



HAL
open science

Enterprise Mashup: Putting a face on the next generation global SOA - Wise 2007

Miguel A. Cañas, Juan J. Hierro, Volker Hoyer, Till Janner, David Lizcano, Marcos Reyes, Christoph Schroth, Javier Soriano

► To cite this version:

Miguel A. Cañas, Juan J. Hierro, Volker Hoyer, Till Janner, David Lizcano, et al.. Enterprise Mashup: Putting a face on the next generation global SOA - Wise 2007. 2007. inria-00198145

HAL Id: inria-00198145

<https://inria.hal.science/inria-00198145>

Submitted on 16 Dec 2007

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.

Enterprise Mashup

Putting a face on the next generation global SOA



SYSTEMATIC THOUGHT LEADERSHIP FOR INNOVATIVE BUSINESS

Miguel A. Cañas², Juan J. Hierro², Volker Hoyer¹, Till Janner¹,
David Lizcano³, Marcos Reyes², Christoph Schroth¹, Javier Soriano³

¹SAP Research (Switzerland)

²Telefónica I+D (Spain)

³Universidad Politécnica de Madrid, (Spain)

The 8th International Conference on Web Information Systems Engineering
3-7th December, 2007, Nancy, France



SAP RESEARCH

THE BEST-RUN BUSINESSES RUN SAP™



Time Schedule (14:00-17:00 CET)



Time	Topic
14:00-14:10	Opening, welcome, and introduction
14:10-14:30	General motivation for Enterprise Mashup, related work, and context of the tutorial
14:30-14:50	Enterprise Mashup foundations – terms and concepts
14:50-15:20	Enterprise Mashup foundations – key technologies
15:20-15:30	Overview (demo), and comparison of existing platforms (part 1)
15:30-16:00	Coffee Break (scheduled for all workshops/tutorials)
16:00-16:15	overview (demo), and comparison of existing platforms (part 2)
16:15-16:45	Live-demos of Telefónica I+D prototypes
16:45-17:00	Q&A, discussion, tutorial closing

SAP RESEARCH

© SAP 2007 / Page 2

Executive Summary



The Situation

- SOA and Web Service technologies lack of end-user integration and human-/ service interaction
- Web2.0 technologies provide new concepts to improve this situation
- Enterprise Mashup Platforms and tools are gaining momentum in research and industry

Theory & Practice

- Introducing the theoretical and technical foundation of Enterprise Mashup
- Market overview and comparison of current Mashup Platforms
- Presentation of live-demos of first prototypes in the Enterprise Mashup context

Enterprise Mashup

- Enterprise2.0: combining Web2.0 and Enterprise SOA
- Enterprise Mashups empower end-users, enabling them to create programs and applications that exactly meet their daily needs
- Enterprise Mashup - putting a face on the next generation global SOA

SAP RESEARCH

© SAP 2007 / Page 3

Agenda



1. Opening, welcome, and introduction
2. General Motivation for Enterprise Mashup
 - 2.1. Related work
 - 2.2. Related projects
3. (Enterprise) Mashup foundations
 - 3.1. Terms and concepts
 - 3.2. Key technologies
4. Overview and comparison of current mashup platforms
 - 4.1. Overview and demo of current mashup platforms
 - 4.2. Comparison and shortcomings
5. Live Demos from Telefónica I+D
 - 5.1. Telefónica I+D Seed Project: EzWeb
 - 5.2. MORFEO-EzWeb project
 - 5.3. Building MORFEO-EzWeb gadgets
6. Q&A, discussion and tutorial closing

SAP RESEARCH

© SAP 2007 / Page 4

Speakers introduction



Name	Till Janner
Affiliation	Institute for Media and Communications Management, University of St. Gallen, Switzerland SAP Research CEC St. Gallen, Switzerland
Position	Research Assistant and Doctoral Candidate at the Institute of Media and Communication Management (MCM) Research Associate at the SAP Research CEC St. Gallen
Fields of research	Enterprise Service-oriented Architectures, Enterprise Interoperability, Enterprise 2.0
Short CV	Till Janner holds a diploma-degree on Information Systems ("Diplom-Wirtschaftsinformatiker") of the University of Hamburg, Germany. Before joining University of St. Gallen and SAP Research he gained experience in Enterprise Software Development at industrial projects in the utilities sector.
Website	http://www.alexandria.unisg.ch/persons/Till_Janner



SAP RESEARCH



Contact:

Phone +41 71 224 7713

Mobile +41 76 220 7713

e-mail till.janner@unisg.ch

© SAP 2007 / Page 5

Speakers introduction



Name	David Lizcano
Affiliation	CoNWeT Lab, Department of Computer Science, Universidad Politécnica de Madrid, Spain
Position	Research associate and Ph.D. candidate at the Computer Networks and Web Technologies Lab (CoNWeT Lab), Universidad Politécnica de Madrid
Fields of research	Web 2.0, Enterprise 2.0, Service-oriented Architectures
Short CV	David Lizcano holds a M.Sc. degree with honors in Computer Science (2006) from the Universidad Politécnica de Madrid and holds a research grant from the UPM and the European Social Fund under their Research Personnel Training program. He has gained experience in Web Software Development at several IT European projects
Website	http://hydra.ls.fi.upm.es/research/



CoNWeT Lab
Computer Networks & Web Technologies Laboratory



Contact:

Phone +34 91 336 7394

e-mail dlizcano@fi.upm.es

SAP RESEARCH

© SAP 2007 / Page 6

Speakers introduction



Name	Miguel Angel Cañas Vaz
Affiliation	Telefónica I+D, Spain, Madrid
Position	PhD Student at the Universidad Politécnica de Madrid (UPM) IT Engineer at Telefónica I+T
Fields of research	Enterprise Software Architectures, Enterprise Collaboration, Enterprise 2.0
Short CV	Mr. Miguel Angel Cañas Vaz holds a B.Sc in Computing (2005) at the UPM. Miguel has gained professional experience at Telefónica I+D, getting involved in some R&D projects dealing with Workforce Management, Service-Oriented Architectures and Enterprise Collaboration applied to big enterprises.
Website	www.tid.es



Telefónica I+D

Contact:

Phone +34 91 337 39 09

Mobile +34 626 39 52 83

e-mail mac@tid.es

SAP RESEARCH

© SAP 2007 / Page 7

Agenda



1. Opening, welcome, and introduction
- 2. General Motivation for Enterprise Mashup**
 - 2.1. Related work
 - 2.2. Related projects
3. (Enterprise) Mashup foundations
 - 3.1. Terms and concepts
 - 3.2. Key technologies
4. Overview and comparison of current mashup platforms
 - 4.1. Overview and demo of current mashup platforms
 - 4.2. Comparison and shortcomings
5. Live Demos from Telefónica I+D
 - 5.1. Telefónica I+D Seed Project: EzWeb
 - 5.2. MORFEO-EzWeb project
 - 5.3. Building MORFEO-EzWeb gadgets
6. Q&A, discussion and tutorial closing

SAP RESEARCH

© SAP 2007 / Page 8

Empowering the end-user to overcome several shortcomings of SOA



SOA shortcomings

- As Gartner says in recent paper, the SOA paradigm has several shortcomings due to the complexity of SOAP, WSDL, UDDI, BPEL, and so on.
- The conventional Web Services stack does not suffice to enable a global SOA (interoperability issues, lack of global service registries)
- Users that are not expert programmers cannot afford to create their own SOA-oriented solutions.

Empowerment of the end-user

- Enterprise mashups empower end-users, enabling them to create programs and applications that exactly meet their daily needs
- Mashup supports the DIY (Do it yourself) ideology, putting a user-friendly face on traditional SOA, solving several problems related to user-service interaction
- End-users apply their expertise during the creation of the mashup, fostering innovation and knowledge emergence in enterprises.

SAP RESEARCH

© SAP 2007 / Page 9

Empowering Business Users Trends Today Are Re-Shaping the Workplace

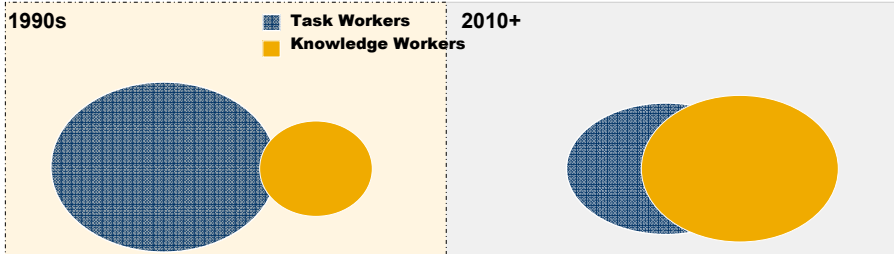


Web 2.0

- Consumer-like expectations
- Internet as lean consumption platform
- Advanced, exception-focused analytics

Nature of Work

- Increasing array of exceptions
- Collaboration across boundaries
- Needs context not information



SAP RESEARCH

* Source: IDC, The Yankee Group, Datamonitor, Unstrung News

© SAP 2007 / Page 10

Related work (part of the tutorial handout)



- **Fostering Innovation in a Mashup-oriented Enterprise 2.0 Collaboration Environment**, Javier Soriano, David Lizcano, Miguel A. Cañas, Marcos Reyes and Juanjo J.Hierro, System and Information Science Notes, Vol 1 Num 1, pp.62 - 69, July 2007. ISSN 1753-2310. SIWN International Conference on Adaptive Business Systems (ICABS'2007) Chengdu, China, 22-24 July 2007
- **Enhancing User-Service Interaction Through a Global User-Centric Approach to SOA**, Juan J. Hierro, Till Janner, David Lizcano, Marcos Reyes, Christoph Schroth, Javier Soriano, The Fourth International Conference on Networking and Services, ICNS'08. Gosier, Guadalupe, 16-21 March 2008. Published by IEEE Computer Society Press.
- Schroth, C., & Janner, T. (2007). **Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services**. IEEE IT Professional (9/3), 36-41.
- Schroth, C., & Christ, O. (2007). **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) (pp. 8): IEEE Computer Society.

SAP RESEARCH

© SAP 2007 / Page 11

Agenda



1. Opening, welcome, and introduction
2. **General Motivation for Enterprise Mashup**
 - 2.1. Related work
 - 2.2. **Related projects**
3. (Enterprise) Mashup foundations
 - 3.1. Terms and concepts
 - 3.2. Key technologies
4. Overview and comparison of current mashup platforms
 - 4.1. Overview and demo of current mashup platforms
 - 4.2. Comparison and shortcomings
5. Live Demos from Telefónica I+D
 - 5.1. Telefónica I+D Seed Project: EzWeb
 - 5.2. MORFEO-EzWeb project
 - 5.3. Building MORFEO-EzWeb gadgets
6. Q&A, discussion and tutorial closing

SAP RESEARCH

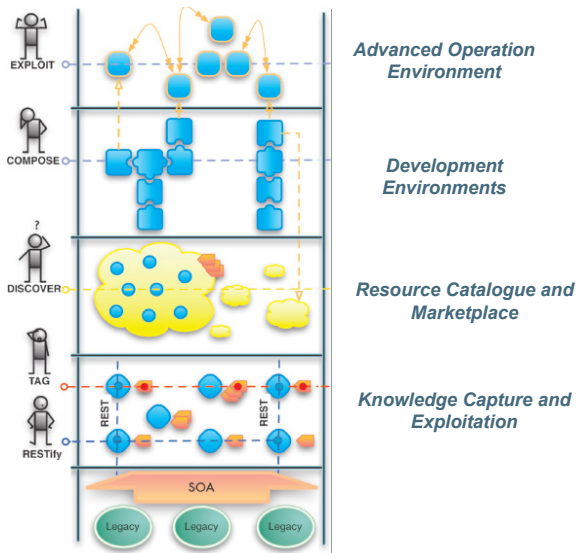
© SAP 2007 / Page 12

EzWeb Project by Telefonica I+D



EzWeb - Putting a Face on SOA

„Target reference architecture and implementation of an open standard platform which manages the front-end access in the next-generation web and supports the economics of a resource marketplace.“



SAP RESEARCH

© SAP 2007 / Page 13

The race for dominating the front-end in the future global SOA



The Internet is about to change...

- **The Internet of Services is about to emerge:** The Web is morphing into a sort of global operating system similar to Microsoft Windows
- **People may leverage resources** available on a global scale and mash them to solutions that meet their individual requirements
- **First examples already exist:** Mash-up platforms (Google, Yahoo pipes), REST-ful interfaces provided by enterprises and more

...EzWeb will act as pace-setter

- **EzWeb will drive the race** for the front-end in a future global SOA
- **Key principles: User-centrism** (user empowerment, ability to share information and context-aware interaction) as well as **Business-Centrism** (creation of a trustworthy marketplace that facilitates sustainable business ecosystems)
- **Private end-users, SMEs and large enterprises** will benefit

Who dominates the front-end access to the web will dominate the future

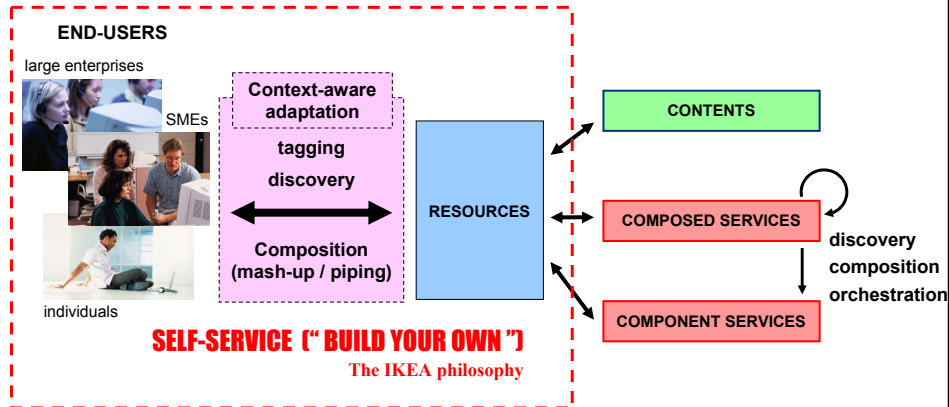
SAP RESEARCH

© SAP 2007 / Page 14

EzWeb: General concept and vision



The EzWeb project aims at providing the target reference architecture and reference implementation of an open standard platform which will support definition of the **front-end layer in the next-generation global SOA** (Service Oriented Architecture).



SAP RESEARCH

© SAP 2007 / Page 15

FAST AND ADVANCED STORYBOARD TOOLS

Collaborative Project, Small or medium-scale focused research project (STREP)
7TH FRAMEWORK PROGRAMME – THEME ICT-2007.1.2



TELEFÓNICA I+D, SAP AG, NATIONAL UNIVERSITY OF IRELAND – GALWAY,
UNIVERSITÄT KASSEL, CYNTELIX CORPORATION AND UNIVERSIDAD POLITÉCNICA DE MADRID



SAP RESEARCH

© SAP 2007 / Page 16

FAST Vision



Fast and Advanced Storyboard Tool



„FAST will create a new **visual programming environment** following a **user-centric approach** that 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”

SAP RESEARCH

© SAP 2007 / Page 17

FAST Project Consortium



Teléfono Investigación y Desarrollo (Coordinator)
Telecommunication Provider



SAP AG
World's leading Enterprise Application Provider



University of Ireland, Galway
Leading research institution in Semantic Web



University of Kassel
Strong research experience in scenario-based methodologies



Cyntelix Corporation
Semantic Web technology and collaboration platform



Universidad Politécnica de Madrid
Semantic Web and Web 2.0-based technologies

EU funded Project (FP7)

- March 2008 – February 2011
- Project Budget: 5,6 Mio €
- 6 Partners from academia and industry

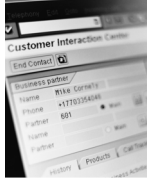
Project Consortium



SAP RESEARCH

© SAP 2007 / Page 18

FAST Objectives



New visual programming environment for the development of complex front-end gadgets

- Involves the modeling and execution of relatively complex business processes
- Relies on back-end Semantic Web Services
- Agile software development from a top-down perspective



Exploitation of the full power of semantics for integrating back-end with front-end (user-centric)

- Developers will start from the front-end gadgets
- Connection to back-end Web Services through business process execution flows if necessary



Plugin Tool for Enterprise Mashup Platforms

- Conceived as a tool that can be compatible with existing and future Enterprise Mashup Platforms
- Tool that enables to develop Mashup-able gadgets which rely on screen-flow Resources and Semantic Web Services stored in a (semantically enriched) catalogue

SAP RESEARCH

© SAP 2007 / Page 19

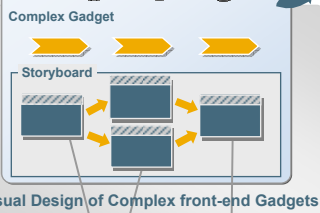
Architecture and Components Overview



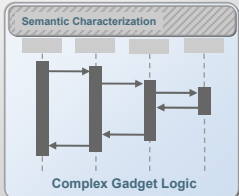
User-Centric Collaboration



Information Worker



Visual Composition of screen-flow Resources and Interoperability with back-end Web Services



Semantic Catalogue of screen-flow Resources and Back-end Web Services

SEMANTIC CATALOGUE

weather create order
calendar email
traffic information delete order
create invoice
hotel booking open invoice

BACK-END



SAP RESEARCH

© SAP 2007 / Page 20

Agenda



1. Opening, welcome, and introduction
2. General Motivation for Enterprise Mashup
 - 2.1. Related work
 - 2.2. Related projects
3. **(Enterprise) Mashup foundations**
 - 3.1. **Terms and concepts**
 - 3.2. Key technologies
4. Overview and comparison of current mashup platforms
 - 4.1. Overview and demo of current mashup platforms
 - 4.2. Comparison and shortcomings
5. Live Demos from Telefónica I+D
 - 5.1. Telefónica I+D Seed Project: EzWeb
 - 5.2. MORFEO-EzWeb project
 - 5.3. Building MORFEO-EzWeb gadgets
6. Q&A, discussion and tutorial closing

SAP RESEARCH

© SAP 2007 / Page 21

Web 2.0 vs. SOA



Web 2.0

“Web 2.0 is defined as the philosophy of mutually maximizing collective intelligence and added value for each participant by formalized and dynamic information sharing and creation.”

vs.

SOA

“... a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with, and use capabilities to produce desired effects consistent with measurable preconditions and expectations.”

SAP RESEARCH

© SAP 2007 / Page 22

Characteristics of Web2.0 (by Tim O'Reilly)



User Focus

- The exploitation of collective intelligence of Web users
- Realization of rich user experiences

Technology and Platform

- Considering the Web a platform for building systems that are "tied together by a set of protocols, open standards, and agreements for cooperation"
- Use of diverse media and devices for the consumption of Internet based applications

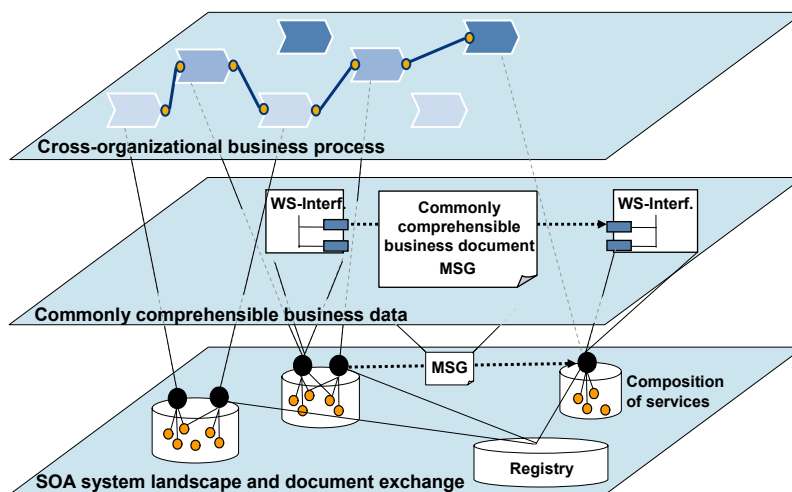
Development Model

- The end of the software release cycle
- The use of lightweight programming models that allow for loosely coupled applications

SAP RESEARCH

© SAP 2007 / Page 23

Cross-organizational Service-Oriented Architecture*



*C. Schroth, T. Janner: Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services, IEEE IT Professional Magazine, Vol. 9, No. 3, 2007

SAP RESEARCH

© SAP 2007 / Page 24

The concept of Mashup and Resources



Mashup ...

- 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.
- A **Mashup Platform** will facilitate flexible, useful, and effective user interaction and management with all kind of **resources**.

Resources

- Comply with **REST** architectural style
- Basically everything on the Web identified by an **URI**
- **Content or service** wrapped by HTTP verbs

Resource categories:

1. **Data sources** (feed the mashup app)
2. **Operators** (transform data sources)
3. **Gadgets** (provide the UI)

SAP RESEARCH

© SAP 2007 / Page 25

From “Traditional” to “Enterprise” Mashup



Traditional Mash-up ...

