Service Discovery Suite for User-Centric Service Creation

Carlos Baladrón[1], Javier Aguiar[1], Belén Carro[1], Jürgen Sienel[2], Rubén Trapero[3], Juan Carlos Yelmo[3], Jose María del Álamo[3], Jian Yu[4], Paolo Falcarin[4]

[1]Universidad de Valladolid
Campus Miguel Delibes, Camino del Cementerio s/n, 47011
Valladolid, Spain.
+34 983 423 704
cbalzor@ribera.tel.uva.es, {javagu, belcar}@tel.uva.es

[2]Alcatel-Lucent Deutschland AG
70435 Stuttgart, Germany
Juergen.Sienel@alcatel-lucent.de

[3]DIT, Universidad Politécnica de Madrid
Ciudad Universitaria s/n,
28040 Madrid, Spain
{rubentb, jcyelmo, jmdela}@dit.upm.es

[4]Politecnico di Torino
Corso Duca Degli Abruzzi 24,
Torino, Italy
{jian.yu, paolo.falcarin}@polito.it

ABSTRACT
Two emerging trends are impacting the CIT world: service orientation and user centricity. The OPUCE project aims at developing a platform to mix together both philosophies into an environment where end-users are capable of building highly personalized services integrating Information Technologies and communication capabilities. Such a platform has the potential to become an open marketplace in order for telco operators to implement innovative business models, and as such imposes specific requirements over the discovery mechanisms to help users locate services fitting their needs. This paper presents the discovery suite implemented in the OPUCE platform.

Categories and Subject Descriptors
H.3.5 [**Information Systems**]: Online Information Services.

General Terms
Design.

Keywords
Discovery, service orientation, SOA, user centricity.

1. INTRODUCTION
Two of the most impacting and emerging trends in the Communications and Information Technologies (CIT) are user centricity and service orientation. User centricity aims at putting the end user in the lead role of its CIT experience, not only as a consumer of contents but also as a creator, transforming the networks from the unidirectional provider-to-customer paradigm into a distributed Web-like approach where everyone is both creator and consumer over the infrastructure provided by the operators. Flickr [1], Youtube [2] and the blog phenomenon are examples of this user centric paradigm. This trend is so greatly impacting the CIT world and the society in general that the prestigious person of the year 2006 named by the Time magazine is "You" [3], in reference to the leading role taken by the end user as a provider of content.

Conference'04, Month 1–2, 2004, City, State, Country.

On the other hand the service orientation paradigm and SOA systems are breaking new ground in the enterprise world, allowing inter and intra-organization integration over heterogeneous infrastructures, finally maximizing the interconnection potential of the Internet for the development of business processes, virtual service marketplaces and the Web of services [4]. The Information Technologies community is already embracing this approach, both in the academic [5] [6] and the enterprise world where organizations are migrating their systems towards service oriented architectures. One prominent example is Amazon, who offers a Web Services interface [7] for its internal catalogue, in order to allow interoperability with external organizations.

Even further, service orientation (in the form of Web Services) and user centricity are so impacting that many emerging initiatives are trying to mix them, giving birth to a bunch of service composition environments which let the user build little Web based application by merging two or more Web services. Two examples are Microsoft Popfly [8] and Yahoo! Pipes [9]. These environments offer rich and intuitive graphical interfaces, allowing non-expert individuals to put together and share highly personalized services without any knowledge on programming languages or Information Technologies in general. The fact that two of the most influential CIT companies in the world, Microsoft and Yahoo!, offer their own user oriented service creation environments is indicative of the relevance of the two trends in the future of the Information Technologies world.

On the other hand, the telco world is also moving in this direction but at a slower pace. Telco operators need ways to rapidly create and deliver services in order to open new market possibilities and face a fierce competition, and service orientation is a perfect answer which offers reusability and composability of services so they do not need to be built from scratch.

