Enabling Innovation across the Enterprise through Mashup-oriented Collaboration Environments

Javier Soriano[1], David Lizcano[1],

[1]Universidad Politécnica de Madrid
Campus Montegancedo s/n
28660 Boadilla del Monte, Madrid, Spain
{jsoriano, dlizcano}@fi.upm.es

Miguel A. Cañas[2], Marcos Reyes[2], and Juan J. Hierro[2]

[2]Telefónica I+D
Emilio Vargas 6
28043 Madrid, Spain
{mac, mru, jhierro}@tid.es

Abstract: Nowadays enterprise collaboration is becoming essential for valuable innovation and competitive advantage. This collaboration must be brought a step forward from technical collaboration till collective smart exploitation of global intelligence. The Internet of Future is expected to be composed of a mesh of interoperable Web Services accessed from all over the Web. This approach has not yet caught on since a global user-service interaction is still an open issue. This paper states our vision with regard to the next generation front-end web technology that will enable integrated access to services, contents and things in the Future Internet. This approach will enable the massive deployment of services over Internet in a user-centric fashion. Having this in mind, the rationale behind EzWeb, a reference architecture and implementation of an open Enterprise 2.0 Collaboration Platform that empower its users to co-produce and share *instant applications* is presented.
Keywords: Enterprise Collaboration, Enterprise 2.0, Web 2.0, Knowledge Work, Internet of Future, Mashup, Innovation.

1. Introduction

As global market opportunities and competition increase, collaboration is becoming more and more essential for improving productivity and accelerating innovation at the personal, team, group, enterprise and business coalition levels. Many enterprise collaboration platforms have already been developed and successfully deployed in both large, and small- and medium-sized enterprises (SMEs). Used in the context of a carefully engineered enterprise collaboration strategy, these platforms provide a wealth of collaborative services for knowledge workers [1].

Enterprise collaboration has recently come to benefit from the emergence of an enterprise-oriented specialization of the Web 2.0 vision, commonly referred to as Enterprise 2.0 [2]. Enterprise 2.0 provides enterprises with new models and tools for emergent collaboration and co-creation. Enterprise collaboration is thus being enhanced by virtual communities that leverage social linking and tagging tools (like tools for social networking, social bookmarking and social search), user-contributed content management platforms (like enterprise Wikis, blogs and forums), tools that leverage user opinions (like tools supporting comments and voting), subscription-based information distribution tools (like Enterprise RSS feeds), etc.

These digital platforms are already popular on the Internet, where they are collectively labeled “Web 2.0” technologies [3]. Now though a number of Enterprise 2.0-based collaboration platforms are beginning to proliferate [4,5], aimed at providing enterprises with specialized subsets of these out-of-the box capabilities. These new collaboration platforms provide enterprises with an ecosystem of *knowledge workers*(Note 1) who collaborate to develop capabilities by collectively generating, sharing and refining information and business knowledge. Enterprise 2.0 collaboration enables firms to leverage desirable Web 2.0 attributes, including harnessing collective intelligence and architecture by participation.

In any case, the design of both traditional and Enterprise 2.0-based enterprise collaboration architectures has focused primarily on creating a structure that supports common processes and stores information to assure that it is easy to find, reliably available, and backed up. They all have been conceived under the premise that teams need to focus on their core business rather than IT issues. The entire operating environment has therefore been traditionally subordinated to IT departments.

Nevertheless, this latter approach has proved to have a number of collaboration-related drawbacks that slow down the pace of innovation. Knowledge workers are thoroughly acquainted with routine procedures and are capable of extracting automatic behavior, suggesting improvements on the IT systems they use through their operating environment and, more importantly, innovating new operating procedures. Operational innovation is an essential requirement in today’s competitiveness-driven business markets.

As the applications needed to make such improvements are immediate and fleeting (they respond to individual unforeseen requirements, the need for which emerges spontaneously and lasts less than 6-12 months), there is a whole bunch of them not being written today because they are not affordable due to time-to-market constraints and/or because there is no justification for IT investment. This paper advocates that enterprise collaboration must evolve towards a new paradigm in which knowledge workers (without previous programming skills, but with thorough business knowledge) are considered as co-producers not only of information, but also of software services and applications that promote specific competitive advantages and/or meet their immediate needs, without involving IT departments, and share the solution with the remainder of the organization.