- **Based on content mash-up**, integrating contents published over the internet
- Users can easily **aggregate** data, contents and isolated functionalities to a web site
- Users can't create powerful applications or solutions only **collecting contents** in a page without “assembling” them

... Enterprise Mash-up

- **Based on Content-driven mashup-oriented programming**, with intercommunication between gadgets
- Users can easily **create** new web applications in a fully visual manner, **composing** solutions from resource orchestration
- Foster knowledge emergency and innovation: users **create applications** that supports their real needs during daily knowledge work

Translating mash-ups into enterprise-class applications is different story

SAP RESEARCH

© SAP 2007 / Page 26

Novel Approaches to Resource Provision



Wide take-up of resource provision

- Large companies such as Google, Amazon, Microsoft, ... start to successfully provide Web-based resources which are leveraged by numerous users

Resource provision in private and corporate context

- Professional services as well as applications that aim at the long tail of private Internet users are emerging

Light-weight, highly standardized protocols and styles

- Light-weight and highly standardized messaging protocols such as RSS, SOAP as well as architectural styles such as REST are quickly gaining momentum on a global scale

SAP RESEARCH

© SAP 2007 / Page 27

Rich Intermediaries Improve Transparency, Navigation and Governance



Transparency

- Intermediaries gather information about published resources over their whole lifecycles and thus improve transparency in a global resource market

Navigation

- Novel intermediaries provide navigation functionality and help user find desired services and content in the global mesh of resources

Governance*

- The reduction of uncertainty plays a relevant role in achieving a reliable, global SOA which is adopted by enterprises to execute mission-critical services over the Web**

* P. Weill and J.W. Ross: IT Governance: How Top Performers Manage IT Decision Rights for Superior Results, Harvard Business School Press, May 2004

** Networked European Software & Services Initiative (NESSI). NESSI Homepage. available online at: <http://www.nessi-europe.com/Nessi/>

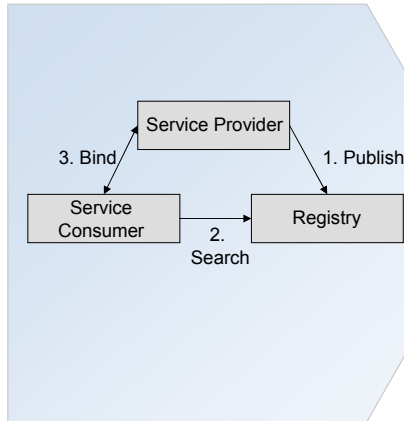
SAP RESEARCH

© SAP 2007 / Page 28

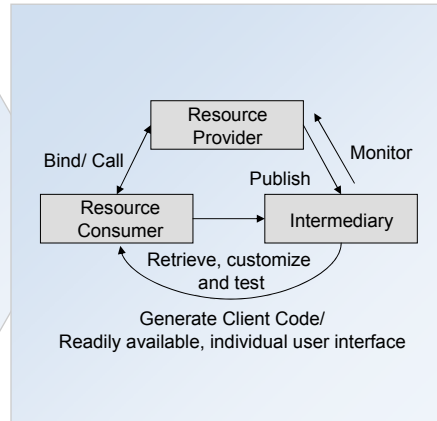
From Conventional Service Registries to Rich Intermediaries



From conventional service registries...



...to rich intermediaries*



*Example: WebRPC.com

SAP RESEARCH

© SAP 2007 / Page 29

New Ways of Resource Consumption enable "User Self-Service"



Google Gadgets

- Aggregation/ Syndication of content/ simple functionality
- Gadget repository
- Template for building own gadgets ("De-facto standard")
- No interaction between the resources (mere syndication)

Yahoo! Pipes

- Pipes can be defined that read in certain RSS-based "feeds"
- Filters can be applied on the pipes (to sort out information that is relevant for the individual user)
- The pipes' outputs can be aggregated and displayed to the user

Kapow Technologies

- Mash-ups can be setup on the basis of a visual modeling interface
- "Stateful" interaction between different resources can be defined and published

SAP RESEARCH

© SAP 2007 / Page 30

The Web Services Stack Does not Suffice to Support a “Global SOA”



Lack of interoperability

- The “corporate household problem”^{**} prevents from seamless interoperability across company boundaries (data structure as well as semantics differ significantly)

Lack of service registries

- Lack of comprehensive, trustworthy and widely accepted service registries prevents global SOAs

„Mute and autistic“

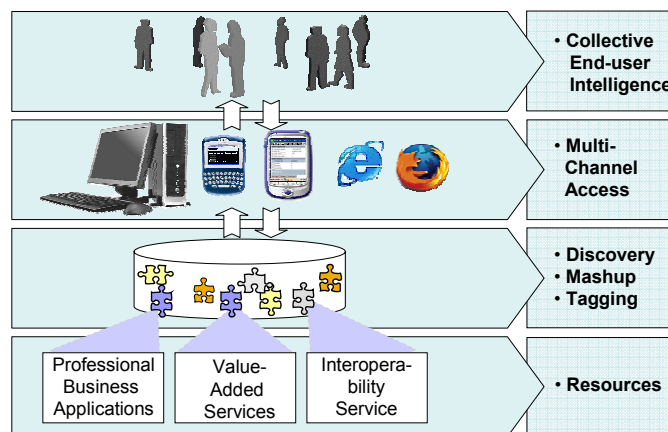
- The Web Services stack aims at supporting the setup of loosely coupled application interconnections especially in a professional context and **assumes users to be technically sophisticated**

* X. Chen, J. Funk, S. Madnick, R. Wang: Corporate Household Data: Research Directions, Proceedings of the Americas Conference on Information Systems, Boston, August 2001

SAP RESEARCH

© SAP 2007 / Page 31

The Internet of Services: Global infrastructure facilitating a networked services economy

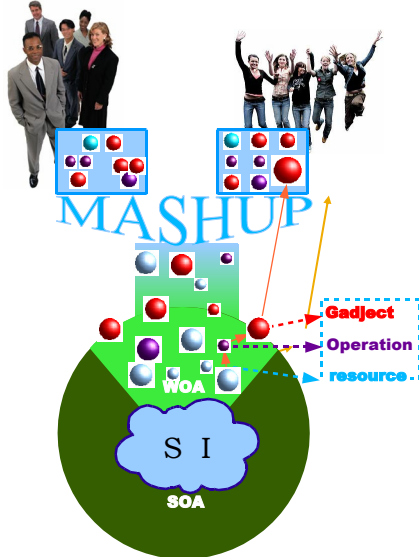


*C. Schroth, T. Janner: Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services, IEEE IT Professional Magazine, Vol. 9, No. 3, 2007

SAP RESEARCH

© SAP 2007 / Page 32

Web Oriented Architecture: WOA



- “Resources” (an URI, exporting HTTP get | post | put | delete operations returning interpretable XML) constitute the set of basic building blocks upon which to create end-user applications
- Web Services will be mainly used in Enterprise Application Integration and will be exposed to the end user through web resources
- Resources may be used and combined in a very simple, straightforward manner, based on preferences and needs from each user or user group

SAP RESEARCH

© SAP 2007 / Page 33

Agenda



1. Opening, welcome, and introduction
2. General Motivation for Enterprise Mashup
 - 2.1. Related work
 - 2.2. Related projects
3. (Enterprise) Mashup foundations
 - 3.1. Terms and concepts
 - 3.2. Key technologies
4. Overview and comparison of current mashup platforms
 - 4.1. Overview and demo of current mashup platforms
 - 4.2. Comparison and shortcomings
5. Live Demos from Telefónica I+D
 - 5.1. Telefónica I+D Seed Project: EzWeb
 - 5.2. MORFEO-EzWeb project
 - 5.3. Building MORFEO-EzWeb gadgets
6. Q&A, discussion and tutorial closing

SAP RESEARCH

© SAP 2007 / Page 34

REST vs SOAP



SOAP

- Service Oriented Architecture is an architectural style based on services, managed/invoked with SOAP
- Services are defined through WSDL, managed with SOAP, discovered with UDDI and orchestrated with BPEL. Each service has its own complex interface
- End-users can't manage these services easily: they need to know exactly how are their interfaces, and then invoke them using SOAP

vs.

REST

- Representational State Transfer architectural style is based on resources
- All modules, services or contents are modeled as resources
- A resource owns an 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 through a uniform interface that end-users can "see and touch"
- Users are empowered by allowing them to choose the best resources to meet their needs

SAP RESEARCH

© SAP 2007 / Page 35

REST Code example



Request:

GET www.parts-depot.com/parts?flavor=xml

Answer:

```
<?xml version="1.0"?>
<p:Parts xmlns:p="http://www.parts-depot.com"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=
"http://www.parts-depot.com
http://www.parts-depot.com/parts.xsd">
<Part id="00345" xlink:href="http://www.parts-depot.com/parts/00345"/>
<Part id="00346" xlink:href="http://www.parts-depot.com/parts/00346"/>
<Part id="00347" xlink:href="http://www.parts-depot.com/parts/00347"/>
<Part id="00348" xlink:href="http://www.parts-depot.com/parts/00348"/>
</p:Parts>
```

SAP RESEARCH

© SAP 2007 / Page 36

SOAP Code example



Request:

POST <http://www.parts-depot.com/soap/servlet/messagerouter>

with xml parameters:

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<p:getPartsList xmlns:p="http://www.parts-depot.com"/>
</soap:Body>
</soap:Envelope>
```

Answer

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<p:getPartsListResponse xmlns:p="http://www.parts-depot.com">
<Parts>
<Part-ID>00345<Part-ID>
<Part-ID>00346<Part-ID>
<Part-ID>00347<Part-ID>
<Part-ID>00348<Part-ID>
</Parts>
<p:getPartsListResponse>
</soap:Body>
</soap:Envelope>
```

SAP RESEARCH

© SAP 2007 / Page 37

Information feeds



- Information feeds are a family of Web formats used to publish frequently updated content
- Information feeds make it possible for people to keep up with their favorite web sites in an automated manner that's easier than checking them manually.
- Content can be read using software called a "feed reader" or an "aggregator." The user subscribes to a feed by entering the feed's link into the reader. The reader checks the user's subscribed feeds regularly for new content, downloading any updates that it finds
- JSON (JavaScript Object Notation) is a lightweight computer data interchange format. It is a text-based, human-readable format for representing simple data
- Important information feeds: RSS, Atom, JSON...



SAP RESEARCH

© SAP 2007 / Page 38

RSS Feed Code example



```
<?xml version="1.0"?>
<rss version="2.0">
  <channel>
    <title>Liftoff News</title>
    <link>http://liftoff.msfc.nasa.gov</link>
    <description>Liftoff to Space Exploration.</description>
    <language>en-us</language>
    <pubDate>Tue, 10 Jun 2003 04:00:00 GMT</pubDate>
    <lastBuildDate>Tue, 10 Jun 2003 09:41:01 GMT</lastBuildDate>
    <docs>http://blogs.law.harvard.edu/tech/rss</docs>
    <generator>Weblog Editor 2.0</generator>
    <managingEditor>editor@example.com</managingEditor>
    <webMaster>webmaster@example.com</webMaster>
    <item>
      <title>Star City</title>
      <link>http://liftoff.msfc.nasa.gov/news/2003/news-starcity.asp</link>
      <description>How do Americans get ready to work with Russians aboard the
      International Space Station? They take a crash course in culture, language and
      protocol at Russia's Star City.</description> <pubDate>Tue, 03 Jun 2003 09:39:21
      GMT</pubDate> <guid>http://liftoff.msfc.nasa.gov/2003/06/03.html#item573</guid>
    </item>
    <item>
      <title>Space Exploration</title>
      <link>http://liftoff.msfc.nasa.gov</link>
      <description>Sky watchers in Europe, Asia, and parts of Alaska and Canada will
      experience a partial eclipse of the Sun on Saturday, May 31st.</description>
      <pubDate>Fri, 30 May 2003 11:06:42 GMT</pubDate>
      <guid>http://liftoff.msfc.nasa.gov/2003/05/30.html#item572</guid>
    </item>
  </channel>
</rss>
```

SAP RESEARCH

© SAP 2007 / Page 39

JSON string Code example (EzWeb gadget representation)



```
{
  "vendor": "Morfeo",
  "description": "This gadget presents information about communication systems
failures",
  "name": "Bandeja_de_Averias",
  "tags": [],
  "image": "http://europa.ls.fi.upm.es/ezweb/gadgets/RedLightBulb.gif",
  "uri": "/user/admin/gadgets/Morfeo/Bandeja_de_Averias/1.0",
  "version": "1.0",
  "user": "userLogin",
  "template":
  {
    "variables":
    [
      { "type": "S", "name": "phoneNumber", "aspect": "EVEN" },
      { "type": "S", "name": "serviceHired", "aspect": "EVEN" }
    ],
    "size": { "width": 3, "height": 11 }
  },
  "mail": "miglesia@pegaso.ls.fi.upm.es",
  "shared": false,
  "xhtml":
  { "uri": "/user/admin/gadgets/Morfeo/Bandeja_de_Averias/1.0/xhtml" }
}
```

SAP RESEARCH

© SAP 2007 / Page 40

Resource composition: Piping and wiring



- wiring composition
 - intercommunicate gadgets
 - this resolves their interdependences to achieve an interconnected graphical mash-up solution to complex functional requirements as a traditional desktop application

- piping composition
 - interconnect a number of heterogeneous data sources to create gadgets
 - this generates composed processing data chains/graphs by concatenating successive operators on data sources

- Data Marshalling and parameterization
 - to improve reusability and thus foster collaboration, both piping and wiring composition activities support marshalling (e.g. XSLT) and parameterization

SAP RESEARCH

© SAP 2007 / Page 41

Gadgets or Widgets



- Is a portable chunk of code that can be installed and executed (commonly within any separate HTML-based web page) by an end user without requiring additional compilation.
- They are derived from the idea of reusable code that has existed for years.
- Gadgets represent the basic building blocks 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 applications in a very short time

- Other terms: badge, module, capsule, snippet, mini and flake

SAP RESEARCH

© SAP 2007 / Page 42

Gadget Code example (a Gadget template)



```
<?xml version="1.0" encoding="UTF-8" ?>
<Template schemaLocation="http://morfeo-project.org/2007/Template">
  <!-- Meta tags define gadgets properties -->
  <Catalog.ResourceDescription>
    <Vendor>Morfeo</Vendor>
    <Name>Configuracion de Red</Name>
    <Version>1.0</Version>
    <Author>miglesia</Author>
    <Mail>miglesia@pegaso.ls.fi.upm.es</Mail>
    <Description>this gadget...</Description>
    <ImageURI>http://europa.ls.fi.upm.es/ezweb/gadgets/red.jpg</ImageURI>
    <WikiURI>http://www.image.com</WikiURI>
  </Catalog.ResourceDescription>
  <Platform.Wiring>
    <Wire name="serviceHired" type="text" label="ServiceHired"
      friendcode="service" wiring="out" />
    <Wire name="deviceId" type="text" label="deviceId"
      friendcode="deviceId" wiring="in" />
    <Wire name="deviceStatus" type="text" label="deviceStatus"
      friendcode="deviceStatus" wiring="in" />
  </Platform.Wiring>
  <Platform.Link>
    <XHTML
      href="http://europa.ls.fi.upm.es/ezweb/gadgets/configuracionRed.html" />
  </Platform.Link>
  <Platform.Rendering width="1" height="3" />
</Template>
```

SAP RESEARCH

© SAP 2007 / Page 43

Agenda



1. Opening, welcome, and introduction
2. General Motivation for Enterprise Mashup
 - 2.1. Related work
 - 2.2. Related projects
3. (Enterprise) Mashup foundations
 - 3.1. Terms and concepts
 - 3.2. Key technologies
4. **Overview and comparison of current mashup platforms**
 - 4.1. Overview and demo of current mashup platforms
 - 4.2. Comparison and shortcomings
5. Live Demos from Telefónica I+D
 - 5.1. Telefónica I+D Seed Project: EzWeb
 - 5.2. MORFEO-EzWeb project
 - 5.3. Building MORFEO-EzWeb gadgets
6. Q&A, discussion and tutorial closing

SAP RESEARCH

© SAP 2007 / Page 44

iGoogle



iGoogle is a mash-up platform that aims to centralize all personal information in a portal (or web page)

iGoogle is based on:

- Free wheel composition of gadgets in a web page, considered as a personal desktop from where a web users' experience starts every day
- Users can choose gadgets from:
 - A catalogue of ready-made components
 - Gadgets created by end-users

SAP RESEARCH

© SAP 2007 / Page 45

iGoogle - Demo



iGoogle

Google Search | I'm Feeling Lucky

Advanced Search
Enterprise
Language Tools

Hangman 2.0 Beta has been removed. Undo | Close

Home | Web 2.0 | Add a tab

Select theme | Add stuff »

Web 2.0 Technology Feeds

TechCrunch | TechMeme | ScoopWizer | GlobalTechnology

TechCrunch
TechCrunch profiles the companies, products and events that are defining and transforming the new web. TechCrunch is written by Michael Arrington.

Tvtrip Brings a Fresh New Look at Video Guides for Hotels

TVtrip is offering a multimedia hotel guide with videos profess

[link 14 nov 2007 15:17:52 CET]

Inbox 2.0 Makes Me Sad

Saul Hansel at The New York Times is

[link 14 nov 2007 09:24:39 CET]

VibeAgent Steps Out Of Beta, Wants To Defeat TripAdvisor With People Power

Later today **VibeAgent**

[link 14 nov 2007 09:00:24 CET]

Ads by Google: Tech RSS Feeds | RSS XML | Blogger RSS

Colombi Spring Feeders

Spring Detangling Feeder SemiAuto. Feeds up to 5 Stations / Operators

www.colombi.com

PeefFinance

The professional and secure way for your payments to Switzerland.

www.peefinance.ch

SAP RESEARCH

© SAP 2007 / Page 46

IGoogle - Advantages



- Catalogue has a lot of existing gadgets to choose from
- You have the option of creating new gadgets to use in the mashup
- IGoogle users can accumulate a lot of useful information on their home page
- End-users can fix the mashup, replacing/resizing gadgets on the page
- Users can create complex applications, located on their home page

SAP RESEARCH

© SAP 2007 / Page 47

IGoogle - Disadvantages



- End-users will find it difficult to create complex mashups, because:
 - Heavy programmatic skills are required
 - There is a poor variety of data inputs
 - There is high coupling between mashups and the platform (a middleware gadget is required to allow gadgets to communicate)
- Created mash-ups are not compatible with any other platform
- There is no inherent communication between gadgets

SAP RESEARCH

© SAP 2007 / Page 48

IGoogle - Links



- User's manual
 - <http://www.google.es/support/?ctx=web>
- Gadgets Development Api
 - http://code.google.com/apis/gadgets/docs/dev_guide.html
- Gadgets Development IDE
 - <http://desktop.google.com/dev/designer.html>
- YouTube Video of the platform
 - <http://www.youtube.com/watch?v=JorJOBbFKWM>

SAP RESEARCH

© SAP 2007 / Page 49

Yahoo Pipes



Visual development environment for generating data-oriented mashups

- Development is based on:
 - Dragging gadgets (pipes) from a toolbox and dropping them in work zone
 - Specifying data input
 - Interconnecting gadgets through pipes
 - Specifying data output format

SAP RESEARCH

© SAP 2007 / Page 50

Yahoo Pipes - Demo



SAP RESEARCH

© SAP 2007 / Page 51

Yahoo Pipes - Demo



SAP RESEARCH

© SAP 2007 / Page 52

Yahoo Pipes - Advantages



- Entirely graphic and intuitive IDE; not one line of code is required
- Inherent internal communication between gadgets
- RSS, CSV and JSON support
- Portability of mashups to other platforms (i.e. QedWiki)

SAP RESEARCH

© SAP 2007 / Page 53

Yahoo Pipes - Disadvantages



- Yahoo pipes does not generate XML from the mashup (limiting hacking and remixing)
- Data inputs are limited to a prefixed set
- It is very difficult to create complex mashups with really powerful functionalities early on
- Other platforms are usually required to create enterprise mashups

SAP RESEARCH

© SAP 2007 / Page 54

Yahoo Pipes - Links



- Yahoo Pipes Blog
 - <http://blog.pipes.yahoo.com/>
- Video tutorial
 - <http://www.jumpcut.com/fullscreen?id=594F555C568011DC9D24000423CEF5B0&type=movie>
- Forum
 - <http://discuss.pipes.yahoo.com/>
- Catalogue
 - <http://pipes.yahoo.com/pipes/pipes.popular>
- Tutorials
 - <http://pipes.yahoo.com/pipes/docs?doc=tutorials>
- FeedBack and suggestions
 - <http://suggestions.yahoo.com/?prop=Pipes>

SAP RESEARCH

© SAP 2007 / Page 55

OpenKapow

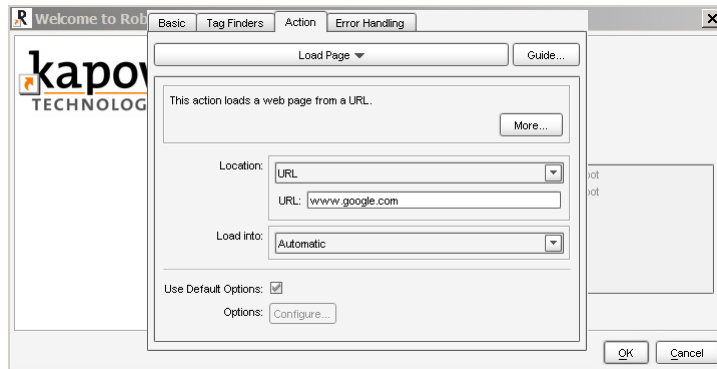


- Mashup platform for creating and remixing gadgets called “robots”. Robots have the following features:
 - Each robot is like a little components mashup
 - Robot inputs:
 - XML
 - HTML through screen scraping
 - Robots can manage a wide variety of data outputs
 - Robots can communicate with each other
 - These robots are created and managed with a desktop IDE, RoboMaker

SAP RESEARCH

© SAP 2007 / Page 56

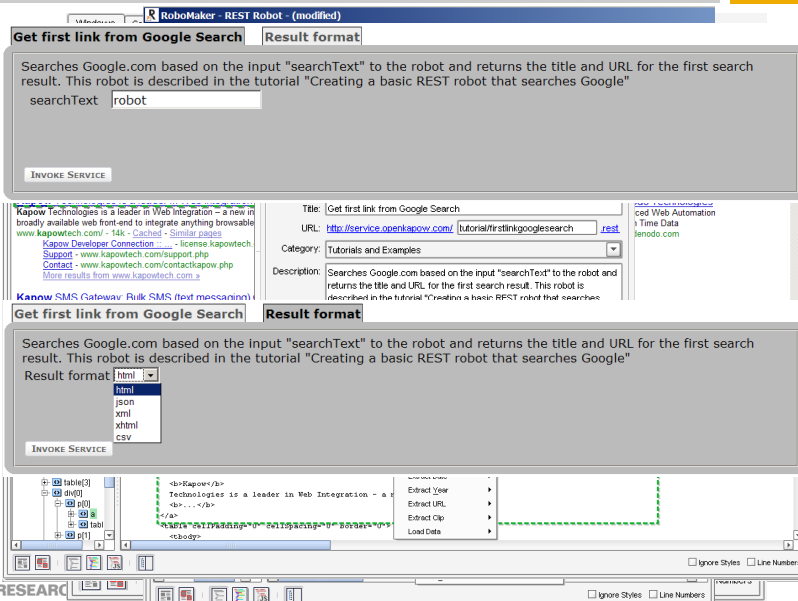
OpenKapow - Demo



SAP RESEARCH

© SAP 2007 / Page 57

OpenKapow - Demo



OpenKapow - Advantages



- This platform has a complete catalogue of prefabricated resources
- There is inherent communication between gadgets
- RoboMaker is a powerful and visual IDE
- It has a wide range of data inputs (XML, HTML, ...)
- It has a wide range of data outputs (XML, HTML, CSV, JSON, XHTML)
- It includes several ways to interact with end-users in final mashups (forms, REST calls, ...)
- Platform users can have their own robots server
- There is a lot of information, user guides, tutorials and support

SAP RESEARCH

© SAP 2007 / Page 59

OpenKapow - Disadvantages



- The platform is focused entirely on development, instead of exploitation
- You have to download an additional desktop IDE
- This platform targets heavy-tech users, with previous programming skills
- Robots can't be easily hacked or remixed
- The OpenKapow mashup server is not free
- The platform is overly centered on HTML and screen scraping

SAP RESEARCH

© SAP 2007 / Page 60

OpenKapow - Links



- Catalogue
 - <http://openkapow.com/files/robothome.aspx>
- Demos
 - <http://openkapow.com/blogs/demos/default.aspx>
- Tutorials
 - <http://openkapow.com/blogs/learn.aspx>
- Forums
 - <http://openkapow.com/forums/default.aspx8>
- Blog
 - <http://openkapow.com/blogs/news/default.aspx>

SAP RESEARCH

© SAP 2007 / Page 61

IBM QedWiki



- Mashup platform that combines two perspectives (data-oriented mashups and presentation mashups)
- Features:
 - Data inputs:
 - Databases
 - Feed RSS
 - Yahoo Pipes mashups
 - Spread sheets
 - XML
 - ...
 - Mashup process based on drag & drop operations, using gadgets from a toolbox, or further searching from a wiki of resources
 - Internal communication between gadgets is supported
 - Mashup results can be entered in a wiki, tagged, and refined by other users

SAP RESEARCH

© SAP 2007 / Page 62

IBM QedWiki - Demo



The screenshot shows a workspace with a 'Show Data' gadget. The gadget displays a news feed from scotsman.com. A red dashed line highlights a specific news item in the feed and its corresponding entry in the 'Show Data' table.