Several entities, such as OMA, 3GPP and the Parlay Group are promoting initiatives to harmonize and adapt the Web Services standards to the Telco marketplace [10].

But in order to also include the end users in the equation, allowing them to create their own communication services like in the Information Technologies world, the community still needs an environment with graphical interfaces and tools to allow non-expert individuals to manage the complicate steps involved in the creation and deployment of a personalized communication service. The OPUCE project aims at delivering this kind of environment.

Therefore the OPUCE platform will offer an environment where the users will design, create, deploy and enjoy their own integrated CIT services. In this ever changing platform there will be an incremental service offer, with new items being added constantly, and so the Publish/Discovery side of the SOA triangle (Figure 1) becomes more relevant.

Even further, as the OPUCE platform aims at becoming a complete marketplace where operators would be able to implement new business models involving end users directly, there is also a need to complete the SOA triangle with a new flow of information between the service broker and the end users: advertising. If discovery could be considered a *pull* method to retrieve information, advertising is the equivalent *push* method in which the platform sends notifications of potentially interesting new services.

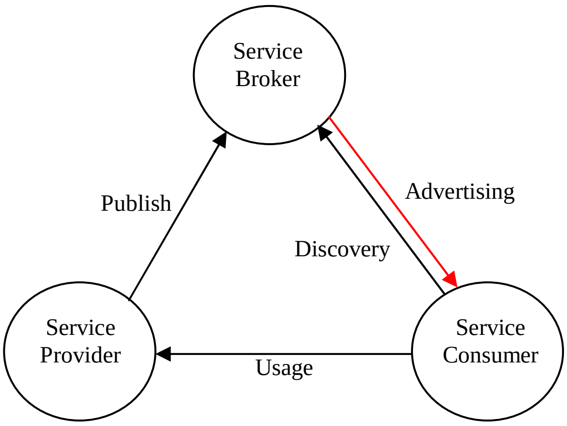

Figure 1. Roles and relations in a SOA.

After this introduction, this paper continues with section 2, where the role of discovery in the lifecycle of an OPUCE service is presented. After that, sections 3, 4 and 5 deal with the platform elements that are involved in the discovery. Section 3 presents the service description, section 4 the service repository and section 5 explains the notification of services as a push method. Finally section 6 exposes the conclusions of the paper.

2. DISCOVERY INSIDE SERVICE LIFECYCLE

The OPUCE project has defined a new paradigm that is focused on the easy end-user service creation. Therefore, it will provide a platform that enables the complete management of the service lifecycle, from the creation and provisioning of the pure components to the enjoying of the service by its subscribers. This lifecycle is depicted in Figure 2.

The service provider is place to set-up the service environment, provide the administrative support functions like service billing, security mechanisms, user and service databases and some generic building blocks, that are the seed into the new self-increasing eco-system similar to the Web 2.0 paradigms, but fully focused on traditional and converged communication services, which will keep the typical assets of the telecommunication operators valid and increase their share in this market.

The role of the service provider includes also the collecting of user related information that can be used to optimize service delivery and service adoption. The role of the end-user can be two-fold. On one hand, he is the one who creates new services by composing existing service elements and building blocks and deploys them on the platform following his own business models, and on the other the end-user that is finally using the services in a personalized way.

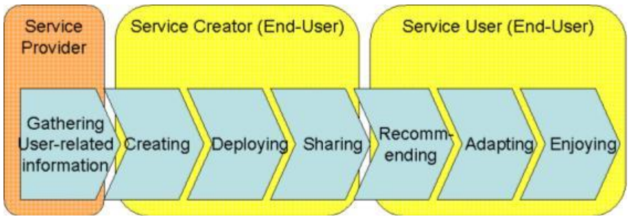

Figure 2. Service Lifecycle in OPUCE

Depending on the role of an end-user, there are several mandatory interactions with the discovery and the recommender system. The creator will need to identify available components and service building blocks (pre-existing compositions) in order to build the service. Since the objective of the OPUCE project is to enable normal end-users without technical background, the graphical service composition tool interfaces with a component and service repository that allows easy selection of the appropriate function. Specific search and retrieval mechanisms have been developed in order to serve this purpose. Semantic descriptions of services and components can be used, based either on the predefined OPUCE ontology, but also using semantic tags that can be given by any user. Such *folksonomies* are especially useful if services have to be deployed in the multilingual context.