This will lead at last to an ecosystem of knowledge workers collaborating to develop capabilities and innovate operating procedures by remixing and integrating available services.

The remainder of the paper is organized as follows. Section 2 sets out the need for a paradigm shift in enterprise collaboration by presenting a real world case study based on Operational Support Systems (OSS) at Telefónica (Note 2). Section 3 then introduces the idea of *enterprise mashup*, which is key to the implementation of the paradigm shift. The suitability of mashups for this purpose is further developed in section 4 through a real world EzWeb Enterprise Mashup at Telefónica OSS. Principles underlying a mashup-enabled Enterprise 2.0 collaboration architecture are then introduced in section 5, and section 6 deals with how these principles have influenced EzWeb design. Section 7 takes a step further and uncovers EzWeb's main architectural components and their interrelationships. The paper's conclusions are set out in section 8.

2. A Case Study: Operational Support Systems (OSS) at Telefónica

Dispatching and trouble-ticketing systems are the key component of software systems processing user queries and claims. Essentially, they are conceptually generic. Being general-purpose systems, they are applicable to a broad spectrum of possible scenarios, ranging from electronic dispatching of administrative records within the Public Administrations to the management of incidents and claims by business services end-users.

These systems have important technological and usability requirements. The first requirement is need for Web access to the system, both by the end-user and by the many workers involved in collaborative dispatching. Second, facilities are required to configure and personalize operating environments to minimize the impact of the implementation, deployment and maintenance of the access applications, subject to constant changes in such systems.

The Telefónica group IT managers noticed that the knowledge workers involved in dispatching incidents gradually stopped using the official corporate applications and started to use small ad hoc Web micro-applications developed by themselves or by other savvy-tech knowledge workers. They then began gradually to use tools and technologies (like PHP, Phyton, XHTML, etc) to rapidly develop share, remix and modify small applications on top of the corporate back-ends. Thanks to the employees' thorough business knowledge, those applications significantly improved the company's system interfaces, increased employee productivity and tailored the systems to the company's real needs.

Early on systems like these were not usually part of and interfered with the set of official, company-supported systems. Therefore, their deployment might even be persecuted. This stand was gradually relaxed, because of user pressure, and operating centers refused to have useful systems withdrawn in exchange for other that were not tailored to their needs. This led to the acceptance of such systems within the organization. However, the high level of programming skills required to develop them prevented the number of ad hoc Web applications from growing at the rate required by knowledge workers.

The enormous impact that these systems were having at the level of overall user satisfaction converted them into strategic and key systems for the organization's smooth operation. However, the immediateness and fleetingness of these applications, together with their heavy personalization requirements, was making them unaffordable for IT departments too, due to the extreme time-to-market restrictions they where imposing and/or the absence of justification for IT investment.

At this point, Telefónica started to feel the inevitable need to involve the knowledge workers responsible for managing vital affairs for the customer actively in business innovation processes, as co-producers of a certain kind of software services and applications.

3. Enterprise Mashups: Collaboration as Co-Production

Content-driven mashup-oriented programming (a.k.a. situational programming or instant programming) [6] is a new agile application development paradigm in which knowledge workers, 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.

These gadgets represent the basic building blocks for knowledge workers to assemble new services (e.g. SOAP or REST-based lightweight Web services), data sources (e.g. Atom/RSS feeds) and other gadgets, and to render them as necessary to develop the application they need in a very short time. The kind of hybrid application that results from applying this new paradigm is often called *enterprise mashup* (a.k.a. situational application or instant application).

Companies are trying to capitalize on these technologies with software and services for relatively short-lived, quick-to-build applications, as is shown in the following section.

4. A Real World EzWeb Enterprise Mashup at Telefónica

The zoomed screenshot in Fig.1 depicts a simple scenario extracted from a Telefónica core OSS environment like the one described above, which is part of a more general mashup now deployed at Telefónica as a fully operational environment. The mashup 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 backend: 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. This enterprise mashup environment is useful for a knowledge worker 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.