Title
Switzerland reflects on the death of Pinochet - NZZ Online
EU agrees partial freeze of Turkey talks - Scotsman
Germany tracks platinum fall - Reuters UK
Report: US Tapped Diana's Phones - WEBM780
Iran opens controversial Holocaust conference - CBC News
Bush Looks to Reverse Course of Iraq War - ABC News
Pakistan Inches Toward 2008 Div. - Space Daily

SAP RESEARCH

© SAP 2007 / Page 63

IBM QedWiki - Advantages



- QedWiki combines several tendencies; it supports data mashups, like Yahoo Pipes, and creates complex presentation mashups, like OpenKapow
- It supports internal communication between gadgets
- It supports a wide range of data inputs, managing databases directly
- Web-based IDE, no additional software has to be downloaded
- It can manage the internal code of each gadget and even of each complete mashup (according to its own XML-based standards)
- Simple documentation support oriented to inexperienced users

SAP RESEARCH

© SAP 2007 / Page 64

IBM QedWiki - Disadvantages



- You need to know code standards and the gadget programming language to build complex solutions because of limitations in the supplied visual IDE
- Screen scraping is not supported
- There is little, poorly detailed and slowly updated documentation. The QedWiki Project is not hot today.

SAP RESEARCH

© SAP 2007 / Page 65

IBM QedWiki - Links



- QEDWiki Help
 - <http://services.alphaworks.ibm.com/qedwiki/learn.html>
- Tutorial
 - <http://www.youtube.com/watch?v=63qlq9t9Gqs>
- Blog
 - http://www.mashboards.net/index.php?option=com_weblinks&catid=27&Itemid=23
- Forums
 - <http://services.alphaworks.ibm.com/qedwiki/forum.html>
- FAQ
 - <http://services.alphaworks.ibm.com/qedwiki/faq.html>

SAP RESEARCH

© SAP 2007 / Page 66

Agenda



1. Opening, welcome, and introduction
2. General Motivation for Enterprise Mashup
 - 2.1. Related work
 - 2.2. Related projects
3. (Enterprise) Mashup foundations
 - 3.1. Terms and concepts
 - 3.2. Key technologies
- 4. Overview and comparison of current mashup platforms**
 - 4.1. Overview and demo of current mashup platforms
 - 4.2. Comparison and shortcomings**
5. Live Demos from Telefónica I+D
 - 5.1. Telefónica I+D Seed Project: EzWeb
 - 5.2. MORFEO-EzWeb project
 - 5.3. Building MORFEO-EzWeb gadgets
6. Q&A, discussion and tutorial closing

SAP RESEARCH

© SAP 2007 / Page 67

Platforms Comparison – General features



Features	iGoogle	pipes	openkapow	QEDWiki
Wiki-based mashup catalogue	✗	✗	✗	✓
Free space to publish mashups	✓	✓	✗	✓
Visual composition (drag & drop)	✓	✓	✗	✓
Gadgets IDE	✗	✗	✓	✓
Internal communication	✗	✓	✓	✓
Mashups reuse between platforms enabled	✗	✓	✓	✓
REST support	✗	✗	✓	only REST inputs
Creation of Web-services based on mashups	✗	✗	✓	✗

© SAP 2007 / Page 68

SAP RESEARCH

© SAP 2007 / Page 68

Platforms Comparison – Supported inputs



Inputs	iGoogle	pipes ^{beta}	openkapow ^{beta}	QEDWiki
RSS	✓	✓	✓	✓
JSON	✓	✓	✓	✓
XML	✓	✓	✓	✓
HTML	✓	✓	✓	✓
Screen Scraping	✗	✗	✓	✗
CSV	✗	✓	✗	✓
Spread Sheets	✗	✗	✗	✓

© SAP 2007 / Page 69

SAP RESEARCH

© SAP 2007 / Page 69

Platforms Comparison – Supported outputs



Outputs	iGoogle	pipes ^{beta}	openkapow ^{beta}	QEDWiki
RSS	✗	✓	✓	✓
JSON	✗	✗	✓	✓
XML	✓	✓	✓	✓
HTML	✓	✗	✓	✗
CSV	✗	✓	✓	✓
XHTML	✗	✗	✓	✗
Spread Sheets	✗	✗	✗	✓

© SAP 2007 / Page 70

SAP RESEARCH

© SAP 2007 / Page 70

Platforms Comparison – Documentation



Documentation	iGoogle	pipes ^{beta}	openkapow ^{beta}	QEDWiki
Help, user guide, tutorials, demo	✓	✓	✓	✓
Blog / Forums	✗	✓	✓	✓
FAQ	✗	✗	✓	✗
User feedback	✗	✓	✗	✗
Detailed mashup documentation & support	✓	✗	✗	✗

© SAP 2007 / Page 71

SAP RESEARCH

© SAP 2007 / Page 71

Platforms Comparison – IDEs



IDE features	pipes ^{beta}	openkapow ^{beta}	QEDWiki
IDE as a service	✓	✗	✓
No previous configuration required	✓	✗	✗
Completely visual development	✓	✗	✓
Given items can be modified	✗	✓	✓
Low-level programming supported	✗	✓	✓
Required Programming level	LOW	HIGH	MEDIUM/HIGH

© SAP 2007 / Page 72

SAP RESEARCH

© SAP 2007 / Page 72

A Feature List For An “Ideal” Enterprise Mashup Platform



In general, all the platforms have several weak points and major shortcomings. An ideal enterprise mashup platform should support all the features below:

- A powerful gadget catalogue, based on an “architecture of participation” (i.e. taggable wiki) users add value
- Free mashup composition, remix and management
- A wide range of inputs/outputs (XML, HTML, DDBB, Spread Sheets, ...)
- Mashup creation as REST-oriented Web Services
- Internal communication between gadgets to create real complex mashups
- Mashups should follow a shared standard to enable compatibility between platforms
- Easy-to-use Web and visual IDE with low-level programming support and a high level of abstraction
- Detailed documentation and user guides:
 - Blogs and forums
 - Feedback
 - FAQ

SAP RESEARCH

© SAP 2007 / Page 73

Agenda



1. Opening, welcome, and introduction
2. General Motivation for Enterprise Mashup
 - 2.1. Related work
 - 2.2. Related projects
3. (Enterprise) Mashup foundations
 - 3.1. Terms and concepts
 - 3.2. Key technologies
4. Overview and comparison of current mashup platforms
 - 4.1. Overview and demo of current mashup platforms
 - 4.2. Comparison and shortcomings
5. **Live Demos from Telefónica I+D**
 - 5.1. Telefónica I+D Seed Project: EzWeb
 - 5.2. MORFEO-EzWeb project
 - 5.3. Building MORFEO-EzWeb gadgets
6. Q&A, discussion and tutorial closing

SAP RESEARCH

© SAP 2007 / Page 74

Telefónica I+D Seed Project: EzWeb



- EzWeb was developed under the scope of a internal innovation process at Telefónica I+D
- It was a first approach in order to extrapolate Mashup and Web2.0 technologies to an enterprise environment
- Simple objective:
 - Validate available Mashup tools in the scope of enterprise needs
 - Validate utility of Mashup related technologies from an enterprise perspective
- At Q4 2006, there were not many Mashup Platforms
 - IGoogle was selected as the target platform
- Structured on two phases:
 - Enterprise Services Wrap-up by REST resources
 - Gadget construction
- Visually Mashing-up the developed gadgets, a complete powerful operational environment can be easily built.
- The operational environment manages typical activities for solving Telco failures:
 - Presenting all kind of user and network info
 - Evolving incidences on an enterprise workflow system
 - Testing Telco devices

SAP RESEARCH

© SAP 2007 / Page 75

MORFEO-EzWeb project



- MORFEO (<http://www.morfeo-project.org>) is a free source community founded by Telefónica I+D and other partners
- MORFEO-EzWeb is a innovation project funded by the Spanish Administration
- The whole project results will be published in the MORFEO community in order to promote its public access
- The project is still under construction but an initial phase release is currently available
- New features (of the initial phase release):
 - Template concept
 - A flexible intergadget communication solution (wiring)
 - A simplified API
 - Advanced rendering capabilities
 - Social software tools integration (WIKI)

SAP RESEARCH

© SAP 2007 / Page 76

Building MORFEO-EzWeb gadgets



- Gadget construction is a three-phase process:
 - Declarative definition of gadget template
 - Writing gadget web interface
 - Writing communication source code via MORFEO-EzWeb API
- Once a set of gadgets are loaded into platform, user can customize their connections freely by the Wiring Manager with no IT expertise

SAP RESEARCH

© SAP 2007 / Page 77

Building MORFEO-EzWeb gadgets Gadget Template



```
<?xml version="1.0" encoding="UTF-8" ?>
<Template schemaLocation="http://morfeo-project.org/2007/Template">
  <!-- Meta tags define gadgets properties -->
  <Catalog:ResourceDescription>
    <Vendor>Morfeo</Vendor>
    <Name>Configuracion_de_Red</Name>
    <Version>1.0</Version>
    <Author>miglesia</Author>
    <Mail>miglesia@pegaso.ls.fi.upm.es</Mail>
    <Description>this gadget...</Description>
    <ImageURI>http://europa.ls.fi.upm.es/ezweb/gadgets/red.jpg</ImageURI>
    <WikiURI>http://www.image.com</WikiURI>
  </Catalog:ResourceDescription>
  <!-- EzWeb Gadget Data Wiring -->
  <Platform:Wiring>
    <Wire name="serviceHired" type="text" friendcode="service" wiring="out" />
    <Wire name="deviceId" type="text" label="deviceId" friendcode="deviceId" wiring="in" />
  </Platform:Wiring>
  <Platform:Link>
    <XHTML href="http://europa.ls.fi.upm.es/ezweb/gadgets/configuracionRed.html" />
  </Platform:Link>
  <Platform:Rendering width="1" height="3" />
</Template>
```

SAP RESEARCH

© SAP 2007 / Page 78

Building MORFEO-EzWeb gadgets Gadget API



Wiring API:

```
# RW variable definition
EzWebAPI.createRWGadgetVariable = function(varName);
# R variable definition
EzWebAPI.createRGadgetVariable = function(varName, callbackFunction);
# Set a variable value
EzWebAPI.RWGadgetVariable.set = function(value);
```

Piping-enabler API:

```
#Traditional AJAX-based REST verbs
EzWebAPI.sendGet = function (url, context, successHandler, errorHandler);
EzWebAPI.sendPost = function (url, params, context,
                              successHandler, errorHandler);
EzWebAPI.sendPut = function ( url, params, context,
                              successHandler, errorHandler);
EzWebAPI.sendDelete = function (url, params, context,
                                 successHandler, errorHandler);
```

SAP RESEARCH

© SAP 2007 / Page 79

Building MORFEO-EzWeb gadgets Gadget Source Code (1/2)



```
<html>
...
<script language="javascript" src="/ezweb/js/EzWebAPI/EzWebAPI.js"></script>
...
<script>
var address = new EzWebAPI.createRGadgetVariable("address",_addressHandler);

function _initialize() {
  if (GBrowserIsCompatible()) {
    map = new GMap2(document.getElementById("map_canvas"));
    map.addControl(new GMapTypeControl());
    map.addControl(new GLargeMapControl());
    map.setCenter(new GLatLng(34, 0), 1);
    geocoder = new GClientGeocoder();
  }
}

function _addressHandler(address) {
  if (address == null || address == undefined) {
    map.setCenter(new GLatLng(34, 0), 1);
    return;
  }
}
</script>

<body style="padding: 0px; margin: 0px" onload="_initialize;">
<div id="map_canvas" style="width:500px; height: 400px;"></div>
</body>
</html>
```

SAP RESEARCH

© SAP 2007 / Page 80

Building MORFEO-EzWeb gadgets Gadget Source Code (2/2)



```
<html>
...
<script language="javascript" src="/ezweb/js/EzWebAPI/EzWebAPI.js"></script>
...
<script>
var phoneNumber = EzWebAPI.createRGadgetVariable("phoneNumber",_phoneHandler);
var address = EzWebAPI.createRWGadgetVariable("address");

function _phoneHandler(phone){
if (phone==null && phone!=undefined){
var content="";
content+="
```

SAP RESEARCH

© SAP 2007 / Page 81

Agenda



1. Opening, welcome, and introduction
2. General Motivation for Enterprise Mashup
 - 2.1. Related work
 - 2.2. Related projects
3. (Enterprise) Mashup foundations
 - 3.1. Terms and concepts
 - 3.2. Key technologies
4. Overview and comparison of current mashup platforms
 - 4.1. Overview and demo of current mashup platforms
 - 4.2. Comparison and shortcomings
5. Live Demos from Telefónica I+D
 - 5.1. Telefónica I+D Seed Project: EzWeb
 - 5.2. MORFEO-EzWeb project
 - 5.3. Building MORFEO-EzWeb gadgets
- 6. Q&A, discussion and tutorial closing**

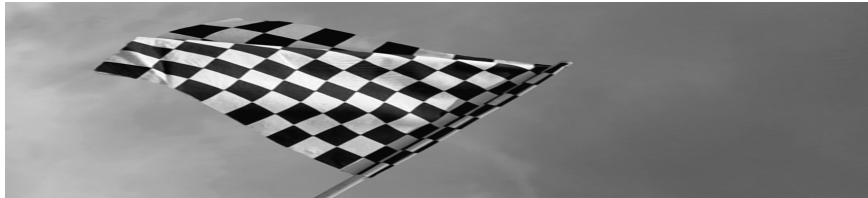
SAP RESEARCH

© SAP 2007 / Page 82

Summary and Outlook

Enterprise Mashup

Putting a face on the next generation global SOA



- Enterprise Mashups combine Enterprise-level services and Web2.0 technologies to **foster end-user innovation and knowledge emergence** in enterprises.
- **FAST**: Visual programming environment following a user-centric approach that will facilitate the development of complex front-end gadgets. **EzWeb**: reference architecture and implementation of an open standard platform which manages the front-end access in the next-generation web.
- Enterprise Mashup will a critical part of the **front-end layer in the next-generation global SOA**.

SAP RESEARCH

© SAP 2007 / Page 83

Q&A, discussion

Thank you!

Till Janner

SAP RESEARCH

SAP (Schweiz) AG
Blumenbergplatz 9
9000 St. Gallen
Switzerland

T +41/71/224-7713
M +41/76/220-7713

till.janner@sap.com
www.sap.com

David Lizcano



Universidad Politécnica de Madrid
Boadilla del Monte
28660 Madrid
Spain

T +34/91/336-7394

dizcano@fi.upm.es
www.fi.upm.es

Miguel Angel Cañas



Telefónica i+D
Emilio Vargas, 6
28043 Madrid
Spain

T +34/91/337-3909
M +34/626/39 52 83

mac@tid.es
www.tid.es

THE BEST-RUN BUSINESSES RUN SAP™



Copyright 2007 SAP AG All rights reserved



No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, Duet, Business ByDesign, ByDesign, PartnerEdge and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned and associated logos displayed are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

The information in this document is proprietary to SAP. This document is a preliminary version and not subject to your license agreement or any other agreement with SAP. This document contains only intended strategies, developments, and functionalities of the SAP® product and is not intended to be binding upon SAP to any particular course of business, product strategy, and/or development. SAP assumes no responsibility for errors or omissions in this document. SAP does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within this material. This document is provided without a warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. This limitation shall not apply in cases of intent or gross negligence.

The statutory liability for personal injury and defective products is not affected. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third-party Web pages nor provide any warranty whatsoever relating to third-party Web pages

Weitergabe und Vervielfältigung dieser Publikation oder von Teilen daraus sind, zu welchem Zweck und in welcher Form auch immer, ohne die ausdrückliche schriftliche Genehmigung durch SAP AG nicht gestattet. In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden.

Einige von der SAP AG und deren Vertriebspartnern vertriebene Softwareprodukte können Softwarekomponenten umfassen, die Eigentum anderer Softwarehersteller sind.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, Duet, Business ByDesign, ByDesign, PartnerEdge und andere in diesem Dokument erwähnte SAP-Produkte und Services sowie die dazugehörigen Logos sind Marken oder eingetragene Marken der SAP AG in Deutschland und in mehreren anderen Ländern weltweit. Alle anderen in diesem Dokument erwähnten Namen von Produkten und Services sowie die damit verbundenen Firmenlogos sind Marken der jeweiligen Unternehmen. Die Angaben im Text sind unverbindlich und dienen lediglich zu Informationszwecken. Produkte können länderspezifische Unterschiede aufweisen.

Die in diesem Dokument enthaltenen Informationen sind Eigentum von SAP. Dieses Dokument ist eine Vorabversion und unterliegt nicht Ihrer Lizenzvereinbarung oder einer anderen Vereinbarung mit SAP. Dieses Dokument enthält nur vorgesehene Strategien, Entwicklungen und Funktionen des SAP®-Produkts und ist für SAP nicht bindend. Einem bestimmten Geschäftsweg, eine Produktstrategie bzw. -entwicklung einzuschlagen, SAP übernimmt keine Verantwortung für Fehler oder Auslassungen in diesen Materialien. SAP garantiert nicht die Richtigkeit oder Vollständigkeit der Informationen, Texte, Grafiken, Links oder anderer in diesen Materialien enthaltenen Elemente. Diese Publikation wird ohne jegliche Gewähr, weder ausdrücklich noch stillschweigend, bereitgestellt. Dies gilt u. a., aber nicht ausschließlich, hinsichtlich der Gewährleistung der Marktgängigkeit und der Eignung für einen bestimmten Zweck sowie für die Gewährleistung der Nichtverletzung geltenden Rechts.

SAP übernimmt keine Haftung für Schäden jeglicher Art, einschließlich und ohne Einschränkung für direkte, spezielle, indirekte oder Folgeschäden im Zusammenhang mit der Verwendung dieser Unterlagen. Diese Einschränkung gilt nicht bei Vorsatz oder grober Fahrlässigkeit.

Die gesetzliche Haftung bei Personenschäden oder die Produkthaftung bleibt unberührt. Die Informationen, auf die Sie möglicherweise über die in diesem Material enthaltenen Hotlinks zugreifen, unterliegen nicht dem Einfluss von SAP, und SAP unterstützt nicht die Nutzung von Internetseiten Dritter durch Sie und gibt keinerlei Gewährleistungen oder Zusagen über Internetseiten Dritter ab.

Alle Rechte vorbehalten.

SAP RESEARCH

© SAP 2007 / Page 85

Enterprise Mashup

Putting a face on the next generation global SOA

Tutorial Handout

SYSTEMATIC THOUGHT LEADERSHIP FOR INNOVATIVE BUSINESS



Miguel A. Cañas², Juan J. Hierro², Volker Hoyer¹, Till Janner¹,
David Lizcano³, Marcos Reyes², Christoph Schroth¹, Javier Soriano³

¹SAP Research (Switzerland)

²Telefónica I+D (Spain)

³Universidad Politécnica de Madrid, (Spain)

The 8th International Conference on Web Information Systems Engineering
3-7th December, 2007, Nancy, France



SAP RESEARCH

THE BEST-RUN BUSINESSES RUN SAP™



Tutorial Handout – Content Overview



- Time schedule
- Tutorial short description
- Poster of the EzWeb project
- Poster of the FAST project
- Related papers from the tutorial authors

SAP RESEARCH

© SAP 2007 / Page 2

Time Schedule (14:00-17:00 CET)



Time	Topic
14:00-14:10	Opening, welcome, and introduction
14:10-14:30	General motivation for Enterprise Mashup, related work, and context of the tutorial
14:30-14:50	Enterprise Mashup foundations – terms and concepts
14:50-15:20	Enterprise Mashup foundations – key technologies
15:20-15:30	Overview (demo), and comparison of existing platforms (part 1)
15:30-16:00	Coffee Break (scheduled for all workshops/tutorials)
16:00-16:15	overview (demo), and comparison of existing platforms (part 2)
16:15-16:45	Live-demos of Telefónica I+D prototypes
16:45-17:00	Q&A, discussion, tutorial closing

SAP RESEARCH

© SAP 2007 / Page 3

“Enterprise Mashup: Putting a face on next generation global SOA”

Short description

Mashup is considered as the central enabler concept/technology of the Web 2.0 era. Based on definitions of [Hof, R.D., (2007)] and [The Economist (2005)] we take the following summary as foundation and definition of a Mashup: *A 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.*

Mashups in an Enterprise context aim at enabling the users to dynamically compose and interconnect their own operational environments and processes in a very simple and flexible fashion. An **Enterprise Mashup platform** will facilitate flexible, useful and effective user interaction and management with all kind of resources. Especially the integration of Web Service technology is important to mention in an Enterprise context. The creation of so-called Gadgets on top of services enables end-users to understand the not human readable WSDL interfaces of current Service-oriented Architectures.