After having finalised the service creation and deploying the resulting service to the execution environment, the creator may intend to advertise his new service. Again the semantic annotation becomes important in order to identify the potential clients of such a service. The role of the service advertiser is the matching of the semantic service descriptions to the interests of the user, while the first iteration of the project, that has been finalised in August, a keyword based subscription model has been used, the second iteration will further enhance this system, by taking advantage of user feedback and usage statistics, that will optimise the optimal user perception of the available services. The final goal of this recommender system is to advertise and automatically select only the most interesting services for the target audience.

On the other hand, end users are always able to perform queries against the service repository in order to discover, in any given moment, services matching their requirements. Together with advertising, this completes a rich and flexible bidirectional discovery suite.

As explained above, the OPUCE platform can be used in a cross-national environment. For this reason both service repository and recommender system support the basic translation of the semantic descriptions into the most common European languages. As such the creator of components does not need to take care about providing language adapted components. Services can be easily adapted by the service user and personalised even if they have been developed in a foreign language. This finally will lead to the utmost experience of the community and the adoption of the new OPUCE service creation paradigm.

3. SERVICE DESCRIPTION

3.1 Faceted approach

In an OPUCE based dynamic service ecosystem with continuous activations and withdrawals of user-created services, it is necessary to implement an efficient way to automatically deploy and retire them together with a common convention to describe services, in order to link the information generated by the Service Creation Environment at creation time with the inner part of the platform, which is in charge of processing and distributing that service created information to the correspondent OPUCE platform element. With this information, the platform can deploy, register, activate, notify, and update the systems so as to facilitate the service discovering.

The need of a service description in a service provisioning platform has already been addressed in other Integrated Projects such as SPICE [11] and SecSE [12]. Actually, the basic structure to describe service has been taken from the SecSE project: the faceted approach to describe services [13].

The schema followed in OPUCE considers the service description as an XML document, which makes processing easier. This document consists of different sections, called facets, each describing a different aspect of the service.

Figure 3 depicts the faceted specification followed in OPUCE.

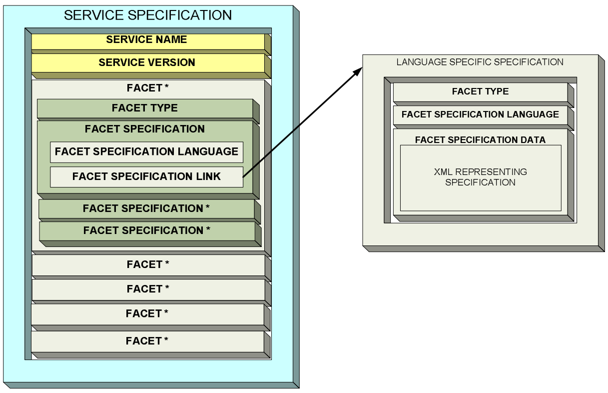

Figure 3. Faceted approach used to describe services.

The service specification consists on two mandatory elements at the beginning of the service description: the name of the service and the version of the service. Next it is the rest of the facets, each describing a different aspect of the service. It is also possible to include nested facets inside “root” facets so as to allow the description of other aspects inside a general one, and also allowing them to be described in a different language. All facets are considered as optional, giving enough flexibility to add, modify or remove facets as it is needed.

Nonetheless, each facet has a common structure. It is necessary a facet type element at the beginning of each facet which contains information about what is described there. Then it appears the facet specification itself (at least one) with information about the language used in this facet and a link to where the specification of the facet is located. Having such a link increases flexibility, allowing clear separation of the facets from the master service description document, and thus simplifying searching and discovering.