It is knowledge workers themselves who develop this "service", and do it on the fly with the help of mashup enablers like those considered in EzWeb (e.g. iGoogle for the mashup being considered here). 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.

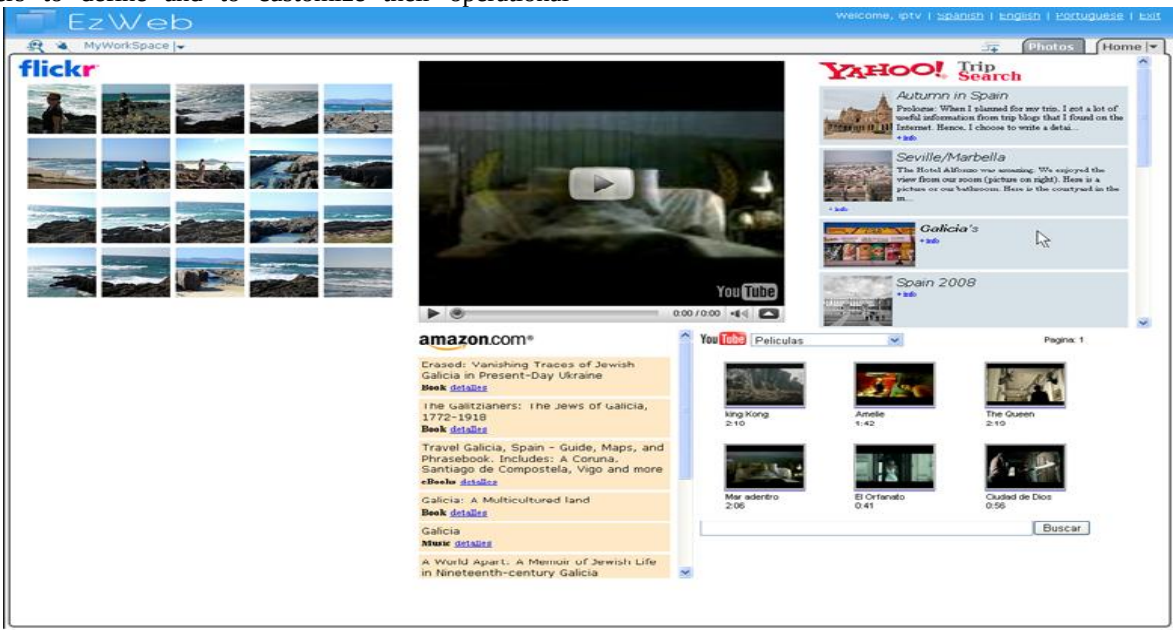

Fig. 1. A real world EzWeb enterprise mashup currently deployed at Telefónica OSS as a fully operational environment

5. Principles Underlying a Mashup-enabled Enterprise 2.0 Collaboration Architecture

EzWeb delivers a mashup-enabled infrastructure to help businesses share and collaborate with the business ecosystem and partners instantly. In doing so, EzWeb is one of the first enterprise collaboration architectures to introduce the mashup-oriented lightweight programming model as the means for knowledge workers to collaborate in solving an immediate, specific business problem by blending externalities with private business content and services.

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 knowledge workers is fundamental. In this respect, user-service interaction must embrace a number of principles to ensure the widest acceptance by knowledge workers. The most important that we have identified when conceiving EzWeb are:

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

(ii) *Active user participation must to be enabled.* Knowledge workers 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.

(iii) *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.

(iv) *IT departments need to embrace the Software as a Service (SaaS) model* as an effective software-delivery mechanism. This will change the department's focus from deploying and supporting applications to managing the services that those applications provide. Knowledge workers will now extend and improve these services in a collaborative fashion to exploit their extensive domain expertise and their thorough business knowledge.

6. EzWeb Design Principles

Taking into account the above premises, three basic design principles have been applied to define the basic EzWeb-associated architecture:

(i) *Knowledge workers can only deal with things they can "see and touch" at the user interface level*, i.e. things that can be visually represented, clicked with the mouse pointer and/or typed, and natural language guidance on their use. They cannot see service interfaces at the service layer or process definitions at the process layer. Service interface definitions in WSDL make no sense to them. There is an urgent need to put "face" to services.

