



HAL
open science

Service oriented computing from the user perspective

Nassim Laga, Emmanuel Bertin, Noel Crespi

► **To cite this version:**

Nassim Laga, Emmanuel Bertin, Noel Crespi. Service oriented computing from the user perspective. KuVS 2010: 3rd Kommunikation und Verteilte Systeme (KuVS) Fachgespräch meeting, Oct 2010, Berlin, Germany. hal-01308988

HAL Id: hal-01308988

<https://hal.science/hal-01308988v1>

Submitted on 28 Apr 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Service Oriented Computing from the User Perspective

Nassim Laga, Emmanuel Bertin

Orange Labs
France Telecom R&D, 42, rue des Coutures,
14000 Caen France

{nassim.laga, emmanuel.bertin}@orange-ftgroup.com

Noel Crespi

Institut Telecom, Telecom SudParis,
9 rue Charles Fourier, 91011, Evry Cedex, France
noel.crespi@it-sudparis.eu

Abstract— The last decade has attracted lot of research work in Service-Oriented Computing (SOC), giving raise to standardized architectures, protocols, and technologies that enable developers to easily expose and reuse services. However, these technologies do not fully consider the users as potential actors in the creation of services based on existing ones, as advocated in Web 2.0 paradigm. In this presentation, we first summarize SOC and its intrinsic SOA paradigm. Second, we propose a new approach based on Widgets. We propose the Widget-Oriented Architecture (WOA); a new paradigm to enable a user-centric service reuse. Third, we show how the WOA is applied in two SOA application fields, namely service composition and business process management.

I. INTRODUCTION

Web 2.0, current “Web era”, is characterized by an increasing number of services, user participation in content creation, user centered design, information sharing, interoperability through standards, and rich user interface technologies. These characteristics have really revolutionized both software engineering methods and users interaction with software features. First, software features are no longer packaged as a single application; instead, they are split into and published as Web services in order to promote cross-network and cross-organizations sharing, collaboration, reusability, and integration. This is known as Service-Oriented Computing (SOC) [1]. Second, user interaction with software features has changed from a passive role (pure consumer) to an active role (consumer and producer of content). Typical examples of this phenomenon are Wikipedia¹ and YouTube² web sites where their success is completely dependent on the quality and the quantity of content generated by users.

These two characteristics of the current Web platform, namely software fragmentation into Web services and user participation in content creation, have encouraged the idea of enabling the user to create and publish new services by composing existing ones. This is known as user service creation. The goal is to really reach the user self-service as promised by Web 2.0 paradigm. Consequently, several approaches based on the SOA have been proposed. Some of them are automatic such natural language based service

composition, and others require additional investment from the user by manually and graphically chaining services. However, being initially conceived for developers, SOA based composition tools have not succeeded to really enable, users, without computing skills, to reuse services in the creation of more sophisticated ones.

In this presentation, we first propose the Widget-Oriented Architecture (WOA) paradigm to tackle the limitations of SOA. Based on the concept of Widget, WOA is a new computing paradigm that aims to be more centered on the user compared to SOA. A Widget is a client side Web application that provides access to one or several functionalities of a service implementation. Second, we show how this new paradigm is used within two SOA application fields, namely service composition and business process management (BPM).

II. WIDGET ORIENTED ARCHITECTURE (WOA) PARADIGM

As we illustrated in Figure 1, the WOA includes five roles: a Widget developer, a Widget provider, a common Widget registry, the user, and the Widget client which embeds the necessary mechanisms that enable the user to combine the Widgets.

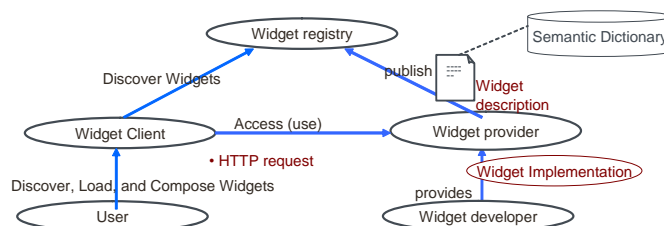


Figure 1. WOA involved roles.

Such as SOA, the WOA is characterized by a set of principles that must be performed by the different roles. The most important ones are related to the developer/provider of the Widgets and the Widget Client, which are summarized below.

A. Widget provider/developer principles

There are three important WOA principles related to the Widget developer/provider roles. First, developers must expose their software features as a set of Widgets. Figure 2 illustrates how applications are exposed as a set of Widgets. Second, such as SOA, WOA requires from Widget providers to define their Widgets in term of the functionalities they provide, and their non-functional parameters (including the provider, the version,

¹ Wikipedia, http://en.wikipedia.org/wiki/Main_Page, accessed Sept. 8th, 2010

² YouTube, <http://www.youtube.com/>, accessed Sept. 8th, 2010

the SLA, and the Qos...etc). Third, the Widget UI must be semantically annotated to enable the Widget client to retrieve resources generated by the Widget.

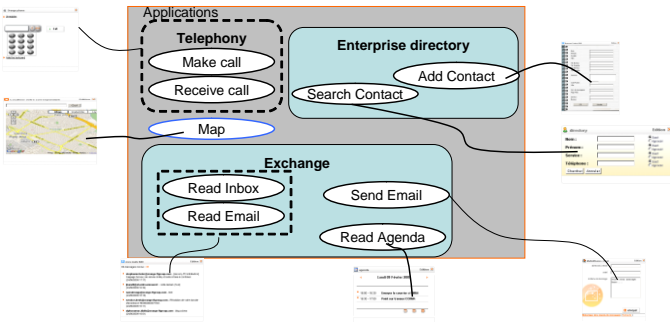


Figure 2. Widget-based exposure of software features.