At a conceptual level, three groups of facets have been defined, depending on how they are treated by the platform (Figure 4):

- *Functional Facets*, including facets to describe the logic, a set of behaviour constraints to be checked before activating the service, a user-context information to customize the user experience, the semantic description to tag the service in order to make discovery easier, and the interfaces offered externally by the service

- *Non-functional Facets*, including aspects such as the set of users allowed using the service, the service level agreements used also to determine the business model applicable and the quality of the service created.

- *Management Facets*, which include the detailed description of the deployment and provisioning specific tasks to be performed by the platform and information about the customization of the activation date of the service.

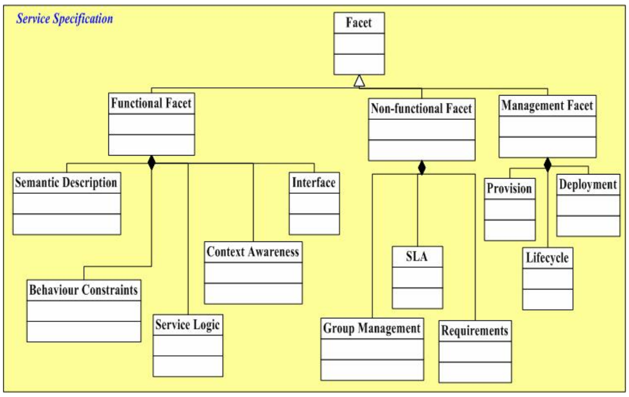

Figure 4. Service Specification in OPUCE

3.2 Role of Facets inside Discovery

At creation time, the user creator inserts a set of tags or keywords which describe what his service will do. For example, if the user is creating a service to send SMSs with the results of the last match of a certain football team, he will probably insert the tags *results, football, notify* and *SMS* in the correspondent label in the Service Creation Environment. The Service Creation Environment will take that information and will automatically generate the semantic description facet.

This information is processed by the service repository in order to facilitate the searching and discovery of new services. For instance, if somebody introduces in the searcher the sentence: *a service to get the results of my football team*, the system will probably return a reference to the aforementioned service.

A similar approach is used to classify the service components. The semantic facet is also mandatory for service components, since it is an essential mechanism to allow searching and discovering of service components at creation time. In this case, a user creator using the Service Creation Environment will look for a service to send SMSs. Thus he will insert the words *send SMS* and the system will look for service components that match the query. The system will return a reference pointing to a messaging component that he can select and add to the OPUCE service flow.

Obviously, the structure of the semantic facet is the same both for the service components and for the OPUCE composed services, and it is also mandatory.

4. SERVICE REPOSITORY

The role of the OPUCE service repository is twofold: first, it is a storage/repository for OPUCE services and components; second, it should provide both basic keyword-based and advanced semantics-aware search functionality to other modules of the platform, including Service Creation Environment, Service Lifecycle Management and Service Advertising.

This crucial requirement drives towards the subdivision of the service repository module into two parts: *Service Storage* and *Service Registry*. Service Storage acts as a common database, taking care of the storage, retrieve, update and deletion of services; while Service Registry manages the metadata of the services in storage. These metadata could be keywords, tags, classifications, relations, and formal ontologies.

Concerning the implementation platform for the repository, common database management systems cannot satisfy the requirements for their weak metadata management capability. But the ebXML Registry standard [14] is proofed to fulfill the main requirement of the OPUCE platform: it is a registry as well as a repository. ebXML Registry has a powerful Registry Information Model (RIM), which provides an upper ontology for defining the service metadata. Most important information models in RIM include:

- Classification Information Model: Defines classes that enable classification.
- Association Information Model: Defines the associations between concepts.
- Service Information Model: Defines classes that enable service description.

Detailed explanation on ebXML Registry RIM can be found in [15].

