access to them all.

From this viewpoint, mash-ups can be created by composing and interconnecting different resources with a uniform interface, without having to remix data, code or low-level services [3]. Therefore, the use of this resource abstraction leads to a real paradigm shift driven by a fully functional data modeling.

We propose the following classification of resources:

1. *Data sources* feeding the mash-up application

2. *Operators* transforming the data sources

3. *Gadgets*, which are responsible for providing graphics, and simple and efficient user interaction mechanisms.

3.2.2 Mash-up platforms as a user-oriented service integration enabler

Mash-up platforms are considered as a further key technological enabler of a user-centric SOA. "Mash-up is a Web-based resource, be it content or application functionality, which has been created through reuse and composition of two or more different resources" [17]. Novel platforms ("mash-up platforms") are currently about to emerge that empower users to loosely couple readily available resources, to enrich and compose them into a novel resource, which may eventually be made publicly available again. As a result, meshes of different Web-based resources emerge. These meshes can be considered rather ungoverned since there is no central control entity imposing formal guidelines for reuse and syndication. The possibility of, on the one hand, actively contributing and gaining a reputation by

publishing resources and, on the other, harnessing the aggregated knowledge and intelligence of all platform participants are the core benefits for users.

This new principle of user-driven application design has also been discussed repeatedly in many articles and companies: content-driven mash-up-oriented programming (also referred to as situational programming or instant programming) is a new agile application development paradigm in which users who do not have previous coding skills but do have extensive domain expertise visually assemble and combine off-the-shelf gadgets, i.e. discrete self-contained domain data-oriented components with both development (service and data binding and interconnection) and runtime rendering capabilities [19]. These gadgets represent the basic building blocks for users to use to assemble new services (e.g., SOAP or REST-based lightweight Web Services), data sources (e.g., Atom/RSS feeds) and other gadgets, and render them as necessary to develop the application they need in a very short period of time. The kind of hybrid application that results from applying this new paradigm is often called Enterprise Mash-up. Companies are trying to capitalize on these technologies to construct software and services for relatively short-lived, quick-to-build applications, as is shown in the following section.

The main element of an Enterprise Mash-up is the gadget or graphical human interaction resource. An Enterprise Mash-up then is composed of one or more separate gadgets that behave consistently in the mash-up setting as if they were one strongly cohesive interface [19]. Each gadget will normally be connected to a number of data processing strings or pipes. The execution of these strings or pipes provides the first data needed to update part of or the entire interface that will drive user interaction. This personalizes the application's up-front workflow. From the software viewpoint, the execution of a pipe sequence from the user level defines the order of execution of the different business systems that support the business (workflows, BPMs, Web Services, etc). The resources act as a gateway to these business systems. Therefore, the availability of an expressive pipes definition language, like Yahoo! Pipes, is a key element for systems developed according to this architecture.

3.2.3 Resource Composition through Wiring and Piping

Besides resource-orientation and the idea of graphical mash-up platforms as integration enablers, two concepts, wiring and piping of resources, together represent another important driver to support user creation, innovation and collaboration in a user-centric SOA. When creating a mash-up, users need to visually compose resources and resolve their interdependencies from two different perspectives. On the one hand, they integrate a number of heterogeneous resources defining composed processing data chains/graphs by concatenating successive resources (data sources and operators). This procedure is called *piping composition* (in reference to the *UNIX shell pipeline* concept). On the other hand, they need to define data flows and interdependencies among gadgets to achieve an interconnected graphical mash-up solution able to solve complex functional requirements like a traditional desktop application does. This is called *wiring communication*.

The creation of a pipe involves an a priori unknown number of resources, with a highly variable number of inputs and outputs. However, any resource responds to http (uniform interface) invocations, returning results in XML format. Therefore, it is easy to chain resources so that the output of one resource directly feeds the input of another. Thanks to this, business systems interaction can be easily modeled by the transformation and validation phases of any XML data type (input and output), resulting in *operator resources* (e.g. XSL transformations). Gadgets need to communicate their state to other gadgets inside a specific mash-up to allow the definition of data flows. The definition of these data links should be developed based on solutions that ensure loose coupling like blackboards, publish/subscribe or notification event schemes. This way, flexible data dependencies can be defined. The combination of visual mash-up interfaces and technologies enabling piping and wiring components that are uniformly encapsulated as resources is an approach that brings to fruition the combination of Web 2.0 and traditional SOA concepts. Components can be orchestrated in a loosely coupled fashion, without requiring any programming effort and allowing the long tail of users to build their own applications

3.2.4 Service composition through wire framing

A new technological Internet trend is for users to be able to manage Web-services as visual elements or gadgets, that is, interface controls that manage one or more services and/or resources. These gadgets can be simple or complex, depending on size.