The central topic of the tutorial, **Enterprise Mashups**, will be presented to the audience with two major parts, theory and practice. The theory part includes the motivation, the necessary definition of important terms and concepts and also an overview of the state of the art with regard to current approaches of mashup platforms. A comprehensive argumentation and presentation of the requirements towards the realization of an Enterprise Mashup platform completes this part.

During the practical part of the tutorial, **demonstrations of major existing platforms** that allow users to create Mashups will be given. Finally the presenters will give a **demonstration with a first prototype of an Enterprise Mashup platform**. A practical scenario will be presented that demonstrates the value of an Enterprise Mashup platform in a large enterprise.

The tutorial is targeted to an audience of (but not limited to) people with a research or industrial background and interest in the following topics:

- Enterprise Mashup
- User-/Service Interaction
- End-user driven service/resource Mashup
- Piping and wiring of Web Resources
- Enterprise 2.0
- Next-generation SOA

- Internet of Services

References

Hof, R.D., (2007): Mix, Match, And Mutate, Business Week Magazine, 2005, available online at: http://www.businessweek.com/magazine/content/05_30/b3944108_mz063.htm, last seen: May 30, 2007

The Economist (2005): Mashing the Web, The Economist 376 (8444), Special section, p. 4

O'Reilly, T. (2007): What is Web 2.0, <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>, last seen: May 30, 2007

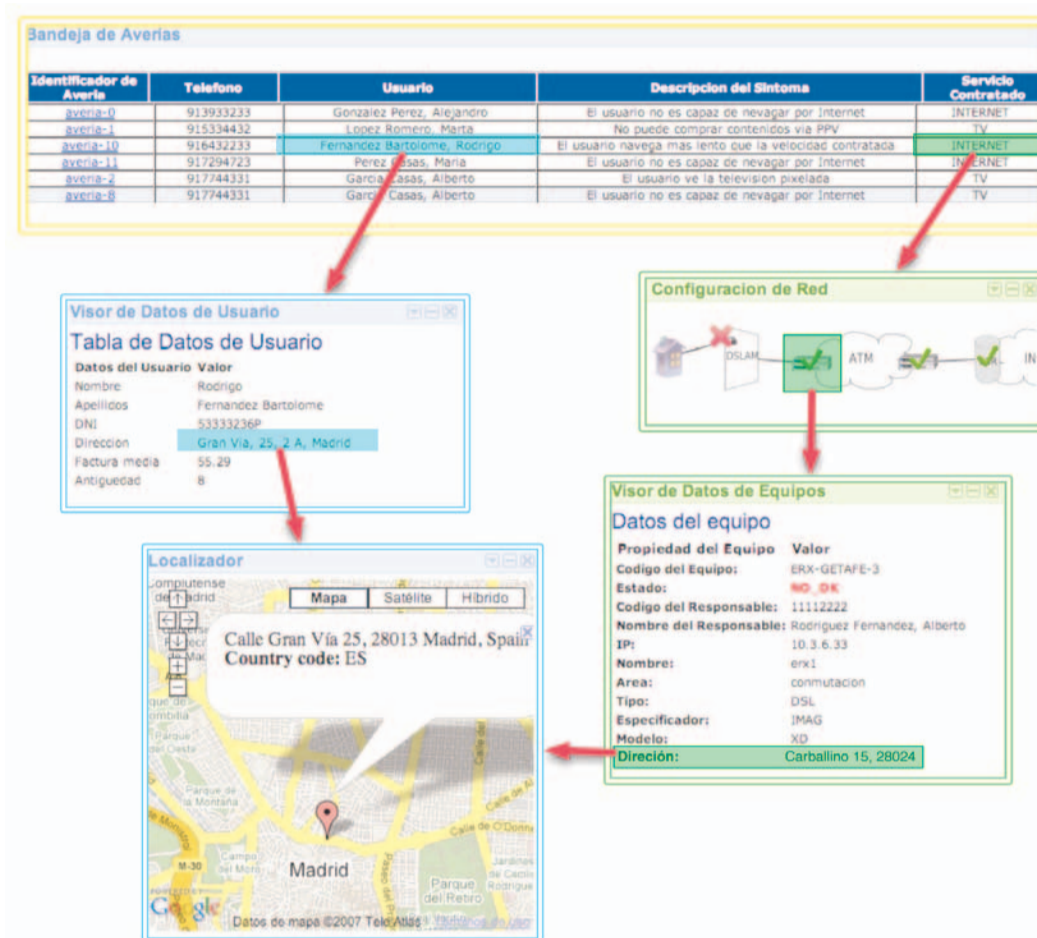
EzWeb - Putting a face on SOA

Target reference architecture and implementation of an **open standard platform** which **manages the frontend access in the next-generation web** and **supports the economics of a resource marketplace**.

Advanced Operation Environment

Users are able to build their **own front-end applications** by combining available resources when they need it and **the way they need it**

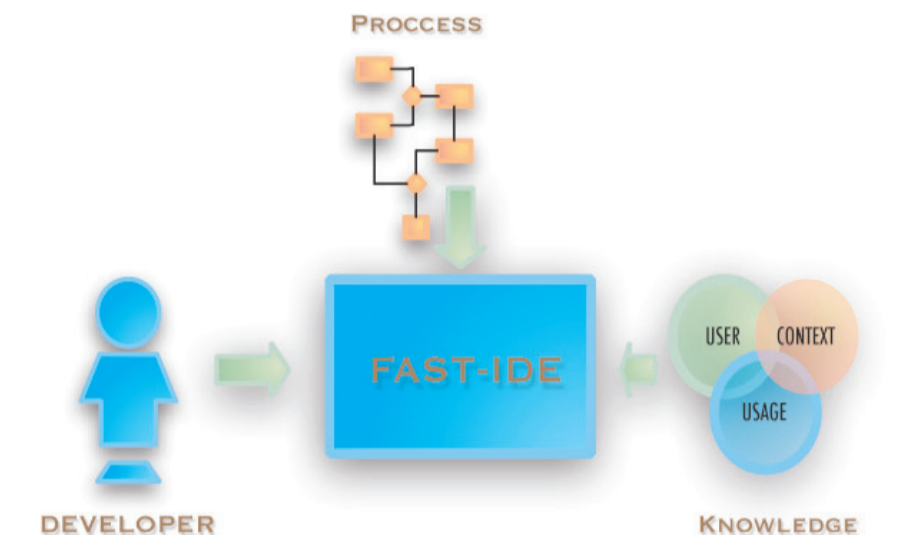
Web 2.0 principles and technologies are used in order to offer flexibility and universal access



Development Environment

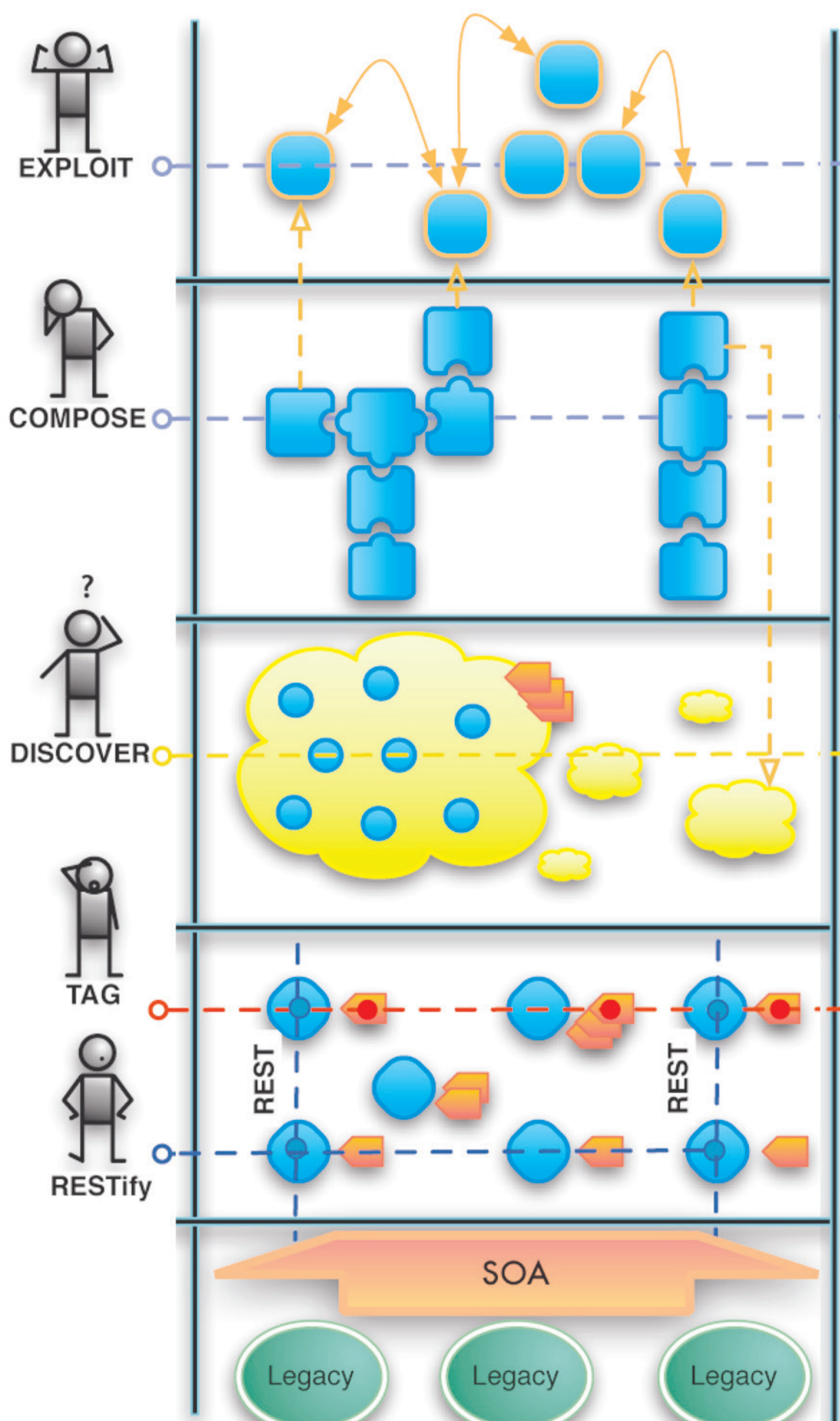
Create complex gadgets and resources combining simpler ones

Piping and ScreenFlow techniques are used to dynamically orchestrate processes and services



Semantic and context information is used to **guide the process**

Global SOA Architecture

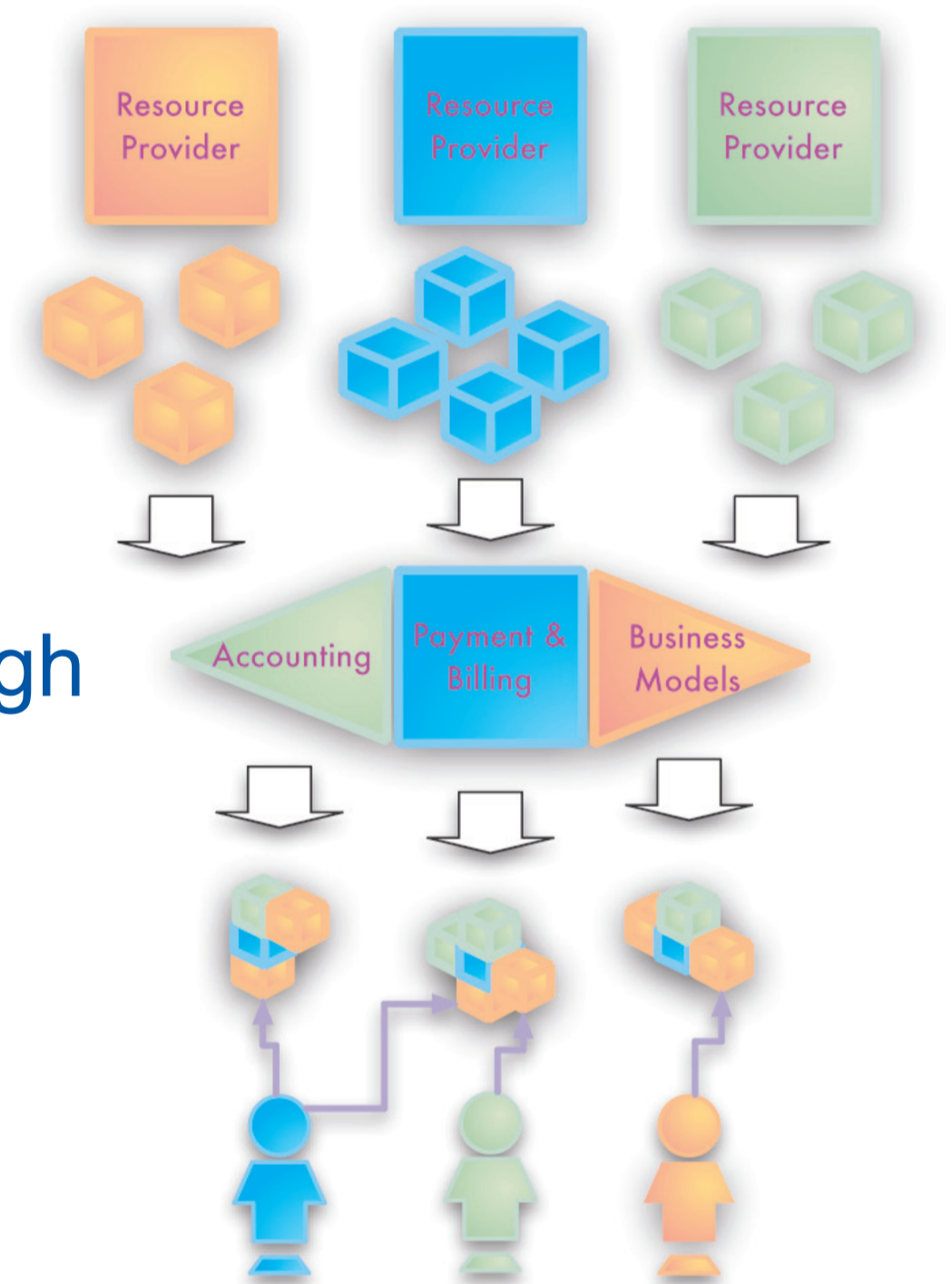


Resource Catalogue and Marketplace

Users can **discover and select** the resources that **best fit their current needs**

Resource providers can add new resources that cover all aspects related to the accounting, auditing payment and billing of the services.

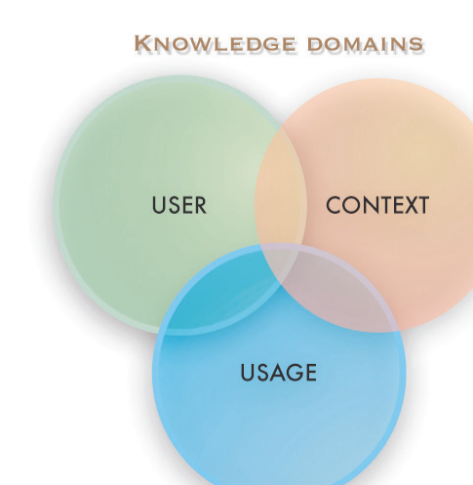
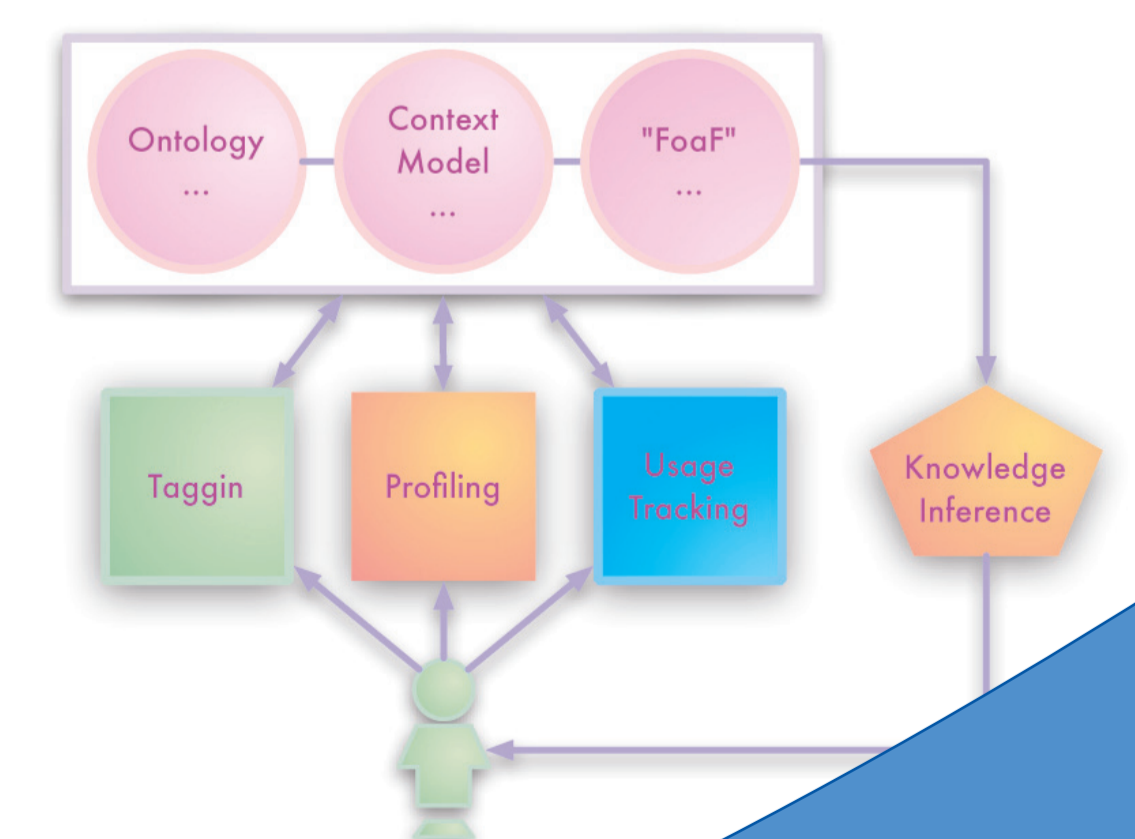
Ultimately the **catalogue is a space where resources can evolve** through combination with other resources



Knowledge capture and exploitation

Lightweight knowledge representation techniques are used: tagging, profiling...

Knowledge is leveraged through sharing and advanced inference mechanisms



Vision
 Creation of a new visual programming environment following a user-centric approach that 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.

Objectives

New visual programming environment for the development of complex front-end gadgets

- Involves the modeling and execution of relatively complex business processes
- Relies on back-end Semantic Web Services
- Agile software development from a top-down perspective

Exploitation of the full power of **semantics for integrating back-end with front-end** (user-centric)

- Developers will start from the front-end gadgets that the user will see and interact with
- Connection to back-end Web Services going through business execution flows if necessary

Plugin Tool for Enterprise Mashup Platforms

- Conceived as a tool that can be compatible with existing and future Enterprise Mashup Platforms
- Tool that enables to develop Mashup-able gadgets which rely on screen-flow Resources and Semantic Web Services stored in a (semantically enriched) catalogue

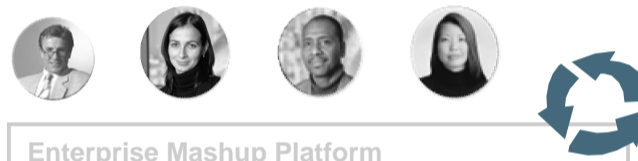
Consortium

	Telefónica Investigación y Desarrollo (Coordinator) Telecommunication Provider	
	SAP AG World's leading Enterprise Application Provider	
	University of Ireland, Galway Leading research institution in Semantic Web	
	University of Kassel Research experience in scenario-based methodologies	
	Cyntelix Corporation Semantic Web technology and collaboration platform	
	Universidad Politécnica de Madrid Semantic Web and Web 2.0-based technologies	

Architecture and Components

Conceptual Model

- Conceptualization of complex gadgets having an associated workflow and possibly human interaction
- Specification of a new ontology to conceptually model to specialise their development to solve concrete user problems
- Definition of screen flows and gadget templates in a visual way
- Definition of mechanisms generating automatically new gadgets from previously developed ones or parts of them

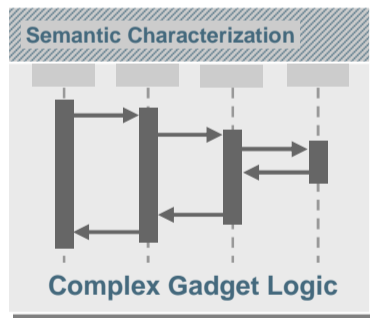


User-Centric Collaboration

- Agile software development from a top-down perspective
- Highly visual integrated development environment
- Continuous improvement

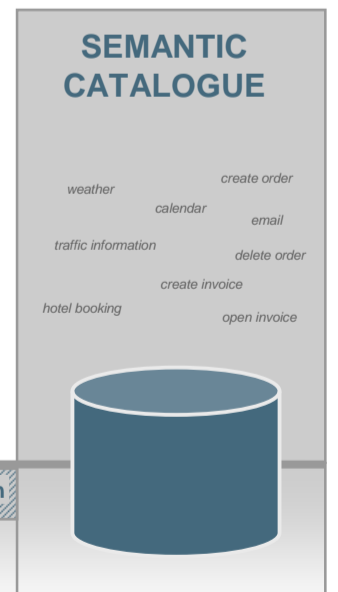
Visual Design of Complex front-end Gadgets

- Tools for the development of complex gadgets (ISO 9000 compatibility)
- Build the different screens composing the gadgets
- Design and implement the flow between these screens
- Define directives and specifications that assure the development and compatibility of the gadget developed in any „gadget enabled“ platform



Visual Composition of screen-flow Resources and Interoperability with back-end Web Services

- Bridge the gap between the ideas managed by user at a user-oriented level on the screen-flow and the underlying resource available in the back-end as Web Services and Resources
- Allow the automatic discovery of services and their friendly use in the visual tool



Benefits

Software industrialization

- Putting a visual face to Service-Oriented Architectures (SOA)
- Lightweight service consumption

Covering the profitable “Long Tail” in Enterprise Application

- User empowerment from service consumer to producer („prosumer“)
- Development of individual gadgets according to user requirements

Faster to market

Semantic Catalogue of screen-flow Resources and back-end Web Services

- Definition of mechanisms to connect gadgets to Semantic Web Services
- Mediation between gadgets ontology and other Semantic Web Services ontologies
- Discovery of Semantic Web Services based on user requirements and capabilities
- Automatically endow gadgets with desired data and functionality

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

Juan J. Hierro, Till Janner, David Lizcano, Marcos Reyes, Christoph Schroth, Javier Soriano

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 that makes up that *Web of Services* and shift the roles and characteristics of resource consumers, providers and intermediaries considerably. A novel platform architecture is presented that builds on all the key technical enablers and has the potential to drive the development of a global, user-centric SOA. Existing prototypes of this platform will be shown as a proof-of-concept.