In fact, the repository functions of ebXML Registry are delegated to a database management system, which works as a plug-in in the registry.

4.1 Mapping Faceted Descriptions into Repository

The mapping of faceted OPUCE service descriptions into the repository can be illustrated by Figure 5.

Take as an example a service called *AutoConference* that contains three facets Master, Signature and Semantics. To put it into the repository, first of all, an *AutoConference Service* registry object is created to represent this service, and put under the classification scheme of *Service*, where *OPUCEService* and *OPUCEComponent* are two classification nodes within. Of course, *AutoConference Service* is classified by *OPUCEService*. The three facets are naturally saved as three repository items, and each corresponds to an extrinsic object in the registry, which serves as its representation in the registry. The linking between a service and its facet extrinsic object is through *ServiceBinding* and *SpecificationLink* classes. ServiceBinding represents technical information on a specific way to access a Service instance. And *SpecificationLink* provides the linkage between a *ServiceBinding* and one of its technical specifications that describes how to employ the service using the *ServiceBinding* [15]. Because a service description does not give information on service bindings, *ServiceBinding* is used just as a placeholder, but each facet of the service is mapped into one *SpecificationLink*, and employed to locate the facet repository item.

4.2 Browsing and Searching of the Repository

Figure 5 depicts a predefined classification scheme for services. This scheme defined one of the browsing structures for the client side user interface, e.g., the Service Creation Environment. In fact, a user can define his own service classification scheme, so that he can browse services based on this customized structure.

Currently, a keywords/tags based search of services is implemented. This is done also based on the mechanism of classification scheme: the *Keyword* scheme and *Tag* scheme are created, and then a specific keyword or tag becomes a node on the scheme. Every time when the user saves a new service in the repository, first its keyword and tag information is retrieved from its semantics facet, and then these keywords and tags are put onto the *Keyword* and *Tag* scheme. Finally this service is connected with these keywords and tags.

In the next iteration of the project it is also planned to support a semantics based search, specifically an OWL [16] based semantic description and search.

5. SERVICE NOTIFICATION

5.1 Service Notification in the OPUCE platform

In the OPUCE platform an additional channel to let end users discover services is implemented in the form of advertising, apart from the traditional browser. The use of this browser represents a pull method where the active role is performed by the user. But several characteristics of the OPUCE platform impose additional requirements over the discovery process:

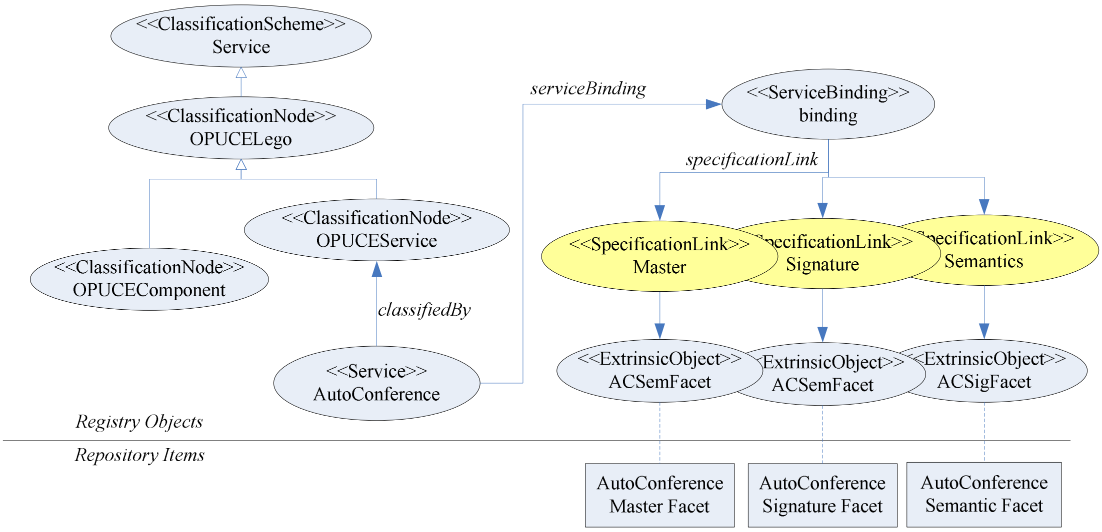

Figure 5. Structure of the AutoConference service in the repository

- The OPUCE platform pretends to be not only an environment for creation and execution of services, but also a complete marketplace where new business models involving end users could be implemented. These business models would eventually imply some form of advertising, with the creator of the service pushing information towards appropriate end users.