Users can interconnect existing simple gadgets with each other to create more and more complex Web services and their APIs, taking up the idea of storyboard-driven creation and exploiting the power of semantics in the back-end systems. Therefore, users have to deal with just APIs and interfaces, like traditional frames or window dialogues, and interconnect them to create complex graphic use cases instead of using a complex B2B-oriented BPEL-based integration. This visual scheme will be mapped later to services and remote components to create an integrated application in a more flexible and remixable way than a traditional BPEL orchestration. The formal and complex BPEL-based composition used for integration in SOA is now replaced by the
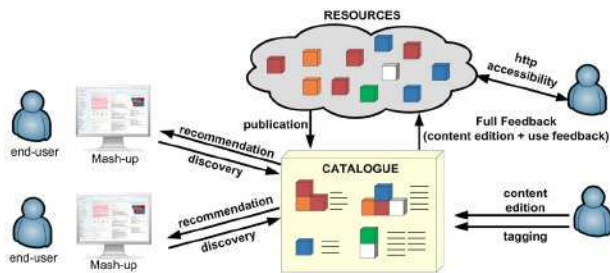
Figure 1. Cataloguing Resources

user-centric techniques of mash-up and wire framing that will boost business innovation as never before.

Applying semantics to this new design and development approach (top-down and user-oriented) would help to bridge the gap between the technical vision of back-end resources and the user functional vision of the front-end [19]. For this purpose, several platforms are beginning to manage complex stateful gadgets and define a complex workflow with a lot of detours from the main branch usually have to deal with human interaction.

3.2.5 Fostering service innovation and global sharing through Web2.0-based service catalogues

If ungoverned and unsorted, the huge number of resources that are envisioned to be available in a user-centric SOA will become unmanageable and thus useless for Internet users. Dedicated catalogues will be required to provide navigation services for users and help them to find out which resources they need to create the mash-ups they want. We propose to address this need by providing a user-generated, "living" catalogue of resources founded on the Web 2.0 vision for user co-production and harnessing of collective intelligence (see Fig. 1). This would provide users with a collaborative semantic Wiki, and tagging and searching-by-recommendation capabilities for editing, remixing and locating resources of their interest. The catalogue sets out the knowledge available within a certain community for composing resources in a graphical and intuitive fashion [7].

The catalogue manages two types of knowledge: (i) a collective pool of resources with capabilities for cloning, modifying and publishing new resources, and (ii) a list of resources with the business processes they support. The catalogue allows users to create complex mash-up solutions by just looking for (or being recommended) "pre-cooked" or off-the-shelf resources and customizing these resources to suit their personal needs, interconnecting resources, and integrating the outcome in their user workspace. "Folksonomies" of user-created tags will emerge, grow as users add information over time and act as important facilitators of a user-centric SOA.

Earlier approaches to service discovery and description like UDDI are not adequate to support human beings in easy resource retrieval and evaluation. By contrast, the exploitation of collective intelligence and user-driven resource categorization is beneficial for users [16]. From an enterprise viewpoint, catalogue accessibility and ease of use encourages all users to continuously improve their operating or desktop environments, taking advantage of the business knowledge generated from colleagues' viewpoints.

3.2.6 Combination in an Enterprise Mash-up Platform Architecture

After elaborating on the shortcomings of traditional SOAs, presenting general technological enablers of a user-centric SOA and examining the change in key SOA principles, this section will present the architecture of a definite possible platform solution based on an enterprise mash-up workflow-oriented enabler that empower its users to co-produce and share *instant applications*, i.e. applications based on composition rather than programming. This platform architecture has been built adhering to the design principles and technologies explained above as an integral solution stretching from users to final services.