(ii) *The best solutions are usually the simplest ones.* Actually, *simplicity*, *usability* and *flexibility* have always been the main technological evolution engines in IT. The lower the access barrier to a given technology is, the bigger the programmer and user communities working with it are. A Web browser, a few clicks and some typing must suffice for any user interaction with the platform; no programming skills should be required.

(iii) *Knowledge work emergence must be fostered with the idea that software is freedom.* Thus, software must be showed to knowledge workers as egalitarian, indifferent to formal organizational identities, free of up-front workflow, and accepting many types of data through a uniform interface. The objective is not to impose any preconceived notion on knowledge workers about how

work should proceed or how output should be structured. Instead, the aim is to let these aspects of knowledge work emerge.

7. Uncovering EzWeb Architecture

EzWeb Architecture has been conceived based on these basic design principles. Its main building blocks and their interrelationships are depicted in Fig. 2a and Fig.2b and are further explained in the following subsections.

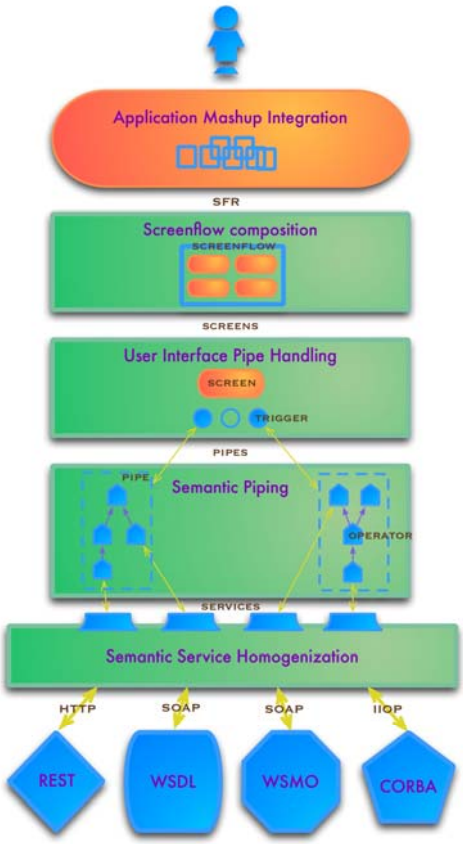

Fig. 2a EzWeb Architecture

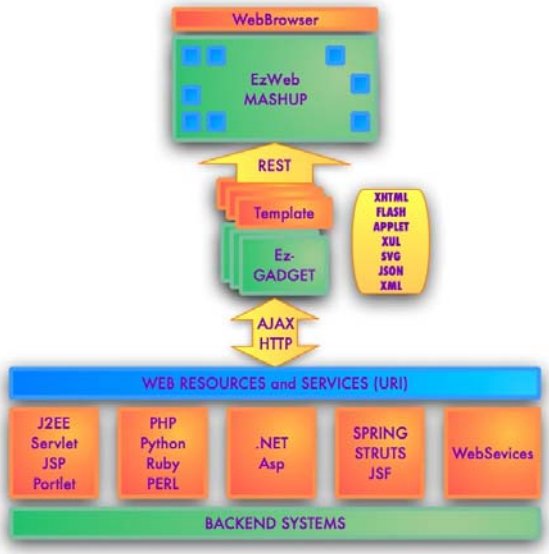

Fig. 2b EzWeb's Connectivity Architecture

7.1 Unified Access to Services through Resources

Usually, enterprise front-ends provide knowledge workers with data from legacy IT through applications built on SOA (Service Oriented Architectures). EzWeb has revisited SOA to focus on end-users as service clients by providing a RESTFul [7] data front-end, transforming this enterprise legacy into a uniform layer of human-accessible resources, as shown at the bottom of Fig. 2.