B. Widget Client principles

The Widget client is a software application through which the user consumes Widgets. Followings are the most important WOA principles related to this role.

First, the Widget client must play a role of a user service environment and a composition framework at the same time. In other words, in WOA paradigm, the composition framework should be the same environment as the daily service environment of the user. The user should not have two separate environments, one for composing services and one for using the composition.

Second, the Widget client must embed front-end reusability and composition tools addressed for users. The reusability and composition must be performed at the UI level. In other words, the intelligence that enables the composition of Widgets between them must reside at the Widget client level. Figure 3 illustrates a Widget client we have defined and implemented [2, 3]. It shows Widgets loaded by the user and combined automatically.

Third, the Widget client must enable the user to personalize his environment by loading only the Widgets he needs.

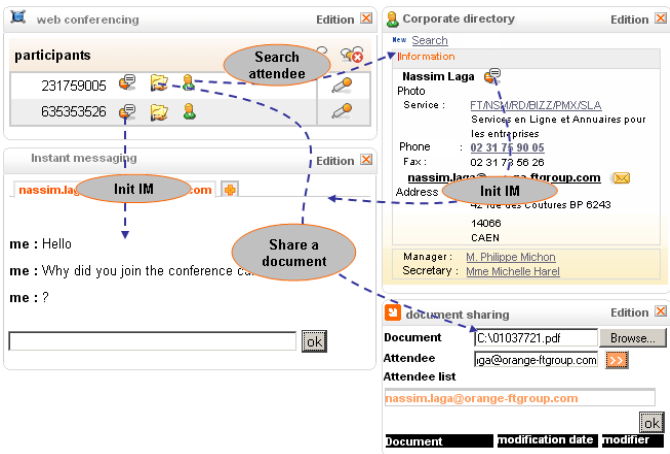


Figure 3. Widget-based exposure of software features.

III. WOA IN SOA APPLICATION FIELDS

We applied the WOA to two SOA application fields: service composition and business process management.

A. Service Composition

In [4], we classified service composition tools into three categories: static, semi-automatic, and automatic composition tools. Static composition aims to provide technologies to developers to perform reusability and composition; Semi-automatic composition aims to enable users to create a composite service by graphically chaining ready-made services; and automatic composition aims to enable users to generate services by expressing their needs using their natural language.

The WOA enhances the static composition by providing to users the capability of personalizing a created composite service. It enhances the semi-automatic composition by simplifying the process of creating a composite service; even ordinary users can now create their own composite service. Finally, the WOA enhances the automatic composition by enabling users to modify a created composite service; this is pertinent as automatic composition still suffers of its inaccuracy due to the ambiguity of natural language.

B. Business Process Management (BPM)

As we detailed in [3], enabling users to combine Widgets provide new opportunities to enhance the management of business processes; which currently suffer from their heterogeneity and dynamicity, which complicates significantly business analysts and developers tasks from one hand, and reduce the automation level from another hand. In the presentation, we will detail our approach based on WOA to tackle these limitations. The idea is to alleviate the business analysts and developers task by harnessing the capability of users to compose Widgets. Basically, we model each business process as a combination of a common part and a user dependent part [2]. In addition of being common to all users, the common part is also static (does not change frequently within the enterprise). The user dependent part is however heterogeneous (user dependent), and dynamic. Thus, by combining WOA and SOA, the common part is modeled and developed respectively by business analysts and developers, and the user dependent part is automated by the users themselves using the WOA.

This combination enables business analysts to focus only on high level view and the common parts of business processes, while ensuring a high automation for users.

IV. CONCLUSION

The introduction of the WOA does not aim to replace the SOA; instead, the two paradigms are complementary and necessary in an end-to-end service environment. It is important to have the two layers. The approach we advocate can be summarized in the following items:

- developers create web services (SOA),
- developers create the corresponding Widgets (SOA and WOA),

- developers compose web services, create the corresponding Widget, and optionally use static composition of Widgets (SOA and WOA),
- users compose Widgets (WOA).

REFERENCES

- [1] Papazoglou, M.P. and Georgakopoulos, D. 2003. Service-Oriented Computing. Communications of the ACM. Vol.46, no10, pp.25-28. October 2003.
- [2] Laga, N., Bertin, E., Crespi, N. 2009. Building a user friendly service dashboard: Automatic and non-intrusive chaining between widgets. In proceedings of the 2009 Congress on Services – I. SERVICES. IEEE Computer Society, Washington, DC, 484-491. Los Angeles, USA. July 2009.
- [3] Laga, N., Bertin, E., Crespi, N. 2010. Business Process Personalization through Web Widgets. In proceedings of IEEE International Conference on Web Services. ICWS 2010. pp. 551-558. Miami, USA. July 2010.
- [4] Laga, N., Bertin, E., Crespi, N. 2008. User-centric Services and Service Composition, a Survey. In proceedings of the 32nd Annual IEEE Software Engineering Workshop. SEW. IEEE Computer Society, Washington, DC, 3-9. Kassandra, Greece. October, 2008.