- In a user-driven environment like the OPUCE platform new services are being constantly created. An end user waiting for the development of a specific service should check frequently the browser. Additionally improved or updated versions of already existing services could also be developed, and it is necessary to let their users know about it without forcing them to check periodically.

- The OPUCE platform provides support for social networks, including communities of users with similar interests and easy access of each user to his social sphere (family and friends). In order to allow easy communications inside communities and social spheres, an asynchronous sharing platform is needed to exchange services.

- The OPUCE platform could eventually require to proactively contact end users about deployment and management aspects of services, such as warnings about withdrawal of services in advance.

In order to cover these four requirements, the OPUCE platform implements a context aware advertising/notification system which actively pushes notifications towards end users. Its purposes are:

- Let service creators advertise their products towards specific audiences categorized by topics of interest.

- Let end users subscribe to keywords of interests in order to receive notifications about services related to those topics, and personalized recommendations intelligently generated from matching of service descriptions and user profiles.

- Allow end users to notify about interesting services to their communities and social spheres.

- Allow the platform to push service-related notifications.

This service advertising system is also designed to provide two very convenient characteristics in order to make notifications as less intrusive as possible. First, it is able to automatically determine the potentially interested users in a given service by means of a matching between the description of the service and the profile of the users. And second, it is context aware. Users are able to specify in their profiles specific preferences about the way in which they want to receive notifications, even including context variables such as "if my presence status is online and my location is my home, send me an instant message; if my presence status is online and my location is my office, send me an email; if my presence status is offline and time is between 8 a.m. and 10 p.m., send me an SMS, and if it is between 10 p.m. and 8 a.m., wait until 8 a.m.".

With these two features, the advertising system is able to push highly focused notifications using the preferred means of each user, and thus minimizing intrusiveness and costs.

5.2 Service Advertiser

Figure 6 depicts a high-level scheme of the Service Advertiser System.

The event handler subsystem exposes generic interfaces to allow issuing of notification orders by other modules of the OPUCE platform. This includes advertisements, notifications about status of services to their subscribers, platform generated recommendations and messages between community and social network members to share services. While conceptually different, they are treated like generic notifications inside the Service Advertiser module.

The Input Stage receives the event from the Event Handler. If the notification includes as target a group of users of any kind (a community or subscribers of a given keyword), an integrated matcher contacts the User Profile Repository to retrieve the specific target users related to that group. In any case, the User Profile Repository is accessed to retrieve the spam filter preferences of each user and block any unsuitable message. This stage then creates individualized notifications, formats them, and inserts them into the internal Notification Database.

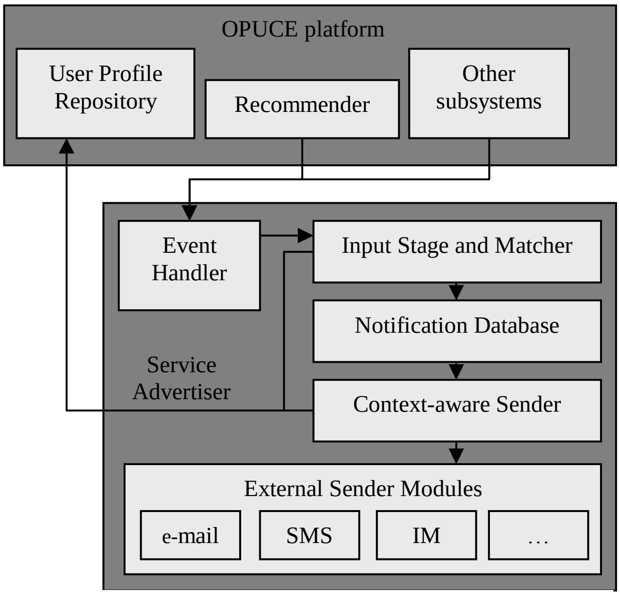

Figure 6. Service Advertiser Architecture.