The main challenge is to create a new visual programming environment that will facilitate the development of complex front-end gadgets, involving execution of relatively complex business processes that rely on traditional back-end semantic Web services. The approach adopted should be user-centric rather than program-centric. Instead of first building programs that orchestrate available semantic Web services and then trying to figure out how to implement interaction with the users at given points of the process execution flow, programmers will start from the front-end gadgets that the user will see and interact with and then visually establish the connection to back-end Web services by tracing back process execution flows, if necessary. To visually establish this connection, programmers could take an approach similar to the visualization of UML sequence diagrams. In this approach, programmers will visually manage and connect front-end gadgets, screen-flow resources and back-end services and overcome the limitations of current business process engine approaches (based on the traditional SOA vision).

Note that the programming tool to use in this solution should be conceived as a tool that is compatible with existing and future mash-up platforms. Its goal is not to develop the gadget mash-up platform. It is a tool that should enable the development of mash-upable gadgets that rely on screen-flow resources and semantic Web services stored in a catalogue.

Clearly, this particular solution is a subset of a GLOBAL user-centric SOA, specified around the creation of software

based on screen-flows and work-flows that end users could establish via a catalogue of existing semantic web services.

Therefore, users can manage Web services, enterprise boundaries and company legacy as gadgets as shown in Fig. 2.

The enterprise back-end must be "RESTified" to present a uniform interface to standardize disperse heterogeneous data, services and sources. These resources can be tagged by users to acquire the business knowledge present in the collective intelligence generated by users handling the resources. At a higher level in the Global SOA user actions stack, the resources published in semantic catalogues can be discovered/ recommended by users.
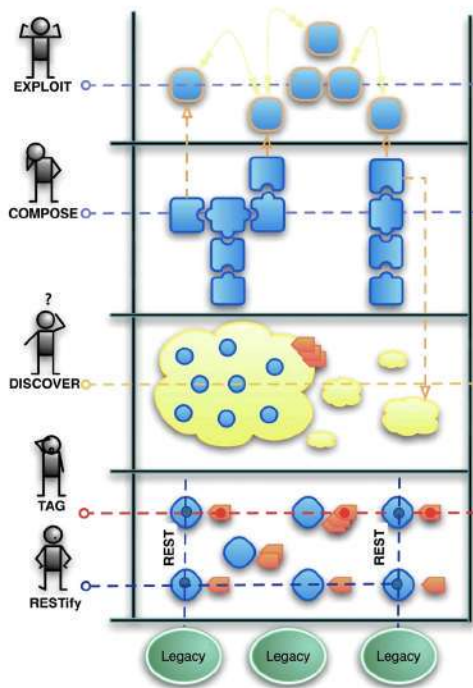

Figure 2. User-centered SOA solution stack

Users then can interconnect existing resources with each other to create increasingly complex web services and their APIs, taking up the idea of resources composition based on storyboard-drive creation. End users can then exploit this complex of creations to achieve enormous improvements in their daily work and innovate by creating/remixing their own business tools.

The power of semantics in the back-end systems is clearly quite astonishing. It is expected that applying semantics to this new, top-down and user-oriented design and development approach would help to bridge the gap between the back-end technological view and the front-end user view. For this purpose, this solution should create all

the mechanisms and tools for proper definition and development of complex gadgets following the current gadget-driven user interface trend.

Therefore, the bases for the proper **conceptualization of complex gadgets need to be defined** from a semantic point of view but this time with a disruptive approach to conceptualize them to the full extent that allows us to semantically compose the functionality provided by the back-end and deliver it to the front-end in a user-oriented way. Building the semantic conceptualization of the logic flow into the gadget itself will relax the strictness of orchestration languages like BPEL.

On the other hand, this solution is concerned with **narrowing the gap between the end user and the Semantic Web**. Nowadays, the Semantic Web is mainly concerned with attaching semantics to data and functionality.