Index Terms—SOA, Web Services, Enterprise Mash-up, Web 2.0, Global SOA, Internet of Services, user-centric SOA, Web of Services, Program by Composition, lightweight program models.

I. 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

Manuscript sent October 16, 2007. This work is supported in part by the European Commission under The Seventh Framework Program (Project FAST, INFSO-ICT-216048) and by the European Social Fund and UPM under their Research Personnel Training program.

Javier Soriano and David Lizcano are with Universidad Politécnica de Madrid, Campus Montegancedo s/n, 28660 Boadilla del Monte, Madrid, Spain (e-mail: jsoriano@fi.upm.es; dlizcano@fi.upm.es). Javier Soriano is an IEEE Member, and the contact author.

Juan J. Hierro and Marcos Reyes are with Telefónica I+D, Emilio Vargas 6, 28043, Madrid, Spain (e-mail: jhierro@tid.es; mru@tid.es).

Christoph Schroth and Till Janner are with SAP Research CEC St. Gallen, Switzerland (email: christoph.schroth@sap.com; till.janner@sap.com).

boundaries at present [18], and the global provision and consumption of services over the Internet is still at an early stage [7]. 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 choreographies, 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.

II. WEB SERVICES-BASED SOAs AND THEIR SHORTCOMINGS ON THE ROAD TOWARDS A REAL WEB OF SERVICES

Many definitions of SOA have been published since the late 1990s [1]. The widely accepted normative OASIS Reference Model for SOA defines SOA as "...a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations" [10]. According to this model, the major components of a basic SOA and their possible interactions are: a service provider publishes its service interface via a service registry where a service requester can find it and subsequently may bind to the service provider. The central concept of the SOA Reference Model is the existence of services that provide access to capabilities by well-defined interfaces to be exercised following a service contract with constraints and policies. This enables a loose coupling of services (thereby minimizing mutual dependencies) and complies with some of the probably best-known software engineering principles: information-hiding and modularization. Services are provided by entities, the service provider, and are to be used by others, the service consumers.

Services may be composed on the basis of other, existing services, thereby adhering to the principle of reuse. They are autonomous (solely control the logic they encapsulate), uniformly described and publicly retrievable via certain discovery mechanisms. Nowadays, IT and Internet evolution has stated that these services should be more easily deployed by

providers, as well as accessed and managed by end users than traditional packaged software [20]. Therefore, **services must be accessible for all users** (not only enterprise stakeholders), because these services should replace a previous wide range of enterprise software [7],[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, represents an architecture focused fundamentally on a B2B context 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 begin 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.

III. 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 [9], 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.

A. *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 implementations. 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.

B. Key technological principles

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.

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.

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

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 [20]. 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 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 [20]. 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.

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 [8].

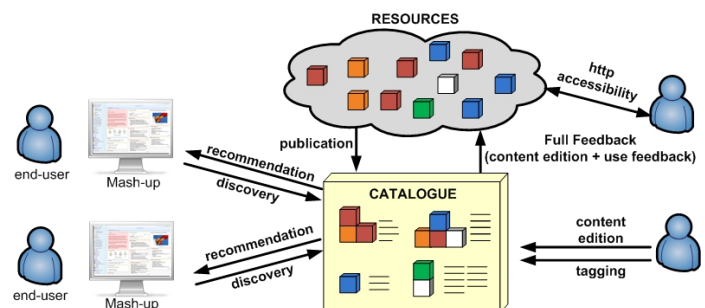


Fig. 1. Cataloguing Resources

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.

IV. 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.

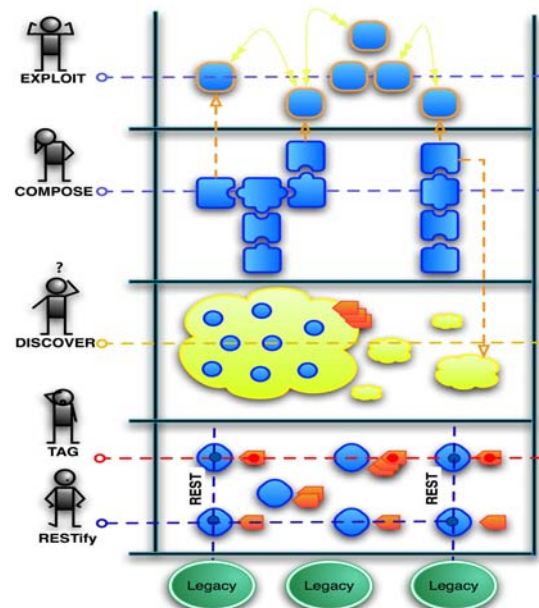


Fig. 2. User-centered SOA solution stack

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. This approach proposes the conceptualization of items like, for example:

- Interaction models: The user interface constitutes a bidirectional communication channel between an end user and a system or machine. The interaction model between these parties needs to be conceptualized to assure proper communication between the user and the machine. This is done by conceptualizing the different interaction models (direct action, event-based, publish and subscribe, etc.) present today and even proposing new ones stemming from the use of semantics in the front-end.
- Internal logic flow: A user interface is just appearance without connecting it with data, resources and functionality. One important aim would be to conceptualize the gadget's internal logic flow, as well as its interaction with semantic Web services. For example, once conceptualized, a gadget action could lead to the execution of a more or less complex logic flow, including invocations to existing semantic Web services to complete a task. 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.

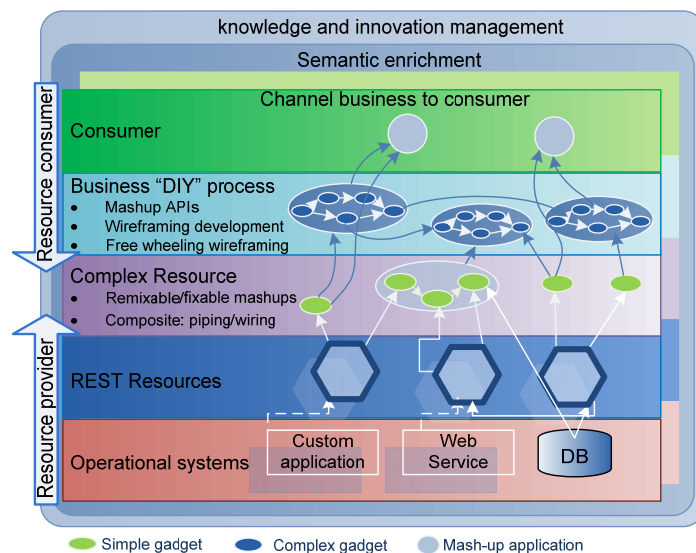


Fig. 3. Logical layers in the proposed approach

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 **automatically**

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.

V. 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.



Fig. 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

VI. 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).

VII. CONCLUSION AND FUTURE TRENDS

In this paper, we elaborated on the traditional Web Services-based approach to SOA as a traditional enabler for integrating distant and potentially heterogeneous Web-based resources. As Web Services have neither lived up to their promise of facilitating a global network of loosely interconnected services nor provided a appropriate front-end for people, new approaches are required to rise to this challenge of enabling a Web of Services. As a key component of the Web 2.0 era, novel technologies and design principles are now about to emerge that will allow human users to use, customize, combine, interconnect and finally display Web-based content or functionality as new resources on the Web. In this paper we have elaborated on the synergies between the Web 2.0 and the SOA worlds that can be exploited to rise to this important challenge and have proposed a generic model of a global user-centric SOA and a novel architecture for enterprise mash-up platforms.

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 [12] 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].

REFERENCES

- [1] Alonso, G., Casati, F., Kuno, H. & Machiraju, V.(2004). *Web Services Concepts, Architectures and Applications*. Springer, 2004
- [2] Anderson, C.(2006). *The Long Tail, Why the Future of Business is Selling Less of More*. Hyperion. July 2006.
- [3] Coveyduc, J., Huang, C., Ostdiek, L. & Reif, J.(2006). *IBM Innovation Factory: An integrated solution for accelerating innovation*. IBM, October 2006. Available at <http://ibm.com/websphere/developer/zones/hipods>.
- [4] Davenport, T. H.(2005). *Thinking for a Living: How to Get Better Performance and Results from Knowledge Workers*. Harvard Business School Press, Boston, MA, USA, 2005.
- [5] Evans, P. & Wurster, T. S.(2000). *Blown to Bits: How The New Economics of Information Transforms Strategy*. Harvard Business School Press, Boston, MA, USA, 2000
- [6] Fielding, R. T.(2000). *Architectural styles and the design of network-based software architectures*, Ph.D. thesis, University of California, Irvine, 2000
- [7] Gartner Inc. (2006). *Hype Cycle for Software as a Service*, Gartner Research, 10 August 2006.
- [8] Giles, J.(2005). *Internet encyclopedias go head to head*. *Nature*, Vol.438, No.7070 (pp.900-990), December 2005.
- [9] Högg, R., Meckel, M., Stanoevska-Slabeva, K. & Martignoni, R.(2006). *Overview of business models for Web 2.0 communities*, *Proceedings of GeNeMe 2006*, p. 23-37, Dresden, 2006.
- [10] MacKenzie, M. (2006) *OASIS - Reference Model for Service Oriented Architecture 1.0*, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm
- [11] McAfee, A.(2005). *Will Web Services Really Transform Collaboration*, *MIT Sloan Management Review*, Vol.46, No.2, 2005
- [12] McAfee, A.(2006). *Enterprise 2.0: The Dawn of Emergent Collaboration*. *MIT Sloan Management Review*, Vol.47, No.3 (pp. 21-28).Spring 2006.
- [13] North, D.C.(1990). *Institutions, Institutional change and economic performance*, Cambridge University Press, Cambridge, 1990
- [14] O'Reilly, T.(2005). *What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software*. www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html
- [15] O'Reilly, T. & Musser, J.(2006). *Web 2.0 Principles and Best Practices*. O'Reilly radar, November 2006.
- [16] Roman, D. (2005). *Web Service Modeling Ontology*, *Applied Ontology*, Vol.1, No.1 (pp. 77 - 106), 2005.
- [17] Schroth, C. & Christ, O. (2007). *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)* (pp. 8): IEEE Computer Society. Retrieved 2007-07-11, from <http://www.alexandria.unisg.ch/publications/37038>.
- [18] Schroth, C. & Janner, T. (2007). *Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services*. *IEEE IT Professional* Vol.9, No.3(pp.36-41) , June 2007.
- [19] Smith, R.(2006). *Enterprise Mashups: An Industry Case Study*. Keynote at the New York PHP Conference & Expo, Manhattan, New York, USA, 14-16 June 2006
- [20] Winewright, P. (2005) *Why Microsoft can't best Google*, *Software as a Service ZDNet editorial*, August 2005. Retrieved September 18 2007, from <http://blogs.zdnet.com/SAAS/?p=13>

Fostering Innovation in a Mashup-oriented Enterprise 2.0 Collaboration Environment

Javier Soriano¹, David Lizcano¹, Miguel A. Cañas², Marcos Reyes² and Juan J. Hierro²

¹ Universidad Politécnica de Madrid, Campus Montegancedo s/n
28660 Boadilla del Monte, Madrid, Spain

² Telefónica I+D, Emilio Vargas 6, 28043 Madrid, Spain
Email: {jsoriano; dlizcano}@fi.upm.es; {mac, mru, jhierro}@tid.es

Abstract: A number of Enterprise 2.0 Collaboration platforms are beginning to proliferate. By leveraging desirable Web 2.0 attributes, these platforms provide enterprises with an ecosystem of employees, partners, suppliers, and customers who collaborate to develop capabilities by collectively generating, sharing and refining business knowledge. This paper advocates the idea that enterprise collaboration must evolve towards a new paradigm in which knowledge workers 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. 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. This architecture will help to evolve the above ecosystem into one in which all the stakeholders will also be able to collaboratively develop capabilities and innovate operating procedures by remixing and integrating already available services through the core ideas of enterprise mashups and collaborative resource repositories.

Keywords: Enterprise Collaboration, Enterprise 2.0, Web 2.0, Knowledge Work, 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, Python, 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 were 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. Google IG 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.

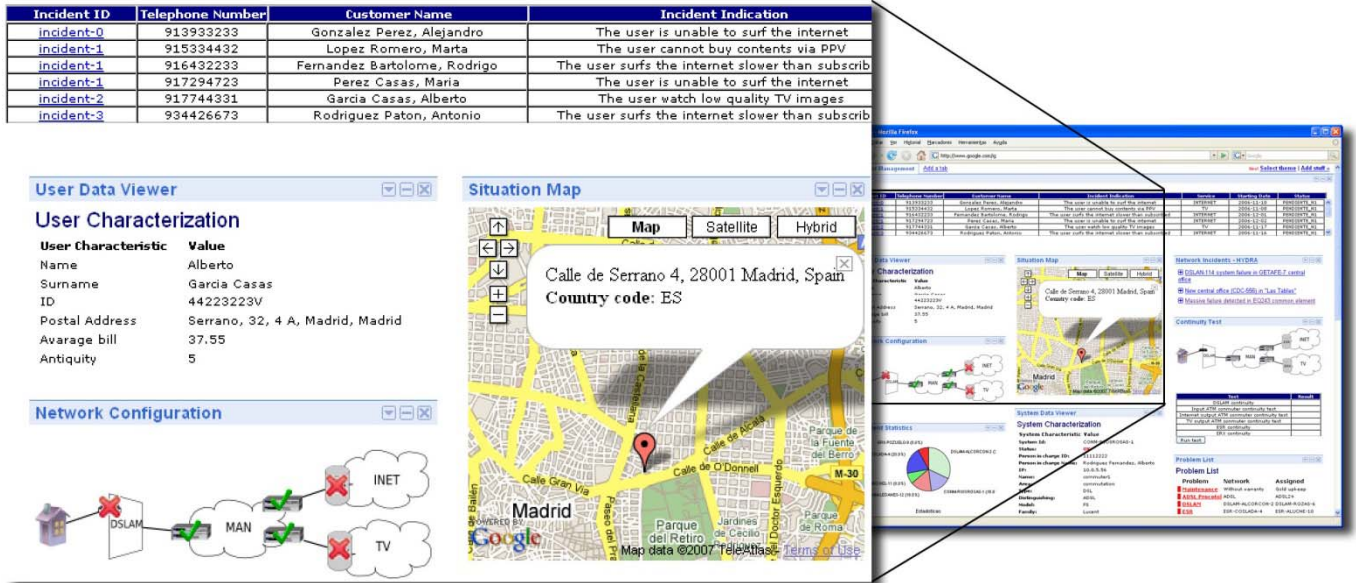


Fig. 1. A real world EzWeb enterprise mashup currently deployed at Telefónica OSS as a fully operational environment.

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 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. 2 and are further explained in the following subsections.

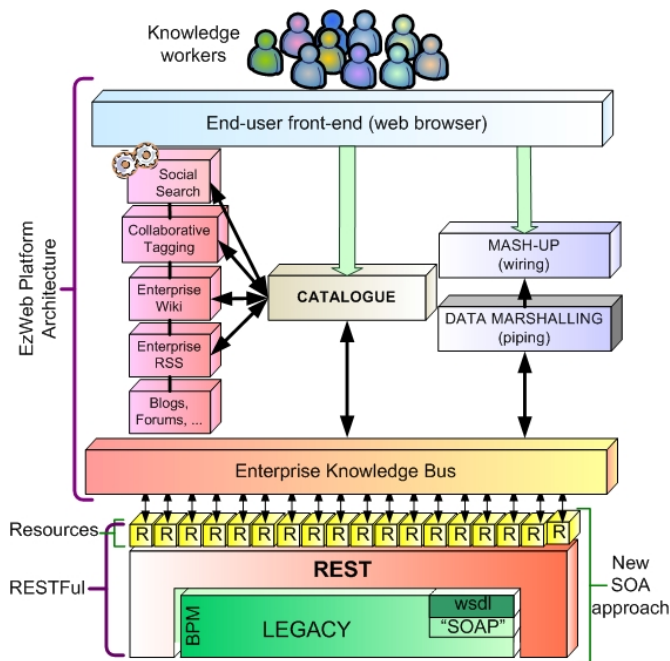


Fig. 2. EzWeb architecture.

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.

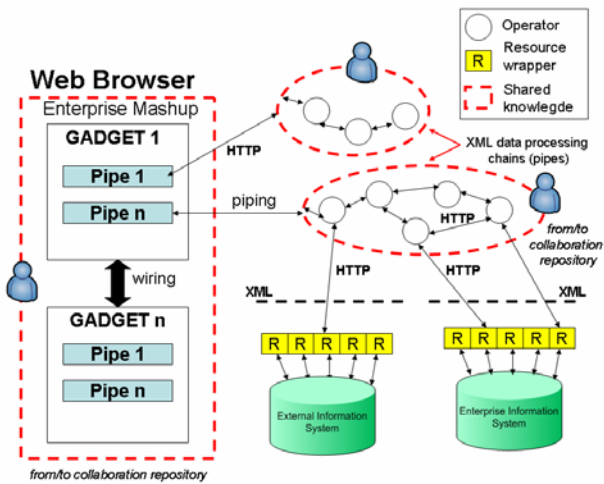


Fig. 3. Resource composition in enterprise mashups.

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.

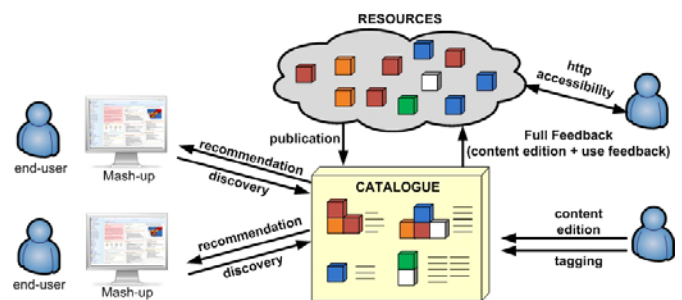


Fig. 4. Cataloguing resources.

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.

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.

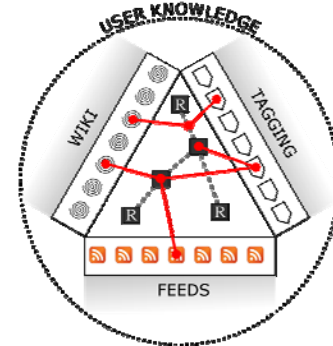


Fig. 5. Collective knowledge emergence through a catalogue.

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.

Acknowledgement

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.

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.suitedtwo.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/yaho-o-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.

Notes

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.

Brave New Web: Emerging Design Principles and Technologies as Enablers of a Global SOA

Christoph Schroth^{1,2}, Oliver Christ²

¹ *University of St. Gallen, =mcm Institute, St.Gallen, Switzerland*

² *SAP Research CEC, St.Gallen, Switzerland*

{christoph.schroth, oliver.christ} @ {sap.com}

Abstract

Web Services have experienced great interest during the last years as they were expected to play a key role as enablers of seamless application-to-application integration both within company boundaries and on a global, cross-organizational scale. As a technical foundation for the realization of Service-Oriented Architectures (SOAs), Web Services encapsulate complexity inherent to individual applications and allow for their loose coupling. However, a truly global mesh of such services has not yet become reality due to various reasons. Novel technologies and design principles are currently about to emerge which allow human users to use, customize, combine, interconnect and finally expose Web-based content or functionality as new resources which are often referred to as Mash-ups. In this article, we provide an overview of existing Mash-ups as well as tools and platforms that empower users to build them in a highly efficient and intuitive fashion. Statistical data and case studies are leveraged to examine new ways of resource provision and consumption and also the relevance of upcoming intermediaries. Finally, we investigate remaining research challenges on the path to a truly global SOA.

1. Introduction

The number of enterprises that automate cross-organizational business transactions is constantly growing. Renowned scientists such as Malone cite the relentless march of improvements in the cost-performance ratio of information technology as main driver of this development [1]. Benefits of performing transactions electronically include extending market reach, saving time, cutting costs and responding to customer queries more agilely [2].

Web Services [3] have experienced great interest as they were expected to play a key role as enablers of this seamless application-to-application integration both within company boundaries and on a global, cross-organizational scale. They allow for encapsulating proprietary application functionality with the help of

uniformly defined interfaces and thus facilitate the loose and quick coupling of potentially heterogeneous applications.

As argued in [4], **Service-Oriented Architectures (SOAs)** [5] on the basis of Web Services mostly only exist within company boundaries and have not yet taken over a significant role in realizing cross-organizational collaboration. Various reasons such as high technical complexity, implementation and maintenance costs, inflexibility and the lack of widely accepted standards for defining service choreographies as well as message semantics have prevented the emergence of a global mesh of interoperable Web services.

Novel technologies and design principles are now about to experience increasing acceptance as they allow for using, customizing, interconnecting and finally exposing Web-based content or functionality again as new resources. **Web 2.0** is frequently used as an umbrella term to summarize the collection of programming languages and general principles that can be considered not as substitute, but as enrichment of SOA concepts and technologies.