The architecture aims to allow knowledge workers to hack (remix, compose and feed) these resources through a friendly user interface. This interface enables end-users to access (using social search) a collaborative catalogue of resources and BPM solutions. They can even semantically enrich these resources through tagging and Wiki-based collaborative edition in a common mashup enabler tool. These four architectural blocks (catalogue, tagging/searching, Wiki and mashup enabler tools) represent a powerful *knowledge grid* that fosters the emergence of collective intelligence. They are further described in the remaining sections.

Based on the above three design principles, the notion of *resource* emerges as the basic concept around which all the EzWeb architecture is organized. Resources are simple components designed following the REST (Representational State Transfer) architectural style [7]. A resource owns an URI (Uniform Resource Identifier) 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 through a uniform interface.

Embracing the REST paradigm will imply incorporating the principles of complexity-hiding and uniformity since this paradigm significantly reduces programming efforts and forces both clients and resources to adhere to a common set of supported operations and interface descriptions.

Resources handle basic http requests by gathering and processing data (content or output from services), and delivering output to other resources or directly to the knowledge worker's browser. Some 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). 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 mashups (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, the mashups will be created by composing and interconnecting these resources with a uniform interface, without it being necessary to remix data, code or low-level services as it is with heterogeneous interfaces.

Users are empowered by allowing them to choose the best resources to meet their real needs from a global catalogue of resources. They will use whichever resources they like to create ad hoc instant solutions following the mashup-inherent DIY ("do it yourself") philosophy. This process is carried out in a fully visual manner. The main goals are usability and

simplicity. Therefore, IT expertise would no longer be required to meet common user requirements.

The composition of user interfaces according to the proposed architecture (see Fig. 1) is based on conceiving all business items as resources furnished with a constrained, uniform and simple access interface. This approach encourages resources mashup, eluding the need to remix data, code or low-level services. This solves the problem of having to integrate different and heterogeneous interfaces.

To assure that the user is finally involved in the creation of business mashup-based solutions, classification of the resources is proposed to provide a generic, intuitive and collaborative business solutions development framework:

(i) *Data sources* feeding the mashup application;
(ii) *Operators* transforming the data sources at whim; and
(iii) *Gadgets*, which are responsible for providing graphics, simple and efficient user interaction mechanisms.

7.2 Resource Composition through Wiring and Piping

As is shown in Fig. 3, when co-producing a mashup, a knowledge worker needs to visually compose resources and resolve their interdependencies from two different perspectives. On the one hand he or she interconnects a number of heterogeneous data sources to create gadgets. This generates composed processing data chains/graphs by concatenating successive operators on data sources – this is called *piping composition***.** On the other hand, he or she intercommunicates gadgets. This resolves their interdependences to achieve an interconnected graphical mashup solution able to solve complex functional requirements like a traditional desktop application does – this is called *wiring composition*. To improve reusability and thus foster collaboration, both piping and wiring composition activities support marshalling (e.g. XSLT) and parameterization.

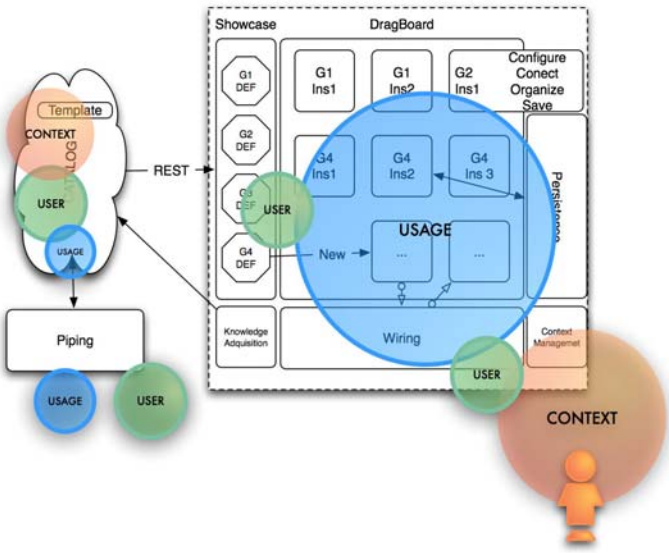

Fig. 3. Resource Composition in Enterprise Mashups

Based on these composition techniques, a dynamic universe of resources is made available to the knowledge worker to boost his or her innovativeness.