This database contains all notifications received by the users, and is accessible through the Web portal of the OPUCE platform, so this could be considered an additional notification means. Users are able to manage notifications from this portal, including their deletion, in an e-mail client fashion.

Finally, the Context-Aware Sender subsystem retrieves notifications from the database and in order to determine the appropriate means of notification, accesses the User Profile Repository to get each specific user's preferences and his context. After a comparison between both, the message is forwarded to one of the external sender modules, or left unsent until the appropriate conditions are met. In the former case, the message is deleted from the database or just marked as "sent", depending on the preferences of each specific user (if, for instance, he wants to keep track of all notifications received to access them through the Web portal).

6. CONCLUSIONS

The OPUCE platform is an environment where end users are able to create their own integrated CIT services, both with information and communication capabilities, and thus represents the telco response to the Web 2.0 paradigm, specifically implementing service orientation and user centricity.

In an ever changing environment a flexible and robust discovery system is needed, in order to let users fully leverage the great number of dynamic items inserted in the platform. OPUCE includes a bidirectional discovery suite, comprised by a Service Repository with browsing capabilities and a context-aware Service Advertising system. This system allows for a non-intrusive, highly-focused discovery/advertising experience.

But development of the system continues in order to enrich this discovery suite with fully semantic capabilities. The adoption of fully semantic service descriptions will allow an even narrower search and notification. Users will then be able to perform semantic queries to retrieve services exactly matching their needs, to subscribe to notifications about services using semantic rules and to receive recommendations based on semantic analysis of profiles and usage history.

7. ACKNOWLEDGMENTS

The work presented in this paper is executed as part of the OPUCE project and partly funded by the European Union under contract IST-034101. OPUCE is an Integrated Project of the 6th Framework Programme, Priority IST.

8. REFERENCES

[1] Flickr, http://www.flickr.com

[2] YouTube, http://www.youtube.com

[3] Person of the year: You, *Time Magazine*, European Edition, Vol. 168, No. 27/28.

[4] O'Reilly, T., What is Web 2.0, http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

[5] Petinot, Y., Giles, C. L., Bhatnagar, V., Teregowda, P. B., Han, H. and Councill, I., A service-oriented architecture for digital libraries. In *ICSOC '04: Proceedings of the 2nd international conference on Service oriented computing* 2004.

[6] Zhang, J., Chung, J.-Y. and Chang, C. K. Migration to web services oriented architecture: a case study. In *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*, 2004.

[7] Amazon Web Services, http://aws.amazon.com

[8] Microsoft Popfly, http://www.popfly.ms

[9] Yahoo! Pipes, http://pipes.yahoo.com

[10] Kimbler, K., Exploring New Business Models with Parlay X – Telecoms Web Services, *Wireless Technology*, January 2004.

[11] SPICE Website: *http://www.ist-spice.org*

[12] SeCSE Website: *http://secse.eng.it*

[13] Sawyer, P., Hutchison, J., Walkerdine, J. and Sommerville, I.: Faceted Service Specification. Proceedings of Workshop on Service-Oriented Computing Requirements (SOCCER'05). Paris, 2005.

[14] OASIS. ebXML Registry, www.oasis-open.org/committees/regrep/

[15] OASIS. ebXML Registry Information Model Version 3.0, 2005.

[16] W3C. Web Ontology Language, http://www.w3.org/2004/OWL/