Probably the most interesting contribution to date in the Semantic Web area is semantic Web services. Semantic Web services define the mechanisms for enriching Web services with semantics, enabling the automation of the discovery, composition, orchestration, enactment and management of Web services. In this sense, it would be interesting to complete the life cycle of semantic Web services by laying the foundations for proper interaction between semantic Web services and the end user of those services. Figure 3 illustrates the general logical approach proposed in this solution.
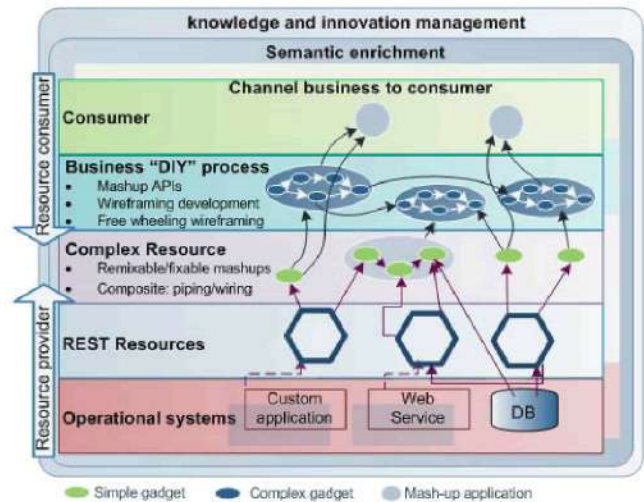

Figure 3. Logical layers in the proposed approach

Based on these ideas, we are considering software development from a top-down perspective as opposed to the conventional bottom-up approach. In this new approach users will play a leading role and the applications will **automat-**

ically adapt to their data and functionality requirements. This new top-down scheme can be summarized as follows:

1. The end user or consumer identifies a need or series of needs in the form of data to be displayed and functionality to be offered. These users will create their own solutions based on the ideas of mash-up and freewheeling wire framing of complex resources and APIs in a do-it-yourself (DIY) business process. This complex of resources has been composed in turn from REST resources by users via a piping and wiring composition of simple resources or by remixing/fixing existing resources. REST resources are a front-end to enterprise legacy, traditional web services, data in enterprise boundaries, etc., resulting from a *"RESTify"* process carried out by enterprises themselves.

2. Users only have to search the gadget registry to find a gadget (or part of one) that meets their needs or more than one gadget that they can put together in a new one of their own composition or creation. The result of this stage is a gadget conceptualization including all the above aspects (user needs satisfied, user capabilities required, interaction models applied, internal logic flow, etc.). This way, a semantic enrichment improves the whole B2C channel of services.

3. Users manage their new gadgets and can publish their improvements to the gadget registry for future reuse, adaptation or specialization. Thanks to this, a knowledge and innovation management is an implicit part of the process, fostering user collaboration and collective intelligence exploitation.

4. Alternatively, users could contribute clarifications, innovations, bugs, enhancements, comments or simply new usages of the components of their mash-ups without actually recomposing, remixing or creating new resources. Increasing the visibility of these business inputs and assuring that they rapidly reach the users of that collective intelligence is vital for boosting business innovation [12]. Therefore, each resource that appears in a mash-up should be associated with standard Web 2.0 communication channels (such as blogging, edition of associated entries in the underlying wiki-based catalogue, etc.). This would enable inputs to be implemented simply and flexibly without users having to create/tailor and publish the solution to be able to contribute their expertise and share it with the enterprise. This way, existing knowledge can be better exploited.

Note that, taking into account the concepts and technologies managed by this idea, the contributions will focus on

the end user without specific background knowledge in semantics, user interfaces, and back-end integration. However, the user-centric approach supports the rapid development and maintenance of applications and information systems.

In summary, this solution aims to define a whole new approach to front-end and back-end integration by developing a new visual programming environment. This new environment will facilitate the development of complex front-end gadgets, involving the execution of relatively complex business processes that rely on back-end semantic Web services and applying the following basic principles:

- As opposed to front-end-oriented mash-up platforms, which are concerned with facilitating retrieval, mashing and utilization of lightweight gadgets, this platform would go a step further and deal with the creation rather than the utilization of such lightweight gadgets. This can significantly improve programmers' operational efficiency.

- Instead of first building programs that orchestrate available semantic Web services with BPEL and then trying to figure out how to implement interaction with the users at given points of the process execution flow, users will start from the front-end gadgets they will see and interact with, and then visually establish the connection to the back-end Web services tracing back the process execution flows, if necessary. The framework will visually establish this connection adopting an approach similar to the visualization of UML sequence diagrams.

- The goal is to build a system that reads the URIs of a number of semantic Web services and is automatically able to interpret and to visualize possible messaging patterns between them for the developer.

- Instead of implementing a choreography from scratch, developers and end users have a visual and efficient interface that allows them to orchestrate services according to their needs and to create a gadget on top that clearly conveys its functionality to human users.