7.3 The Enterprise Mashup Programming Metaphor

The key element of EzWeb's business mashup is the gadget or graphical human interaction resource (see Fig 3.). An EzWeb mashup then is composed of one or more separate gadgets that behave consistently in the mashup setting as if they were one strongly cohesive interface. 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 [8], is a key element for systems developed according to this architecture. However, this language, necessary for executing pipes, will be totally transparent to the knowledge worker, as he or she will have access to wiring- and piping-driven graphical composition mechanisms.

As shown in Fig. 3, 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 extremely easy to chain resources so that the output of one resource directly feeds the input of another. Thanks to this orthogonality, business systems interaction can be easily modeled by the transformation and validation phases of any XML data type (input and output), producing what are known as *operator resources* (e.g. XSL Transformations).

The pipe resources directly communicated with gadgets should take charge of supplying the gadgets with XHTML code. Therefore, they all are orthogonal too, and simply generate the rendering, requiring no further processing at all.

7.4 Fostering Collaboration and Innovation through the Catalogue

The huge numbers of resource-oriented enterprise services created by knowledge workers through daily collaboration constitutes a complex universe of available resources, characterized by a fast emergence pace. Even if a repository service is provided, it will eventually become difficult for knowledge workers to find out which resources they need to create the mashups they want, let alone to discover recently available capabilities and/or previously unknown relations between resources that could help them to innovate.

EzWeb architecture resolves this requirement from a Web 2.0 perspective by providing a user-contributed catalogue of resources (see Fig. 4). Knowledge workers are thus provided with highly collaborative Wiki, tagging and searching-by-recommendation capabilities to locate or even discover resources of their interest.

The catalogue sets out the knowledge available within the company for composing resources (including pipes, operators and mashups) in a graphical and usable fashion. There are two types of knowledge: (i) a collective pool of resources with capabilities for cloning, modifying and publishing new

resources, (ii) a list of resources with the business processes they support.

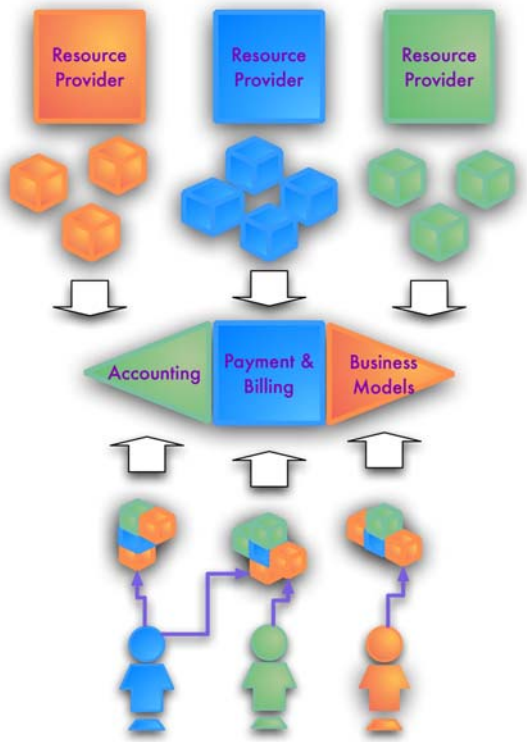

Fig. 4. Cataloguing resources

This allows knowledge workers to create complex mashup solutions by just looking for (or being recommended) "pre-cooked" off-the-shelf resources, customizing these resources, if needed, to suit their personal needs, interconnecting the resources, and integrating the outcome in their user workspace. The following sections describe and compare the two key contributions of this approach to the design of a components repository with other traditional conceptions: Wiki capabilities for co-editing knowledge about resources and the use of *folksonomies* to collaboratively categorize resources.

7.4.1. Wiki Philosophy Applied to the Catalogue

An Enterprise Wiki is a powerful collaborative tool specifically designed to allow knowledge workers to freely create and edit content through a Web browser [9]. This allows any reader to edit that content if they want to contribute new or refine existing ideas or they feel that the content is incorrect or poorly organized through iterative refinement. Through such iterative refinement, knowledge builds up very quickly as many people contribute small, manageable items, and each contribution iteratively improves the knowledge [10].