Numerous various definitions of Web 2.0 exist that mostly only describe certain aspects of the overall concept. Tim O'Reilly, who originally coined the term, identifies seven major characteristics inherent to the Web 2.0 concept [6]. First, the Web is considered as a platform for building systems that do not necessarily have a specific owner and are "tied together by a set of protocols, open standards and agreements for cooperation". Harnessing collective intelligence of Web users represents the second major paradigm. The ownership of mission-critical data is regarded a further cornerstone of numerous Web 2.0 applications. Fourth, O'Reilly propagates the end of the software release cycle as another central paradigm. The use of lightweight programming models that allow for loosely coupled systems and applications, the provision of software above the level of a single device and the realization of rich user experience represent the last major paradigms inherent to the concept of Web 2.0. Besides such analyses that properly describe parts of the super-ordinate concept, there are only very few comprehensive scientific definitions available. An in-depth investigation of numerous different, successful Web

2.0 applications performed by Hoegg et. al. [7] condensed the respective characteristics in the following statement which works as underlying definition for this work: “Web 2.0 is defined as the philosophy of mutually maximizing collective intelligence and added value for each participant by formalized and dynamic information sharing and creation”.

Besides social communities (e.g., LinkedIn.com, GiveMeaning.com) or online collaboration tools, **Mash-ups** are considered as central to the Web 2.0 era. *The Economist* defines Mash-ups as follows [8]: “The term Mash-up is borrowed from the world of music, where it refers to the unauthorised combination of the vocal from one song with the musical backing of another, usually from a completely different genre. Web Mash-ups do the same sort of thing, combining websites to produce useful hybrid sites and illustrating the internet’s underlying philosophy: that open standards allow and promote unexpected forms of innovation.” The *BusinessWeek* [9] explains this novel form of using Web-based resources similarly: “Mash-ups portend big changes for software companies, Web sites, and everyone online. No longer just a collection of pages, the Web is morphing into a sort of global operating system, à la Microsoft [...] Windows. And now, people are learning to program Web 2.0 with much of the same innovative energy of the personal computer’s early days.”

We basically adhere to these explanations and take the following summary as foundation for this work: *A 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.*

Mash-ups can be setup by using resources such as the OpenStreetMap¹ service, which provides free geographical data such as street maps. By leveraging lightweight protocols and a simple interface, it can be seamlessly integrated into other Web-based applications and thus be enriched with additional, arbitrary features such as pictures of local buildings or tourist attractions. Platforms and tools are provided in the Web that empower users to loosely couple readily available applications, to enrich and compose them into a novel service, which can again be made publicly available. As a result, meshes of different Web-based resources emerge that can be considered rather ungoverned since there is no central control entity existing that imposes formal guidelines for reuse and syndication. The possibility to actively contribute and gain reputation by publishing resources on the one side and to harness the aggregated knowledge and intelligence of all other platform participants on the other side represent core benefits for users.

Recently, the relationship between Mash-ups and traditional SOAs has experienced considerable interest

[10, 11]. In this article, we investigate this interrelation and propose Mash-ups as key enablers of a truly global SOA that is freely developed by human users who are fascinated and empowered by a set of Web 2.0 technologies and design principles.

In the following sections, we first of all provide an exact definition of SOA and also elaborate on the shortcoming of conventional, Web Services-based implementations with regard to the emergence of a global mesh of interoperable services (Section two). We then use the key three elements of a SOA, namely the service provider, the service consumer and a registry [5] to systematically analyze success factors of the above mentioned Mash-ups on the basis of statistical data and case studies (Section three). As argued below, new ways of resource provision and consumption as well as certain intermediaries are characteristic for the emergence of Mash-ups which can be considered new and human-driven forms of SOAs.

2. Vision and shortcomings of the traditional Web Service-stack to enable a global SOA

2.1. SOA and the vision of loosely coupling heterogeneous applications worldwide

Since the late 1990s, many definitions of SOA have been published [3, 12]. The normative OASIS Reference Model for SOA [5] defines SOA as “...a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations”.

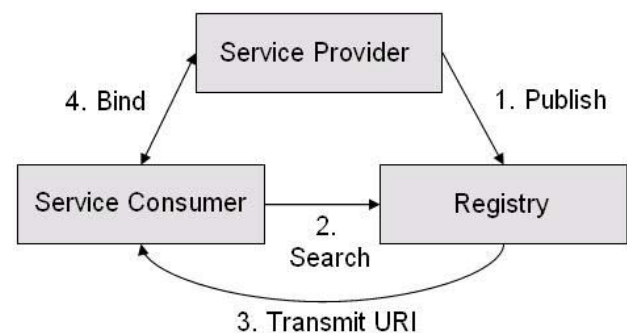


Figure 1: Basic SOA Architecture [5]

According to this model, the major components of a basic SOA and their possible interactions are (see Figure 1): a service provider publishes his service interface via a service registry where a service requester/ consumer can find it and subsequently may bind to the service provider. The central concept of the SOA Reference Model is the existence of services which provide access to capabilities

¹ <http://wiki.openstreetmap.org/index.php/REST>

by well-defined interfaces to be exercised following a service contract with constraints and policies. This enables a loose coupling of services (thereby minimizing mutual dependencies) and complies with some of the probably most-known principles in software-engineering, information-hiding and modularization [13]. Services are provided by entities, the service provider, and are to be used by others, the service consumers. Services may be composed on the basis of other, existing services, thereby adhering to the principle of reuse. They are autonomous (solely control the logic they encapsulate), uniformly described and publicly retrievable via certain discovery mechanisms.

The concept of supporting loosely coupled, business-aligned and networked services as introduced above can be realized with the help of numerous different technologies. As **WSDL**² and **SOAP**-based Web Services are the most widely spread standards used to setup SOAs, these are in the focus of this work [3]. The Web Service Description Language (WSDL) defines a uniform, machine-readable XML format for service interfaces. The Universal Description, Discovery, and Integration (**UDDI**³) standard specifies publicly available service registries (see Figure 1) that are needed for service search, identification and invocation. SOAP specifies the data format and an exchange protocol for the messages to be sent between service providers and requesters. While SOAP only defines formal aspects of data structure, the interpretation of information semantics has not yet been defined by a common standard. Machines may describe the same piece of information with different terms, thereby preventing from seamless system interoperability. The Business Process Execution Language (**BPEL**⁴) is widely accepted as a standard for imposing a choreography on different services. Also based on XML, it focuses on the description of a business process from one participant's point of view.

Many have generated great enthusiasm about Web Services, which are considered as "open, Internet-era standards for exchanging data between applications" [4]. With a SOA, tight couplings between the applications of different companies are expected to be replaced by loose couplings. As a consequence of common standards for data description and connection protocols, applications can be seamlessly connected to others without costly reprogramming efforts. By loosely coupling application functionality via commonly defined interfaces, companies are enabled to modify their operations more agilely in response to changing business requirements. Summing up, especially companies expected the dawn of Web Services to turn traditional supply chains inside out and to replace static cross-organizational business relations by a

flexible mesh of loosely coupled services which can be identified via publicly accessible registries.

2.1. Shortcomings of the conventional Web Services stack with regard to realizing a global SOA

Right now, Web services have not lived up to this promise as mostly people are still involved in loose couplings all over the world. Rather than automatically inter-connecting applications across company boundaries with the help of Web Services technologies, most loose couplings are based on "e-mail and instant-messaging applications, groupware such as Lotus Notes/ Domino and nascent "social software" such as wikis, Friendster.com and LinkedIn.com" [4]. As argued in [14], the "corporate household" problem prevents Web Services-based SOAs from being setup quickly and easily. Companies and their respective systems are subject to rigid governance mechanisms [15] and adhere to different conventions with respect to business documents and also internal business processes: "The applications used in different departments or different locations of a single company, for example, are virtually guaranteed to have inconsistency in customer lists, part-numbering schemes, approved suppliers, addresses for these suppliers and so on" [4].

Besides the heterogeneity of conventions with respect to data structures, semantics and processes, the lack of comprehensive, trustworthy and widely accepted service registries prevents global SOAs from quickly establishing. The few globally available UDDI-based registries are mostly only usable for technical experts, do not offer any additional functionality such as service performance monitoring and only feature a small number of references to different Web Services.

Apart from that, the traditional Web Services stack aims at supporting the setup of loosely coupled application interconnections especially in a professional context and assumes users to be technically sophisticated. This prevents normal Internet users from easily discovering and interacting with services available via the Web. WSDL-based service interfaces, but also the UDDI-compliant service registries are not designed to be interpreted and used by humans but rather by machines.

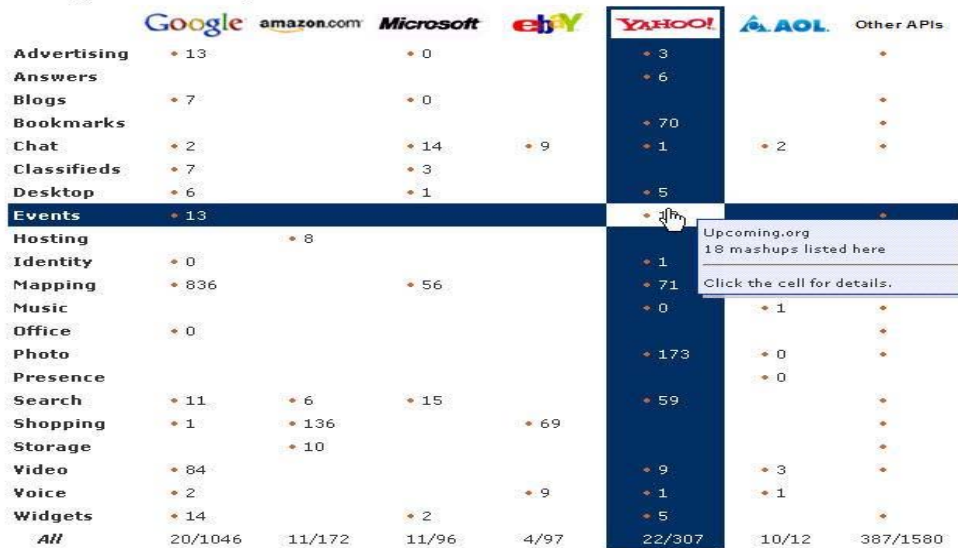
2. Web 2.0 design principles and technologies as enablers of a global SOA

² <http://www.w3.org/tr>

³ <http://www.oasis-open.org/>

⁴ <http://www-128.ibm.com/developerworks/>

Figure2: Mash-up API Scorecard



After elaborating on the actual vision and the promise of traditional Web Services to allow for the efficient setup of loosely coupled application interconnections, we present emerging design principles and technologies which enable a global Service-Oriented Architecture which is driven by individual Web users and their requirements. By introducing new ways for **resource provision and consumption**, but also as a consequence of the emergence of **new intermediaries**, the major obstacles which traditional Web Services struggled with can be overcome. In this way, Mash-ups can be freely designed by individuals on the basis of lightweight programming models and interfaces. The following sections will thoroughly investigate the prerequisites for the success of these Mash-ups and thereby leverage statistical data as well as case studies.

3.1. Resource provision

As mentioned above, traditional Web Services are provided as functionality described by uniform interfaces (mostly WSDL compliant) that define input and output messages as well as the supported functions of the services. These interfaces are not human-readable and do not facilitate the interaction of users with the underlying services. Resources in the Mash-up context do not aim at technical experts in the corporate context any more, but at the huge number of individuals, the “long tail” [16] of Internet users who require intuitive visual means for retrieving resources in the Web and for capturing their respective functionality.

The Website **ProgrammableWeb.com** acts as a major aggregator of numerous (currently more than 1600) references to diverse existing Mash-ups all over the

world. It provides statistics about used resources, a classification of Mash-ups into categories as well as user statistics to evaluate their popularity. Therefore it serves as one central source of empirical data used in this work.

To investigate the most frequently used resources and their respective Application Programming Interfaces (APIs), the Web Site exposes an “API-Scorecard” (Figure 2) which has six major API-providers on the X-axis and the most relevant categories of APIs offered on the Y-axis. In this context, the term API refers to the interface type, data formats and the protocol to be used for interconnecting to an application. The last line of the figure lists the number of APIs referenced on the Website and the number of Mash-ups using one of these APIs for each of the 6 major providers. For example, 20 Google-based APIs and 1046 Mash-ups that leverage one of these APIs are listed on ProgrammableWeb.Com.

Several facts can be inferred from this graphic: First, large companies such as Google, Amazon and Microsoft start to successfully provide Web-based resources which are leveraged by numerous users for creating Mash-ups on their basis. As can be seen in Figure 2, at least 836 Web-based applications have already integrated the “Google Maps”- resource that offers geographical data. The Website “Vuelta a California” can be cited⁵ as an excellent example for a Google Maps-based Mash-up. It uses information from a Web log that deals with road profiles for bikers in California and mashes it with road map data from Google Maps to achieve a rich visual representation.

The second important fact that can be inferred from the API scorecard is the shift away from the professional corporate context towards a private, end-user driven field

⁵ <http://www.berndstrahle.com/map.html>

of applications. As opposed to the focus of traditional Web Services, the resources used for building Mash-Ups aim at the long tail of Internet users and deal with media management, shopping functionality, entertainment or desktop applications. Heavy-weight enterprise applications for automating business transactions can only rarely be found in this context.

Third, the Website also provides information about the protocols and interface standards used by the different listed resources. The analysis of the provided data leads to the following results: Besides the **SOAP** protocol which is also used in the context of traditional Web Services, particularly “light-weight” protocols such as REST [17] and RSS are widely leveraged to allow for fast and seamless mashing of different resources.

The Representational State Transfer (**REST**) is an architectural style that has been proposed by Fielding in his doctoral dissertation. With the help of REST, Web clients can interact with arbitrary Web resources (sources of specific information) in a uniform way and thereby exchange XML-based messages. Resources are uniquely addressable by means of a global identifier (URI) and support a well defined and also constrained set of supported operations (so-called HTTP verbs such as GET, PUT). By using resource identifiers and HTTP verbs together, clients can invoke REST-based services and request the delivery of, for example, an XML file containing the desired specific information. The Simple Storage Service (S3⁶) provided by Amazon is one of many exemplary applications of REST technology: On the basis of a simple service interface, users are enabled to store any amount of data on Amazon servers and also retrieve it from arbitrary locations on the Web. REST incorporates the principles of complexity-hiding and uniformity since it significantly reduces programming efforts and forces both clients and recourses to adhere to a common set of supported operations and interface descriptions.

The Really Simple Syndication (**RSS**⁷) format represents a further technology that is affiliated with numerous resources in the Web 2.0 context. An XML based file format, RSS supports the easy aggregation of content from arbitrary sources in the Web. With the help of this standard, the information offered by frequently changing Web sites such as Web logs can be exported into XML files of a uniform structure and is thus made publicly available. The files’ structures are determined by the RSS specification which intentionally only allows for a very limited set of content categories such as title, Web link and textual description of a piece of information. Web applications that aim at distributing content via RSS technology must ensure that the respective data files always contain the latest information. Applications that

present the information obtained from different RSS-based sources in a combined manner are referred to as content aggregators. Users are thus enabled to gather information that has been created by a number of authors and is easily accessible via uniform interfaces, redesign it and also publish it.

Summing up, the Web increasingly can be considered as a comprehensive and global development platform containing numerous easily usable and mashable resources that are provided by large firms as well as by individual end-users. As argued above, the provision of resources that use lightweight protocols such as REST or RSS and also the focus on end-user requirements rather than Business-to-Business relationships represent core success factors for this new global SOA consisting of numerous Mash-ups. Figure 3 finally shows the currently most popular providers of resources leveraged for the creation of Mash-ups.

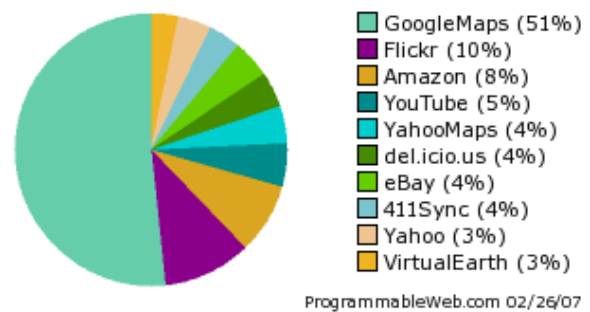


Figure 3: List of most popular Mash-up resource providers (according to ProgrammableWeb.com)

3.2. Emerging intermediaries

As presented above, a publicly available registry constitutes one of the three major elements of a SOA. UDDI is widely accepted as standard and offers a platform-independent, XML-based registry specification for stakeholders worldwide to list the services they provide via the Web. The key information published by such a registry contains white pages (address, contact, and known identifiers of the service providers), yellow pages (industrial categorizations based on standard taxonomies) and finally green pages (technical information about services exposed by the business).

Similar to the technologies used for describing services in traditional SOA implementations (e.g. WSDL), the UDDI standard aims at technical experts and only specifies very limited functionality. Merely UDDI-based registries are not adequate to serve the large number of normal Internet user who require intuitive, human-readable interfaces to identify and to evaluate existing services and to finally integrate and mash them into their applications.

⁶ <http://www.amazon.com/gp/browse.html?node=16427261>

⁷ <http://www.rssboard.org/rss-specification>

Novel forms of intermediaries are currently about to emerge which offer resource registry functionality and extend the role of the traditional UDDI-based implementations. As depicted in Figure 4, they do not only contain references to services, but also offer statistics about resource performance or information about evaluations conducted by previous users.

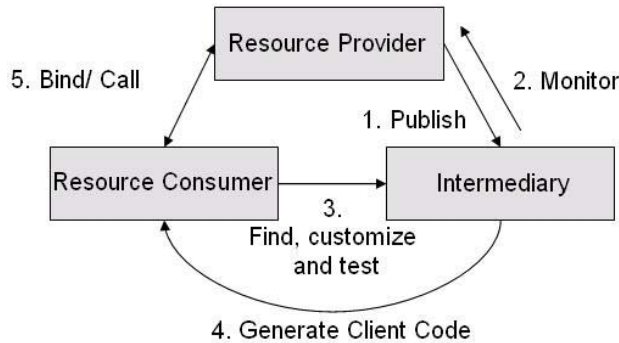


Figure 4: Advanced intermediaries facilitate retrieval and use of Web-based resources

The typical resource discovery and utilization process is therefore different from the one depicted in Figure 1. A resource provider first of all publishes a reference to his resource at the intermediary which stores it and subsequently starts to monitor parameters such as general resource availability and response latency. Potential consumers may then access the intermediary's Website and use the performance metrics and other evaluation results while searching for adequate resources. Some intermediary platforms even allow for building customized code for integrating selected resources into the users' Mash-ups in order to relief them from any programming efforts. The intermediary **WebRPC**⁸, for example, offers this functionality and uses the advertising slogan "Find, customize, monitor, and call SOAP, REST, or HTML services in minutes". After performing these four steps, the user may finally call the customized resource.

Besides such one-stop-shop resource registries, other kinds of intermediaries are about to emerge which all follow the goal to simplify user/ service interaction and thus facilitate the development of a global SOA. The platform **FeedBurner**⁹ represents an excellent example for such intermediaries: It allows users to register RSS-compliant pieces of information ("feeds") and provides users such as bloggers, podcast publishers and other Web-based content creators with several value-added services. After registering the URL of an RSS feed from an arbitrary source on the platform, FeedBurner offers free readership statistics, promotion services that make sure

feed search engines are notified about updates as well as a human-readable and harmonized feed address as a substitute for the mostly heterogeneously designed original addresses. In this way, users benefit from both flexibility and persistence: In case the actual source of a user's feed changes, he can edit the respective entry at the central Feedburner platform and thus ensures that the potentially high number of users who read or even mash the feed with other resources are not affected by this change. For them, the feed's address remains the same. According to the information provided by the platform, as of August 2006, FeedBurner hosts feeds for 250,000 publishers, and delivers feeds to 19 million subscribers.

The emergence of such intermediaries that act as central resource publishing and management platforms can be motivated with help of **institutional economics**. According to North [18], the reason for institutions to exist is the uncertainties involved in human interaction; institutions act as constraints and framework to structure that interaction and make it more reliable. The proper establishment and application of informal constraints, formal rules and enforcement are adequate to lower transaction costs and boost economic performance. Both the WebRPC and the Feedburner platform are recognized as trustworthy central points of access to different kinds of Web-based resources. By monitoring performance and by gathering user-based evaluations, they lower the risk for new users to select a bad performing resource for their Mash-up. They also act in a normative fashion as they provide common standards for naming and format and thus reduce uncertainty of resources to be unavailable as a consequence of a URL change, for example.

3.3. Use of resources

After investigating new principles of offering resources and after presenting novel forms of intermediaries which provide institutional frameworks, we deal with the changes regarding user-side consumption of resources in this section. Before presenting three major cases of tools and platforms that empower users to mash up almost arbitrary kinds of web-based content or application functionality, we first present a Top Ten list of Mash-up categories (Figure 5) as of February 2007 and according to the data gathered by the ProgrammableWeb.com platform which references more than 1600 different Mash-up use-cases. As can be seen in this figure, end-user centric applications which are related to geographical maps, media management or online shopping prevail, while enterprise class Mash-ups do not yet occur very frequently.

Many of the Mash-ups cited and examined above have been created by individuals who are still technically experienced to some degree. Light-weight programming models and protocols as well as intermediaries which support the retrieval and evaluation of resources facilitate

⁸ <http://www.webrpc.com/>

⁹ <http://www.feedburner.com/>

the Mash-up setup procedure. However, some programming effort is still required in those cases. The following three major cases of Mash-up-platforms highlight a rather recent way of even empowering users without any technical sophistication to create and change Mash-ups and to thus foster the emergence of a user-driven, global SOA.

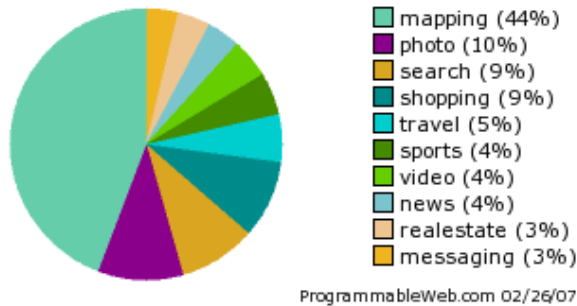


Figure 5: Top ten list of Mash-up categories

3.3.1. Case studies

The **Google Website**¹⁰ represents a first intuitive example for a Mash-up platform as it allows users to aggregate and arrange different “gadgets” which represent either mere content (e.g., pictures, text, weather forecasts, Web cams) or certain application functionality (e.g., calculators). An intuitively usable gadget repository can be accessed for free and enables users to select and subsequently drag and drop some of the gadgets on their individual user interfaces. Finally, individuals are also empowered to create and register gadgets on their own by leveraging freely available templates and by complying with the Google de-facto standard. This kind of Mash-up-platform facilitates the mere aggregation of resources (which is also often referred to as syndication) but does not foresee any kind of interaction between them. Gadget interoperability issues are thus completely avoided since the resulting Mash-ups solely reside on the presentation layer. The gadgets offered by Spring Widgets¹¹ also belong to this category of presentation-layer syndication Mash-ups.