4 Use Case: creation of an enterprise mash-up at Telefónica on the basis of the proposed approach

As a proof of concept, this section sets out the need for a paradigm shift in the user-service interaction paradigm by presenting a real world case study based on Operational Support Systems (OSS) at Telefónica. This scenario is one example of the many solutions that are being developed based on novel user-centric SOA-oriented developments.

This section explains a Telefónica-based mash-up deployed on an existing prototype of the EzWeb platform, where a service-oriented environment is created by visually attaching different gadgets to each other and to the enterprise back-end. This specific enterprise mash-up environment is useful for an end user responsible for the task of testing the status of all systems used by a customer. In the event of a problem in the customer's local telecommunication infrastructure, customer geographical location is a big help for the technician to prepare the visit to the customer's home.

Figure 4. Creation of an enterprise mash-up on the basis of the proposed platform

The zoomed screenshot in Figure 4 depicts the simple scenario extracted from a Telefónica core OSS, which is part of a more general mash-up now deployed at Telefónica as a fully operational environment. The mash-up connects four gadgets: a list of tasks involving customer requests, a customer agenda, a Google map and a network status map. A fully functional environment is created by visually attaching these gadgets to each other and to the enterprise back-end in a wireframing-oriented integration: the agenda gadget will display customer details and have a customer/task selection option, the network map will represent the selected customer's network status and the Google map gadget will display the selected customer's address on a map, as a given task of the list is selected.

It is the end users themselves who develop this "service" to meet their own requirements. They do this on the fly with the help of mash-up enablers like the one proposed in the last section, based on the user-centric approach considered in this paper. Additionally, these systems help users to define and to customize their operational environment, thus improving their use of the entire infrastructure of a real enterprise system in a collaborative and knowledge-driven fashion

5 Related Work

Companies are beginning to focus on people as the entry point to SOA. Thus they need a means to bridge the

gap between people and services and come up against SOA's traditional shortcomings. Consequently, a number of user-centric SOA frameworks are beginning to proliferate. IBM's solution, named *SOA for people*, is noteworthy. It focuses on a portal framework acting as a SOA front-end to maximize people productivity and collaboration.

IBM claims that "with portal and collaboration software, an SOA environment can simplify the way people interact". The increasing interest in this approach is indicative of the current importance of user-centric SOA in the business world.

However, IBM's approach focuses on employing particular Web 2.0-based technologies to deliver a front-end to SOA, instead of reconsidering the whole SOA vision from a user viewpoint. In this paper all SOA elements, ranging from the stakeholders involved in the process to the management, creation and composition of new services and applications following a user-centric approach based on enterprise mash-ups and resource wire-framing are redefined, and SOA registries are completely revisited from a Web 2.0 perspective. Additionally, IBM's approach only considers mash-ups composed of simple gadgets. Our approach goes a step further and also considers the development of complex gadgets based on the *storyboard* notion (an approach used when filming movies, visually identifying the use cases and user needs in the form of a flow of abstract windows, pages or screens). Complex gadgets are now composed of gadgets of more than one page/screen and help to deal with real-life workflows and business processes. This is quite an original approach, and is being developed as part of the FAST initiative, a STREP project being partially funded under the 7th Framework Programme of the European Commission (INFSO-ICT-216048).

Other mash-up platforms are now proliferating. Google IG, Netvibes or Yahoo Pipes are relevant representatives of the idea of composing GUIs from simple gadgets. Nevertheless, they are not called to redefine either the traditional SOA approach or its frameworks (e.g. they do not allow gadget wiring in one and the same GUI and do not consider user interaction with services).

6 Conclusion and Future Trends

The appearance of user-centric approaches to SOA, such as the one proposed in this paper, will be a major step forward, providing solutions to currently hard-to-solve problems in the traditional SOA paradigm. The emergence of such platforms will solve key problems in three different scenarios. Large enterprises may capitalize on faster application development (for what are known as instant applications), a more agile system landscape and the empowerment of their employees to design their own applications that best satisfy their unique requirements, and to share this

knowledge with other employees better than traditional web SOAs. This approach to setting up and managing enterprise IT systems will considerably reduce the backlogs that IT departments often have to cope with and increase operational flexibility, which is crucial in a world with fierce global competition and rapidly changing market demands. It also creates an ecosystem for collaboration and co-creation of new services with other stakeholders to take advantage of a well-known truism, now known as Joy's law, "Innovation happens elsewhere".