Therefore, the Enterprise Wiki has been considered the base technology for enabling sharing, refinement, and emergence of the mashup conceptualization knowledge present in the catalogue. This is a similar approach to networked ontologies. Focused on the catalogue, the Wiki helps to collect the knowledge required to represent composition relations between resources in the pool, recommendations on which are the best resources for a particular problem or at least about the business processes or subprocesses that these knowledge items can support.

7.4.2. Collaboratively Managing and linking knowledge work through tagging

The fact that the resources are provided through a uniform interface is a definite advantage in terms of simplicity, scalability and flexibility. On the other hand, it leads to the loss of most of the semantics associated with the above data and services. Existing standards have omissions concerning the people who directly handle the knowledge about services. Standards like WSMO (Web Service Modeling Ontology) [11] focus on the creation by experts of services ontologies for integration. The chosen Web 2.0 vision solution involves using tagging through the catalogue for resource and content users to attach collaborative, flexible and lightweight semantics to the resources and the associated Wiki content.

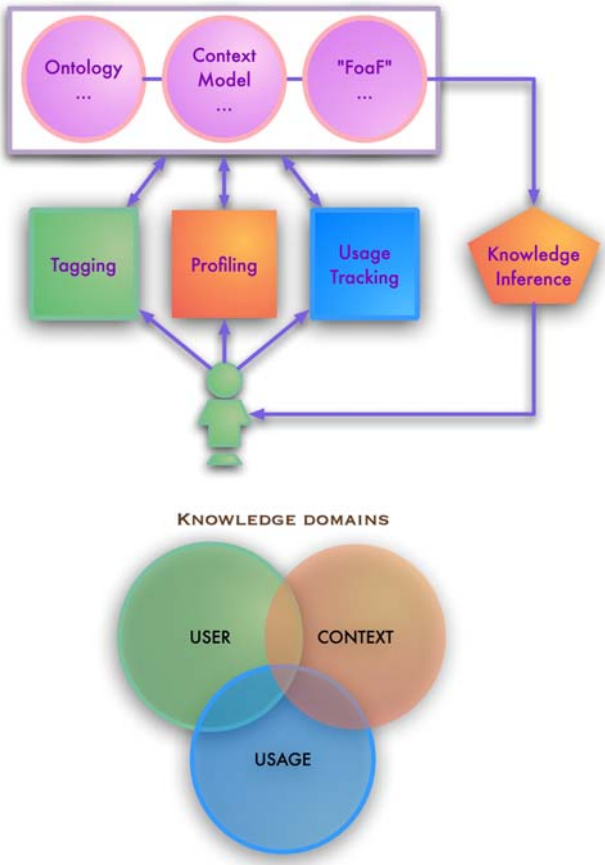

Fig. 5. Collective knowledge emergence through a catalogue

Managing the meaning of every tag in the folksonomy and the semantic relations between resources with the catalogue, together with its collaborative editing philosophy, leads to a common conceptualization in collective resource-related knowledge emergence, as shown in Fig. 5.

7.4.3. Advantages of adopting this catalogue vision

Suffice it to compare this solution with earlier approaches to service discovery and description like UDDI (Universal Discovery, Description and Integration) and WSDL (Web Services Description Language) to confirm the benefits of emerging collective intelligence. Multidimensional resource

categorization and the semantic description of resource utility and interrelations feed off real use by people. Even current solutions with semantic support like WSMO are clearly oriented to interoperability and automatic services composition. They are therefore very difficult for people to use directly, so the proposed approach focus primarily on exploitation by users instead.

Catalogue accessibility and ease of use encourages all users to continuously improve their operating environments, taking advantage of the business knowledge generated from the employee's viewpoint. This knowledge has so far been underused in this respect, even though it is viewed as the most valuable business structure asset.

In any case, this vision of the catalogue can be added to on a formal plane, not perceived by users, through a possible exploitation of the socially emerged catalogue. The fact that the catalogue ultimately contains the actual low-level resources can even be obviated, as its simple, natural language content, with specific descriptions and user-created relations, enables business rules and knowledge, which used to be really hard to extract, to be elicited from this *Wiki-tag* catalogue.