Yahoo pipes¹² represent an example for a second category of Mash-up platforms which are not limited to mere resource syndication. Figure 6 depicts an exemplary use-case for this platform: The “pipe” (which has been designed by a user) reads in an RSS-based feed provided by the New York Times homepage, passes it through a customizable content analysis (marked orange) and uses filtered keywords to find related photos at the widely known media repository Flickr¹³. The pipe’s output issues

links to the photos in a RSS-compliant way. Mash-up platforms like this are different from the ones assigned to the first category as they involve the processing of resources rather than their mere syndication. Content from the New York Times website is filtered and parts of another feed are selected according to the result of the filtering at the time the user triggers the Mash-up application. This advanced category assumes the resources to comply with certain standards to allow for their seamless connection.

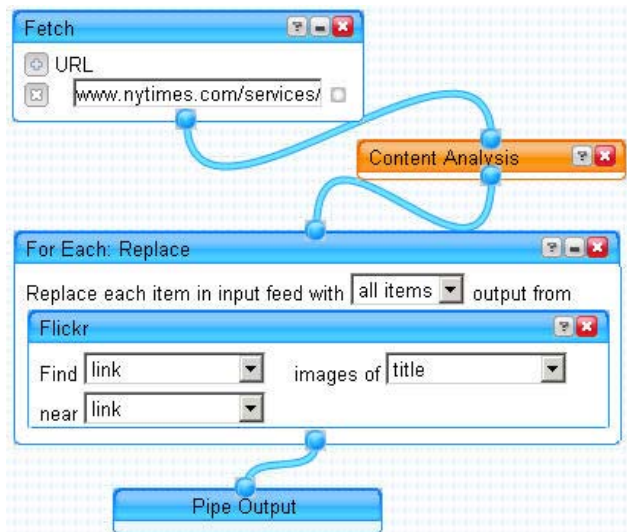


Figure 6: Yahoo Pipes facilitate mashing of content from arbitrary sources

Still, resources are only loaded and processed at one point of time and do not interact in a stateful fashion similar to Web Services that are called according to a previously defined choreography. Only very few examples exist that try to automate the mashing of application functionality rather than their simple aggregation. The firm **Kapow Technologies**¹⁴ has recently published its vision of so-called Enterprise Mashups. It focuses on empowering users of their Web Integration Platform to integrate Web-based resources on several different levels and thus efficiently realize rapid portal creation, content migration, synchronization and syndication, composite application integration and SOA enablement. According to the firms publications, logic-based Mash-ups can be setup on the basis of a merely visual, technology- and complexity hiding modeling interface. Enterprise applications which are encapsulated as, for example, Web Services or by REST-based interfaces can be mashed up to build new functionality.

3.3.2. Results

¹⁰ <http://www.google.com/>

¹¹ <http://www.springwidgets.com/>

¹² <http://pipes.yahoo.com/>

¹³ <http://flickr.com/>

¹⁴ <http://www.kapowtech.com/>

The analysis and categorization presented above shows that mainly three types of platforms have emerged which all aim at empowering individuals to mash up resources in an as quick and intuitive manner as possible. First, platforms such as Google facilitate the mere syndication of gadgets which are not interlinked at all. Second, environments such as Yahoo Pipes allow for processing and interconnecting resources that adhere to a certain standard (in this case RSS) and expose the result of this processing as a new resource. The mashing of resources on a application logic-level represents the third and last category investigated in the course of this work. Only very few examples exist which are capable to partly realize this sort of Mash-up (e.g., Kapow technologies).

Projects are under way and start-up firms are created that tackle this remaining challenge on the path to a global SOA that is driven by end-users and facilitated by lightweight programming models and intuitive Mash-up platforms and tools.

3. Conclusion

In this article, we elaborated on the Web Services stack as traditional enabler for coupling distant and potentially heterogeneous Web-based resources. As Web Services have not lived up to their promise to facilitate a global network of loosely interconnected services, new approaches are required to cope with this challenge. As a key component of the Web 2.0 era, novel technologies and design principles are now about to emerge which allow humans to use, customize, combine, interconnect and finally expose Web-based content or functionality as new resources on the Web. The results of reusing and composing existing resources are referred to as Mash-ups.

In this work, we provided an overview of already existing Mash-ups as well as different tools and platforms that empower users to build them in a highly efficient and intuitive fashion. Statistical data and case studies were leveraged to examine new ways of resource provision and consumption and also the relevance of upcoming intermediaries. Finally, we investigated remaining research challenges on the path to a truly global SOA: The development of platforms and tools for the automated mashing of application functionality rather than the mere syndication of resources will represent a key milestone which has to be achieved in future work.

Acknowledgments

The term “Brave New Web” has been brought up in the Web 2.0 context by Kenneth P. Morse, Managing Director, MIT Entrepreneurship Center

5. References

- [1]T. Malone: *The Future of E-Business*, Sloan Management Review, Vol. 43, No. 1, p. 104, 2001
- [2]M. Porter, *Strategy and the Internet*, Harvard Business Review, 2001
- [3] G. Alonso, F. Casati, H. Kuno, V. Machiraju: *Web Services Concepts, Architectures and Applications*, in: Springer, 2004
- [4] A. McAfee: Will Web Services Really Transform Collaboration, in: MIT Sloan Management Review, Vol.46, No.2, 2005
- [5]M. MacKenzie, K. Laskey, F. McCabe, P. F. Brown, R. Metz: OASIS - Reference Model for Service Oriented Architecture 1.0, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm, 2006
- [6]O'Reilly: What is Web 2.0, <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>, last seen: February 23, 2007
- [7]R. Högg, M. Meckel, K. Stanoevska-Slabeva, R. Martignoni: Overview of business models for Web 2.0 communities, Proceedings of GeNeMe 2006, p. 23-37, Dresden, 2006
- [8]Mashing the Web, *The Economist* 376 (8444), Special section, p. 4, 2005
- [9]R. D. Hof: Mix, Match, And Mutate, *Business Week Magazine*, 2005, available online at: http://www.businessweek.com/@@76IH*ocQ34AvyQMA/magazine/content/05_30/b3944108_mz063.htm, last seen: February 23, 2007
- [10]D. Hinchcliffe: i-Technology Viewpoint: Is Web 2.0 the Global SOA?, *SOA Web Services Journal*, <http://webservices.sys-con.com/read/164532.htm>, last seen: February 23, 2007
- [11]A. McAfee: Enterprise 2.0 vs. SOA, in: Harvard Business School Faculty Blog, http://blog.hbs.edu/faculty/amcafee/index.php/faculty_amcafee_v3/enterprise_20_vs_soa/, last seen February 28, 2007
- [12]P. Frost, S. Frost: *Component-based development for enterprise systems: applying the SELECT perspective*, in: Cambridge University Press, 1998
- [13]D. L. Parnas: On the criteria to be used in decomposing systems into modules, in: *Commun. ACM*, Vol.15, No.2, ACM Press, 1972
- [14]X. Chen, J. Funk, S. Madnick, R. Wang: *Corporate Household Data: Research Directions*, Proceedings of the Americas Conference on Information Systems, Boston, August 2001
- [15]P. Weill and J.W. Ross: *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results*, Harvard Business School Press, May 2004
- [16]C. Anderson: *The Long Tail: Why the Future of Business Is Selling Less of More*, Hyperion Books, 2006
- [17]R. T. Fielding: *Architectural styles and the design of network-based software architectures*, Ph.D. thesis, University of California, Irvine, 2000
- [18]D. C. North: *Institutions, Institutional change and economic performance*, Cambridge University Press, Cambridge, 1990

By examining and contrasting Web 2.0 and SOA, the authors envision a new Internet of Services that incorporates the best of both.

Christoph Schroth and Till Janner



Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services

Recently, the relationship between Web 2.0 and service-oriented architectures (SOAs) has received an enormous amount of coverage because of the notion of complexity-hiding and reuse, along with the concept of loosely coupling services (see <http://blog.hbs.edu/faculty/amcafee/index.php>). Some argue that Web 2.0 and SOAs have significantly different elements and thus can not be regarded as parallel philosophies (see http://edgeperspectives.typepad.com/edge_perspectives/2006/04/soa_versus_web_.html). Others, however, consider the two concepts as complementary and regard Web 2.0 as the global SOA. In this article, we investigate these two philosophies and their respective applications from both a technological and business perspective.

DEFINITION OF TERMS AND RESEARCH APPROACH

Tim O'Reilly coined the term Web 2.0 to describe a quickly growing set of Web-based applications. SOA is considered the philosophy of encapsulating application logic in services with a uniformly defined interface and making these publicly available via discovery mechanisms (C.M. MacKenzie et al., "Oasis—Reference Model for Service Oriented Architecture 1.0," 2006; <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf> and

A. McAfee, "Will Web Services Really Transform Collaboration," *MIT Sloan Management Review*, vol. 46, no. 2, 2005, pp. 78-84).

O'Reilly identifies seven major characteristics inherent to the Web 2.0 philosophy: first, the Web is considered a platform for building systems that are "tied together by a set of protocols, open standards, and agreements for cooperation" (T. O'Reilly, 2005, <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>). The exploitation of collective intelligence of Web users, ownership of mission-critical data, and the end of the software release cycle are quoted as central characteristics as well. The use of lightweight programming models that allow for loosely coupled applications, the use of diverse media and devices for the consumption of Internet-based applications, and the realization of rich user experiences represent further paradigms inherent to the concept of Web 2.0.

In 2006, Högg et al. conducted an in-depth investigation of 40 successful Web 2.0 applications (R. Högg et al., "Overview of Business Models for Web 2.0 Communities," *Proc. Gemeinschaften in Neuen Medien*, Technische Universität Dresden, 2006, pp. 23-37). They condensed their respective characteristics into the following statement, which works as an underlying definition for this article: "Web 2.0 is defined as the philosophy of mutually

maximizing collective intelligence and added value for each participant by formalized and dynamic information sharing and creation.”

Since the late 1990s, many definitions of SOA have been published (P. Frost and S. Frost, *Component-Based Development for Enterprise Systems: Applying the Select Perspective*, Cambridge Univ. Press, 1998 and G. Alonso et al., *Web Services Concepts, Architectures, and Applications*, Springer, 2004). The Oasis Reference Model for SOA (MacKenzie et al., 2006) defines SOA as

... a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with, and use capabilities to produce desired effects consistent with measurable preconditions and expectations.

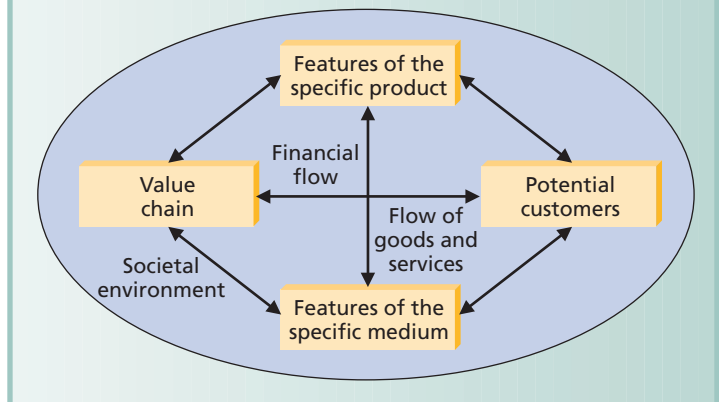
According to this model, a service provider might publish a well-defined interface on a registry that enables other stakeholders to retrieve and loosely couple the offered service with their own services.

The research approach applied for this comparative analysis is based on the examination of 40 real-world use cases of both Web 2.0 and SOA. In the case of Web 2.0, 40 popular applications (Högg et al., 2006) were selected, while case studies provided by Systeme Anwendungen und Produkte in der Datenverarbeitung (SAP), IBM, and Gartner were leveraged to investigate SOA solutions.

The Institute for Media and Communications Management (MCM) at the University of St. Gallen’s business model framework (see Figure 1) was leveraged to structure the analysis along seven major components (Högg et al., 2006):

1. *Features of the specific product* comprise the design of a product or service and its value for the customer.
2. *Features of the specific medium* define the technological foundation of a product.
3. The *customers* component refers to the target groups of an offered product or service and explains their respective business needs.
4. The *value chain* is devoted to reflecting all players that are involved in the production and delivery of a product and their respective interrelationships.
5. *Financial flow* identifies revenue models and explains the roles that different stakeholders play.
6. *Flow of goods and services* describes the stakeholders’ activities that are essential for the product’s or service’s creation.
7. Last, the *societal environment* reflects relevant outside influences on a business model (such as legal and social aspects and competitive situations).

Figure 1. The Institute for Media Communications Management’s (MCM’s) business model framework.



WEB 2.0 VS. SOA

In comparing each product or service, it’s worthwhile to consider Web 2.0’s and SOA’s pros and cons more thoroughly. In the following paragraphs, we analyze Web 2.0 and SOA based on the MCM business model framework’s seven major components.

Features of the specific product or service

According to Högg et al. (2006), we can classify Web 2.0 applications as follows:

- Communities that aim to unify their users by means of a common ideal such as social networking or knowledge sharing.
- Platforms or tools that help users create and share content with a broad audience (for example, Web logs and online directories). Mashup platforms let users retrieve content or functionality from arbitrary sources, mix it with other resources, and expose it for further reuse by other applications.
- Online collaboration tools support users in collaboratively performing certain tasks, such as maintaining time schedules or processing text online.

In comparison, the following are ways we can differentiate SOA use cases:

- First, SOAs allow for a cross-organizational integration of services. By adhering to common standards for the description of their service interfaces, corporations are enabled to setup loosely coupled electronic business transactions with other companies and thus automate business transactions in a quickly changeable fashion.
- Second, SOAs facilitate the intraorganizational integration of disparate services. On the basis of a central integration layer (often referred to as an Enterprise

Service Bus, or ESB), heterogeneous applications can be encapsulated and composed to a seamlessly integrated IT landscape (R.W. Schulte, "Predicts 2003: Enterprise Service Buses Emerge," Gartner Research Note, 2002; http://www.gartner.com/DisplayDocument?doc_cd=111977).

- Third, SOA-based application development significantly reduces development time thanks to the availability of reusable application building blocks.

The first major analogy between product design in the fields of Web 2.0 and SOA is the notion of reusing and composing existing resources. Both concepts let users reuse, remix, and enrich existing resources and components to new and potentially higher-level applications. The second commonness is the affinity to collaboration and coupling of remote resources or services. Both Web 2.0 and SOA applications enable the loose coupling of distant and possibly heterogeneous resources. A third apparent resemblance between Web 2.0 and SOA is the shared principle of agility and the support of permanent structural change.

Web 2.0 and SOA also have divergent elements. First of all, many Web 2.0 applications incorporate a social aspect, as they facilitate human interaction and also mainly deal with human-readable content (such as text and pictures). In contrast, conventional SOAs merely aim at interconnecting dispersed business functionality and facilitating seamless machine-machine collaboration. Second, Web 2.0 is clearly about presentation and user interface integration, whereas SOA deployments are more abstract and less visible to its users. Third, and last, the degree of ex-ante determination and involved governance (McAfee, 2005) is a key differentiator between Web 2.0 and SOA. Because of their frequent implementation in the corporate context, SOAs are subject to requirements that don't exist in the case of most Web 2.0 applications and thus underlie governance mechanisms (P. Weill and J.W. Ross, *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results*, Harvard Bus. School Press, 2004).

Features of the specific medium

As argued in Kolbitsch and Maurer, Web 2.0 doesn't stride along with a fundamental technological innovation, but is facilitated by a number of technologies (J. Kolbitsch and H. Maurer, "The Transformation of the Web: How Engineering Communities Shape the Information We Consume," *J. Universal Computer Science*, vol. 12, no. 2, 2006, pp. 187-213). First of all, the Representational State

Transfer (REST) is an architectural style that enables Web clients to interact with arbitrary Web resources in a uniform way (R.T. Fielding, "Architectural Styles and the Design of Network-Based Software Architectures," doctoral dissertation, Information and Computer Science Dept., Univ. of California, Irvine, 2000). The Really Simple Syndication (RSS) format supports the easy aggregation of content from arbitrary sources in the Web. AJAX represents a composite of several other technologies that together allow for rich user experiences, which is one of the key paradigms of Web 2.0 applications.

As opposed to lightweight Web 2.0 technologies (Hagel, 2006), SOAs rely on a set of more complex standards. As the Web Service Description Language (WSDL) and SOAP-based Web services are the most widely spread standards used to set up SOAs, these are in the focus of our technical analysis.

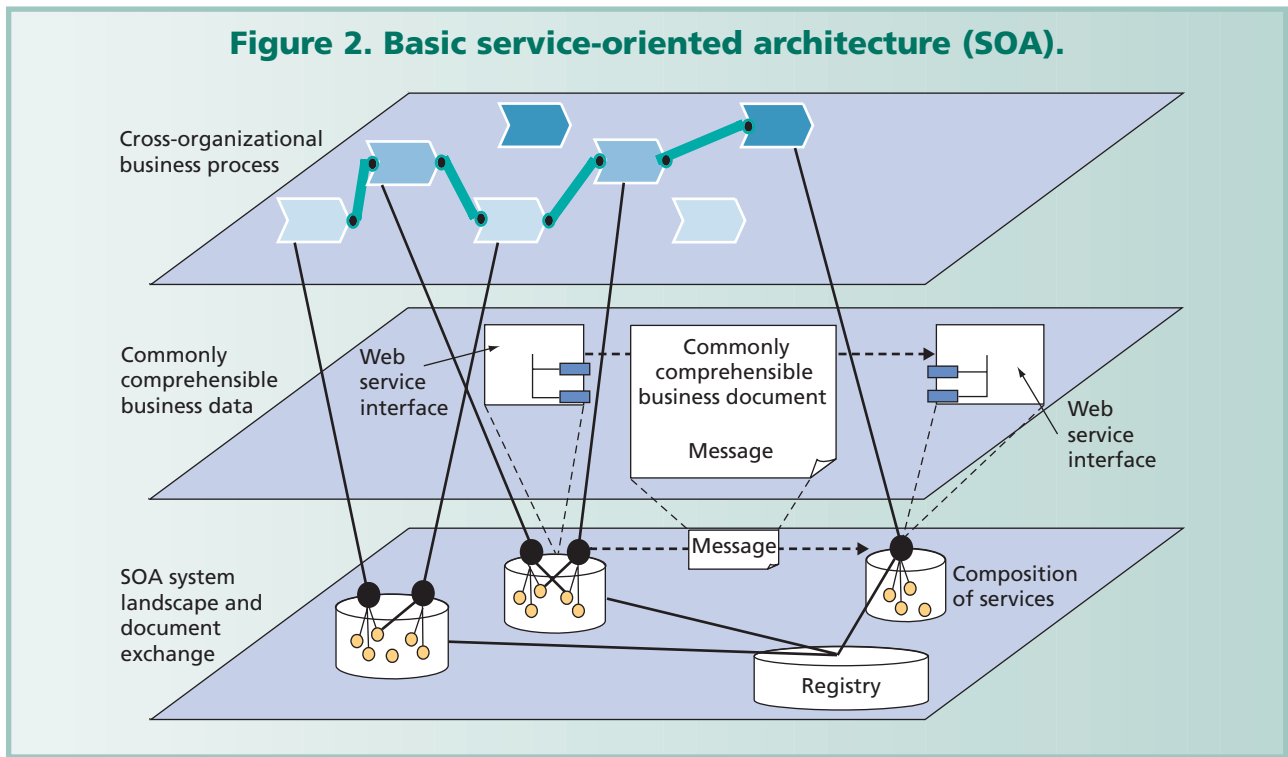
As Figure 2 shows, possibly heterogeneous applications (visualized as orange circles) are encapsulated as services and can be composed to new, aggregated functionality (black circles). WSDL is used for uniform service interface descriptions, while the Universal Description, Discovery, and Integration (UDDI) standard allows for open-service discovery. SOAP specifies a protocol for message exchange between services, while no widely accepted standard exists for data semantics (see the middle layer of Figure 2). The Business Process Execution Language is frequently used as a standard for orchestrating different services into one process choreography (see the upper layer in Figure 2).

As a first major technical similarity, standards applied in both Web 2.0 (such as RSS and REST) and SOA applications (such as WSDL) support the loose coupling of remote applications via uniform interfaces. Second, Web 2.0 and SOA technologies also share the notion of complexity hiding and reduced programming effort with the help of uniform descriptions of interfaces and data structures.

We can identify two major differences between the technological basis of Web 2.0 and SOA: First, we can make a distinction between the terms *syndication* and *coordination*. Many technologies used in the Web 2.0 context focus on static syndication of content and services, while SOAs often incorporate service coordination protocols (different services are invoked in a predefined sequence). The second difference refers to semantic interoperability. Web 2.0 applications directly involve human beings who are fault tolerant with respect to the information they're provided, whereas SOAs typically are limited to the mere interaction of machines, which aren't flexible to formal errors or semantic differences.

Because of their frequent implementation in the corporate context, SOAs are subject to requirements that don't exist in the case of most Web 2.0 applications.

Figure 2. Basic service-oriented architecture (SOA).



Potential customers

In the Web 2.0 context, basically every Web user can be regarded as a potential customer. Instead of heading for a small number of huge customers, Web 2.0 applications involve the bulk of private users or small businesses, also known as “the long tail” (C. Anderson, *The Long Tail: Why the Future of Business Is Selling Less of More*, Hyperion Books, 2006). As argued by O’Reilly, Web 2.0 is about leveraging customer self-service and thus is able to “reach out to the entire Web, to the long tail and not just to the head” (O’Reilly, 2005). In the examined SOA cases, medium-sized or larger corporations are the customers of choice. These aim to introduce SOA as a software design principle, as a possibility to streamline and harmonize internal IT landscapes or to set up cross-organizational business relationships. Because of their substantially different application domains, Web 2.0 and SOA applications serve different customer needs and requirements.