On the other hand, Enterprise 2.0 collaboration platforms enable SMEs to find, customize, combine, catalogue, share and finally use applications that exactly meet their individual demands by leveraging the SaaS model, viewed as Utopian from a traditional SOA perspective. Supported by new Web 2.0-based approaches, they can select and combine services and resources hosted by third parties rather than buying a pre-determined, inflexible and potentially heavyweight solution or deal with complex B2B services.

Finally, individuals benefit from a strongly increased capability of personalization and participation. The proposed approach will provide end users with intuitive, unsophisticated IT ways to discover, remix and use those Web-based services that they consider interesting and useful. It will also allow them to participate, swap information with other users and service providers and to actively contribute in a way that encourages extensive use of the resources offered. This speeds up the service innovation pace. Focusing on the "long tail" advanced by Chris Anderson rather than a limited number of sophisticated experts, a user-centric SOA will involve the bulk of private users or small businesses and allow for "customer self-service".

Future work will concentrate on evolving EzWeb, an open source mash-up platform that builds on all the key technical enablers described above and on the proposed model of global user-centric SOA. It will also concentrate on developing and extending a FAST prototype as proof-of-concept. We expect them to become a major hub for the publishing, brokerage, customization and finally the consumption of Web-based resources on a global, cross-organizational scale [18].

Acknowledgment

References

[1] G. F. C. Alonso, H. Kuno, and Machiraju. *Web Services Concepts, Architectures and Applications*. Springer, 2004.

[2] C. Anderson. *The Long Tail, Why the Future of Business is Selling Less of More*. Hyperion, July 2006.

[3] J. C. H. Coveyduc, L. Ostdiek, and J. Reif. *IBM Innovation Factory: An integrated solution for accelerating innovation*. IBM, October 2006.

[4] T. H. Davenport. *Thinking for a Living: How to Get Better Performance and Results from Knowledge Workers*. Harvard Business School Press, Boston, MA, USA, 2005.

[5] P. Evans and T. S. Wurster. *Blown to Bits: How The New Economics of Information Transforms Strategy*. Harvard Business School Press, Boston, MA, USA, 2000.

[6] R. Fielding. *Architectural styles and the design of network-based software architectures*. Ph.D. thesis, University of California, Irvine, 2000.

[7] J. Giles. Internet encyclopedias go head to head. *Nature*, 438(7070):900–990, December 2005.

[8] R. M. M. Hgg, K. Stanoevska-Slabeva, and R. Martignoni. *Overview of business models for Web 2.0 communities, Proceedings of GeNeMe*. Proceedings of GeNeMe, Dresden, 2006.

[9] G. Inc. *Hype Cycle for Software as a Service*. Gartner Research, July 2007.

[10] M. MacKenzie. *OASIS - Reference Model for Service Oriented Architecture 1.0*. OASIS Research, 2006.

[11] A. McAfee. Will web services really transform collaboration. mit sloan management review. *MIT Sloan Management Review*, 46(2), 2005.

[12] A. McAfee. Enterprise 2.0: The dawn of emergent collaboration. *MIT Sloan Management Review*, 47(3):21–28, Spring 2006.

[13] D. North. *Institutions, Institutional change and economic performance*. Cambridge University Press, Cambridge, 1990.

[14] T. O'Reilly. *What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software*. www.oreillynet.com, 2005.

[15] T. O'Reilly and J. Musser. *Web 2.0 Principles and Best Practices*. O'Reilly radar, November 2006.

[16] D. Roman. Web service modeling ontology. *Applied Ontology*, 1(1):77 – 106, 1995.

[17] C. Schroth and O. Christ. *Brave New Web: Emerging Design Principles and Technologies as Enablers of a Global SOA*. In Proceedings of the 2007 IEEE International Conference on Services Computing (SCC 2007), IEEE Computer Society, 2007.

[18] C. Schroth and T. Janner. Web 2.0 and soa: Converging concepts enabling the internet of services. *IEEE IT Professional*, 9(3):36–41, June 2007.

[19] R. Smith. *Enterprise Mashups: An Industry Case Study*. Keynote at the New York PHP Conference and Expo, Manhattan, New York, USA, June 2006.