8. Conclusions

The appearance of Enterprise 2.0 Collaboration platforms, such as EzWeb, will be a major step forward, providing solutions to currently hard-to-solve problems in up to three different scenarios. Large enterprises may capitalize on faster application development (for the kind of applications known as instant applications), a more agile system landscape and the empowerment of their employees to design their own applications that match best their unique requirements, and to share this knowledge with other employees. This approach to setting up and managing enterprise IT systems will considerably reduce the backlogs that IT departments frequently have to cope with and increase operational flexibility, which is crucial in a world with fierce global competition and quickly changing market demands. It also creates an ecosystem for collaboration and co-creation of new services with other stakeholders, then profit from a well-known truism, now known as Joy's law (Note 3): "Innovation happens elsewhere".

On the other hand, Enterprise 2.0 Collaboration platforms enable SMOs to find, customize, combine, catalogue, share and finally use applications that exactly meet their individual demands by leveraging the SaaS model. Supported by the EzWeb platform, they can select and combine applications hosted by third parties rather than buying a pre-determined, inflexible and potentially heavyweight solution.

Finally, individuals benefit from a strongly increased capability of personalization and participation. The EzWeb mashup platform will provide end-users with intuitive, unsophisticated IT ways to discover, remix and use those Web-based applications that they consider interesting and useful. It also allows 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 via the platform. This speeds up the service innovation pace. Focusing on the "long tail" advanced by Chris Anderson [13] rather than a limited number of sophisticated experts, the EzWeb platform will involve the bulk of private users or small businesses and allow for "customer self-service".

The key idea behind this vision, and the lesson many businesses must learn, is that next-generation IT systems must be conceived to acquire the knowledge they operate on directly from who really has it, i.e. their knowledge workers and from the operation and communication processes the latter enter into, and that with the aid of enterprise 2.0 collaboration architectures like EzWeb much of this operational knowledge can be easily and effectively communicated as enterprise mashups.

References

[1] E. Driver et al. Road Map to an Enterprise Collaboration Strategy. Forrester Research, August 2, 2004.

[2] A. P. McAfee. Enterprise 2.0: The Dawn of Emergent Collaboration. MIT Sloan Management Review, 47(3), Spring 2006, (pp. 21-28).

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

[4] Intel Suite Two. The Enterprise 2.0 Suite, powered by Intel. Available at http://www.suitetwo.com.

[5] J. Coveyduc, C. Huang, L. Ostdiek, J. Reif. IBM Innovation Factory. An integrated solution for accelerating innovation. IBM. October 2006. Available at http://ibm.com/websphere/developer/zones/hipods.

[6] R. Smith. Enterprise Mashups: An Industry Case Study. Keynote at the New York PHP Conference & Expo, Manhattan, New York, USA, 14-16 June 2006, .

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

[8] T. Anderson. Is Yahoo Pipes all it's cracked up to be? IT Week, 19 February 2007. Available at http://www.itweek.co.uk/itweek/comment/2185589/yahoo-pipes-cracked.

[9] B. Venners. Exploring with Wiki: A conversation with Ward Cunningham, Part I. Artima Developer. 20 October 2003. Available at http://www.artima.com/intv/wiki.html

[10] J. Giles. Internet encyclopedias go head to head. Nature, 438(7070), December 2005, pp. 900-901.

[11] D. Roman et al.: Web Service Modeling Ontology, Applied Ontology, 1(1): 77 - 106, 2005.

[12] 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.

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

Acknowledgements

This work is being supported in part by the Spanish Ministry of Industry, Tourism and Commerce under its National Program of Service Technologies for the Information Society (contract FIT-350400-2006-23) and by the UPM and the European Social Fund under their Research Personnel Training program.

Note

1. The term *Knowledge worker* is used extensively throughout the paper in the sense of [12], to refer to employees, partners, suppliers, customers and other possible stakeholders.
2. Telefónica . http://www.telefonica.com/home_eng.shtml
3. Bill Joy, the co-founder and Chief Scientist of Sun Microsystems, famously uttered this truism.