Value chain

In the Web 2.0 context, traditional value chains are broken up to a large extent and substituted by loose networks of providers and consumers (Högg et al., 2006). Every user may publish his or her own content or functionality on the Web and thus become a platform operator, consuming resources or reusing them to compose new applications and make them publicly available. In contrast, the high technical complexity inherent to SOAs requires the existence of one or several expert players within the value

chain who 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), Web 2.0 value chains are mostly loosely coupled (many-to-many) networks of self-managed users who can offer and consume resources via the Web.

Financial flow

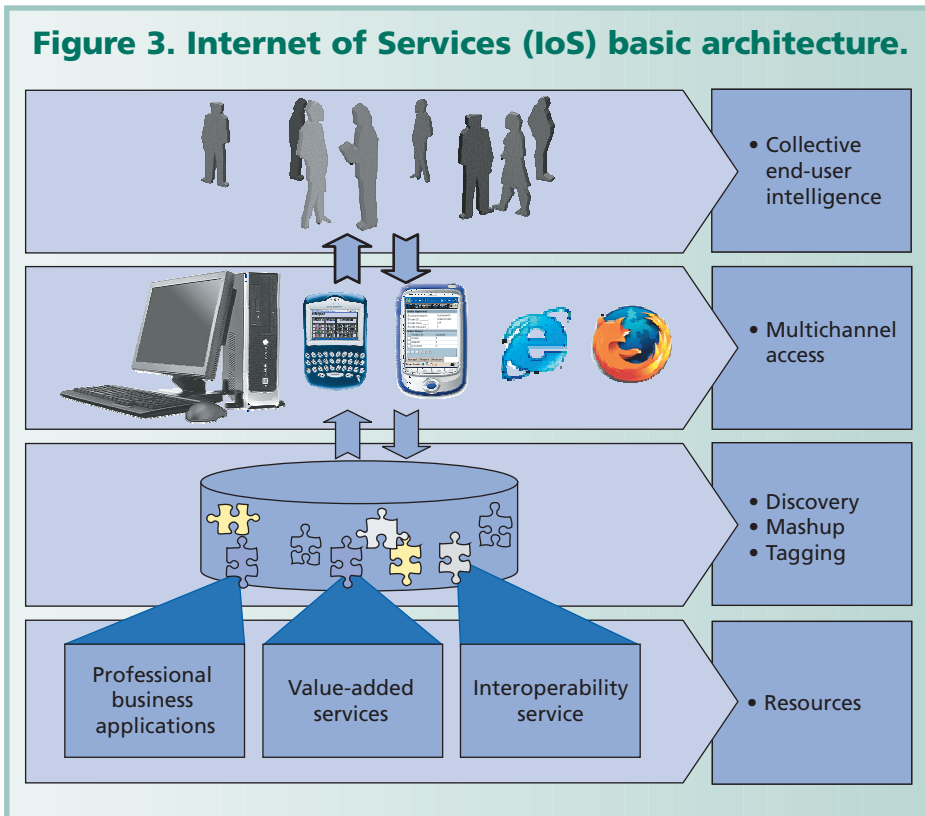
Revenue models of Web 2.0 applications strongly differ from traditional models in the software industry. First, software applications aren’t packaged and sold as over-the-counter products anymore, but provided as services (or the “end of the software release cycle,” Högg et al., 2006). For a significant share of Web 2.0 applications, the number of users is part of their central value, so providers generally don’t introduce service fees. They take this approach to avoid limiting the amount of users, but then this approach leads to challenges with monetizing their products.

The integration of third-party advertisements represents one possible solution to this revenue model dilemma. SOA revenue models in most cases follow a more traditional approach, as license fees are charged for using the respective solutions. This could change in the next few years as enterprise services become easily retrievable and usable via the Web.

Flow of goods and services

The decisive difference between typical product creation and provision in the fields of both Web 2.0 and SOA results

Figure 3. Internet of Services (IoS) basic architecture.



CONVERGING WEB 2.0 AND SOA CONCEPTS

In general, the philosophies of Web 2.0 and SOA serve different user needs and thus expose differences with respect to the design and used technologies of real-world applications. However, recently numerous novel use cases demonstrate the great potential of combining the technologies and principles of Web 2.0 and SOA.

One major example of the convergence of the two philosophies is the emergence of a global SOA that we refer to as Internet of Services (IoS). Up to now, normal Internet users with little IT sophistication haven't been able to easily retrieve and use certain services. This is because these services mostly reside within company boundaries and are only accessed for professional use in a corporate

context. However, the provision of easily accessible services for end users might drive a novel generation of Internet use and allow for the ad-hoc setup and configuration of IT-supported business models. Without an intuitive "face" toward human users, Web-based services will remain fairly unusable by the common Web surfer, because currently these interfaces (such as those described in WSDL) are designed to be processed by machines.

Figure 3 visualizes a basic IoS architecture that's based on a combination of principles and technologies from both Web 2.0 and SOA. Resources that are accessible via the Web are registered in platforms and can thus be discovered, tagged (to collect user evaluation and to allow for folksonomies), and also mashed up (composed and interlinked with the goal of designing new resources) according to the users' requirements (O'Reilly, 2005).

Arbitrary stakeholders can provide and host services, thereby leading to a global market with decentrally organized platforms that act as brokers. The actual users can access these platforms for discovery purposes via intuitive interfaces that facilitate tagging and mashing activities without requiring any coding effort. Then we could leverage all of the different channels (such as PCs or mobile devices) to enter and use the platforms. The establishment of an open architecture that comprises possibilities for human beings to use and interact with Web-based resources has the potential to drive the development of a global mesh of services.

Also, numerous Web 2.0 application providers don't rely on traditional marketing and sales activities but aim at *viral marketing*—that is, recommendations autonomously propagating from one user to another. In the SOA context, large solution providers such as SAP and IBM mostly treat SOA products as software artifacts that are packaged and then sold to customers.

Societal environment

As we previously noted, Web 2.0 use cases considerably benefit from the lack of formal guidelines and governance mechanisms with which SOAs typically must cope. However, cases exist where providers of Web-based platforms have been enforced to review the content they publish, thereby limiting the growth and dynamics of the offered solutions (Högg et al., 2006). As opposed to most Web 2.0 applications, SOAs are subject to clearly defined regulatory frameworks (such as the Sarbanes-Oxley Act), because they mostly exist in the corporate context. The design, provision, maintenance, and coupling of services must be compliant with legal frameworks and thus they do not allow for flexibility as observed within the Web 2.0 context.

A combination of principles from both Web 2.0 (user self-service and collective end-user intelligence) and SOA (a composition of reusable building blocks) can facilitate the wide dissemination of many resources. Examples include professional business applications, value-added services (including location-based services), and interoperability services (for example, applications that can be leveraged by trading partners to initiate business-to-business transactions).

Enterprise mashups represent one specific use case of this type of architecture that could easily be situated at the interstice of Web 2.0 and SOA (A. Mulholland et al., *Mashup Corporations: The End of Business As Usual*, Evolved Technologist Press, 2006). Technically unsophisticated business experts would be empowered to model and deploy business models in an extremely quick and efficient fashion. The interconnection of presentation-layer-focused Web applications to internal SOA implementations could be of significant value for enterprises, as this could extend their services' reach to the Web for further use and composition by their business partners and customers.

The firm Kapow Technologies (see <http://www.kapowtech.com>) recently announced the release of a technical solution for these kinds of enterprise mashups. It focuses on empowering users of their Web integration platform to integrate resources available via the Web on several different levels.

First, this helps to seamlessly integrate and expose user-interface content and functionality from arbitrary sources as a new service on the Web. Second, it helps establish application mashups by composing and coupling applications that are accessible via REST or WSDL interfaces into new services.

The key goal of such platforms for creating enterprise mashups is to provide businesses with speed, flexibility, and agility in creating and changing cross-enterprise applications on the basis of a merely visual, complexity-hiding modeling interface. The envisioned result of this approach that combines Web 2.0 and SOA technologies (for example, AJAX, SOAP-based Web services, RSS, and REST) is achieving fast and business requirements-driven content and application integration.

The United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) provides another excellent example of the combination of Web 2.0 and SOA (T. Janner et al., "From EDI to UN/CEFACT: An Evolutionary Path Towards a Next Generation e-Business Framework," *Proc. 5th Int'l Conf. e-Business*, 2006, King Mongkut's Univ. of Technology; <http://www.alexandria.unisg.ch/Publikationen/30346>). It proposes a novel approach for the standardization of business processes. Instead of prescribing yet another fixed standard for the establishment of cross-organizational SOAs, the UN/CEFACT envisions establishing a publicly accessible


repository featuring a basic set of modeling building blocks that can be used, extended, and tagged by the users according to their actual business requirements.

In this work, we thoroughly contrasted the two philosophies of Web 2.0 and SOA from a technical and economic perspective. We identified numerous similarities—but also clear differences—in each of the seven criteria for comparison.

Recent applications show the paramount importance of unifying the two philosophies to drive the next wave of value creation within and across enterprises. Web 2.0 incorporates a social philosophy that we consider complementary to the technology-focused SOA philosophy, as it provides techniques and design principles that strongly facilitate the active consumption of Web-based resources. By integrating the long tail of Web users (Anderson, 2006) into application design via all relevant media channels and on the basis of easily usable platforms that allow for discovering, mashing, and tagging diverse resources, we can realize a comprehensive IoS. ■

Christoph Schroth is a Research Associate and PhD candidate in information management at the University of St. Gallen and Systeme Anwendungen und Produkte in der Datenverarbeitung (SAP) Research in Switzerland. Contact him at christoph.schroth@sap.com.

Till Janner is a Researcher Associate at SAP and is working toward his PhD in information management at the University of St. Gallen. Contact him at till.janner@sap.com.



**The IEEE
Computer
Society**

**publishes over 150 conference
publications a year.**

**For a preview of the latest
papers in your field, visit**

www.computer.org/publications/

Enterprise Mashup

Putting a face on the next generation global SOA

Tutorial Handout

SYSTEMATIC THOUGHT LEADERSHIP FOR INNOVATIVE BUSINESS



Miguel A. Cañas², Juan J. Hierro², Volker Hoyer¹, Till Janner¹,
David Lizcano³, Marcos Reyes², Christoph Schroth¹, Javier Soriano³

¹SAP Research (Switzerland)

²Telefónica I+D (Spain)

³Universidad Politécnica de Madrid, (Spain)

The 8th International Conference on Web Information Systems Engineering
3-7th December, 2007, Nancy, France



SAP RESEARCH

THE BEST-RUN BUSINESSES RUN SAP™



Tutorial Handout – Content Overview



- Time schedule
- Tutorial short description
- Poster of the EzWeb project
- Poster of the FAST project
- Related papers from the tutorial authors

SAP RESEARCH

© SAP 2007 / Page 2

Time Schedule (14:00-17:00 CET)



Time	Topic
14:00-14:10	Opening, welcome, and introduction
14:10-14:30	General motivation for Enterprise Mashup, related work, and context of the tutorial
14:30-14:50	Enterprise Mashup foundations – terms and concepts
14:50-15:20	Enterprise Mashup foundations – key technologies
15:20-15:30	Overview (demo), and comparison of existing platforms (part 1)
15:30-16:00	Coffee Break (scheduled for all workshops/tutorials)
16:00-16:15	overview (demo), and comparison of existing platforms (part 2)
16:15-16:45	Live-demos of Telefónica I+D prototypes
16:45-17:00	Q&A, discussion, tutorial closing

SAP RESEARCH

© SAP 2007 / Page 3

“Enterprise Mashup: Putting a face on next generation global SOA”

Short description

Mashup is considered as the central enabler concept/technology of the Web 2.0 era. Based on definitions of [Hof, R.D., (2007)] and [The Economist (2005)] we take the following summary as foundation and definition of a Mashup: *A 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.*

Mashups in an Enterprise context aim at enabling the users to dynamically compose and interconnect their own operational environments and processes in a very simple and flexible fashion. An **Enterprise Mashup platform** will facilitate flexible, useful and effective user interaction and management with all kind of resources. Especially the integration of Web Service technology is important to mention in an Enterprise context. The creation of so-called Gadgets on top of services enables end-users to understand the not human readable WSDL interfaces of current Service-oriented Architectures.

The central topic of the tutorial, **Enterprise Mashups**, will be presented to the audience with two major parts, theory and practice. The theory part includes the motivation, the necessary definition of important terms and concepts and also an overview of the state of the art with regard to current approaches of mashup platforms. A comprehensive argumentation and presentation of the requirements towards the realization of an Enterprise Mashup platform completes this part.

During the practical part of the tutorial, **demonstrations of major existing platforms** that allow users to create Mashups will be given. Finally the presenters will give a **demonstration with a first prototype of an Enterprise Mashup platform**. A practical scenario will be presented that demonstrates the value of an Enterprise Mashup platform in a large enterprise.

The tutorial is targeted to an audience of (but not limited to) people with a research or industrial background and interest in the following topics:

- Enterprise Mashup
- User-/Service Interaction
- End-user driven service/resource Mashup
- Piping and wiring of Web Resources
- Enterprise 2.0
- Next-generation SOA

- Internet of Services

References

Hof, R.D., (2007): Mix, Match, And Mutate, Business Week Magazine, 2005, available online at: http://www.businessweek.com/magazine/content/05_30/b3944108_mz063.htm, last seen: May 30, 2007

The Economist (2005): Mashing the Web, The Economist 376 (8444), Special section, p. 4

O'Reilly, T. (2007): What is Web 2.0, <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>, last seen: May 30, 2007

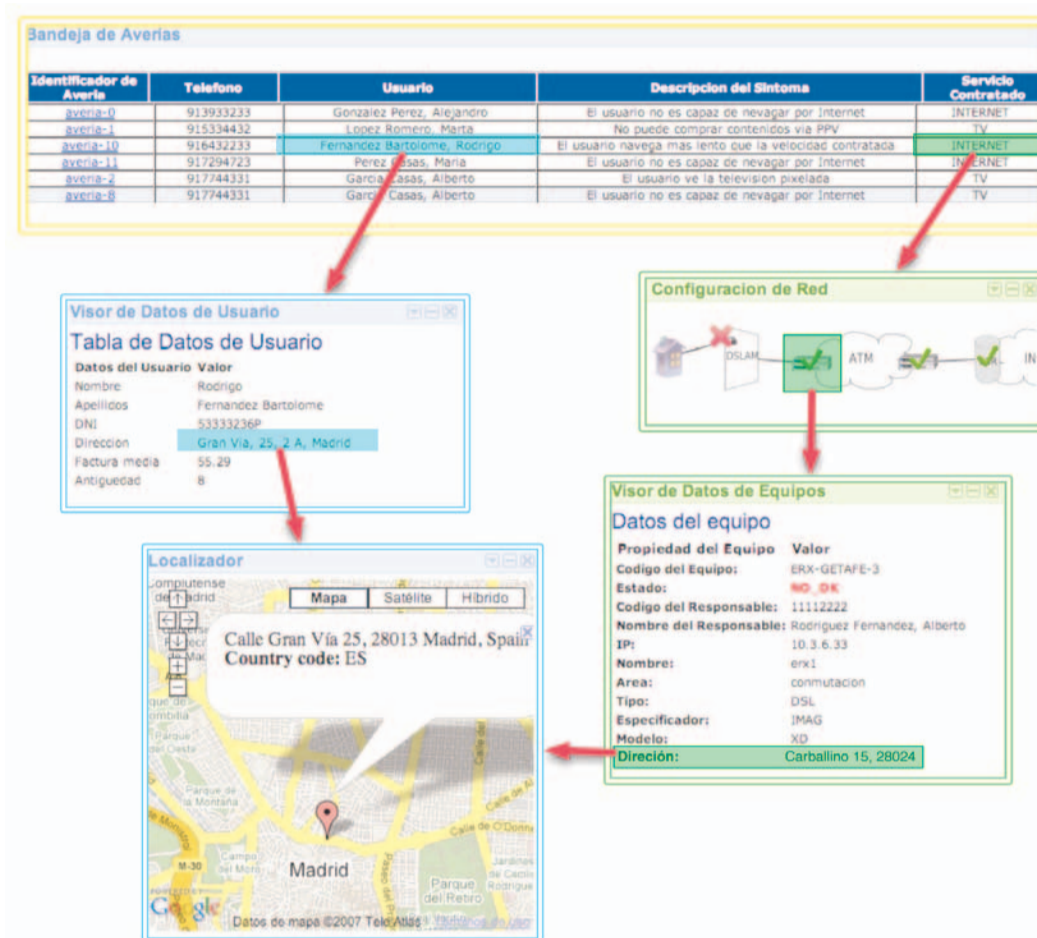
EzWeb - Putting a face on SOA

Target reference architecture and implementation of an **open standard platform** which **manages the frontend access in the next-generation web** and **supports the economics of a resource marketplace**.

Advanced Operation Environment

Users are able to build their own front-end applications by combining available resources when they need it and the way they need it

Web 2.0 principles and technologies are used in order to offer flexibility and universal access

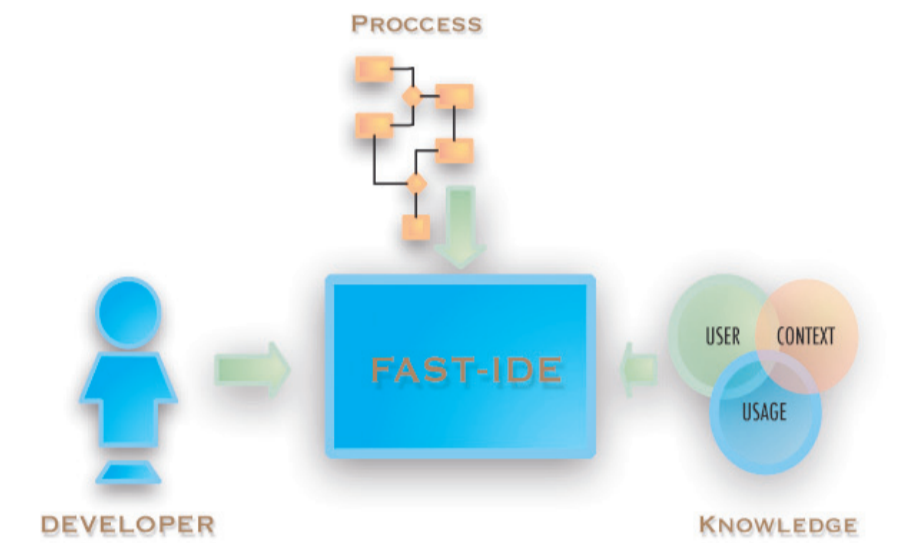


Development Environment

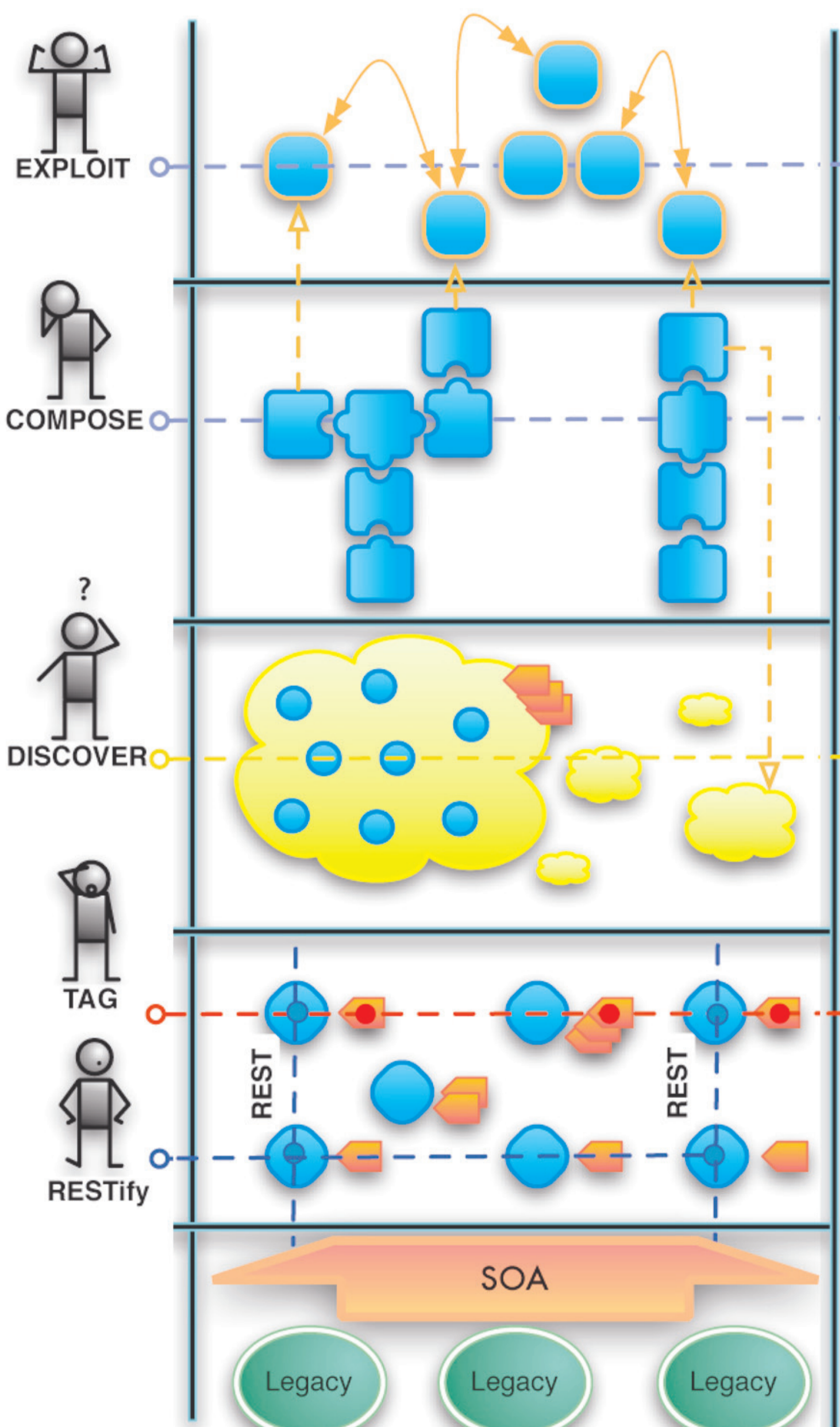
Create complex gadgets and resources combining simpler ones

Piping and ScreenFlow techniques are used to dynamically orchestrate processes and services

Semantic and context information is used to guide the process



Global SOA Architecture

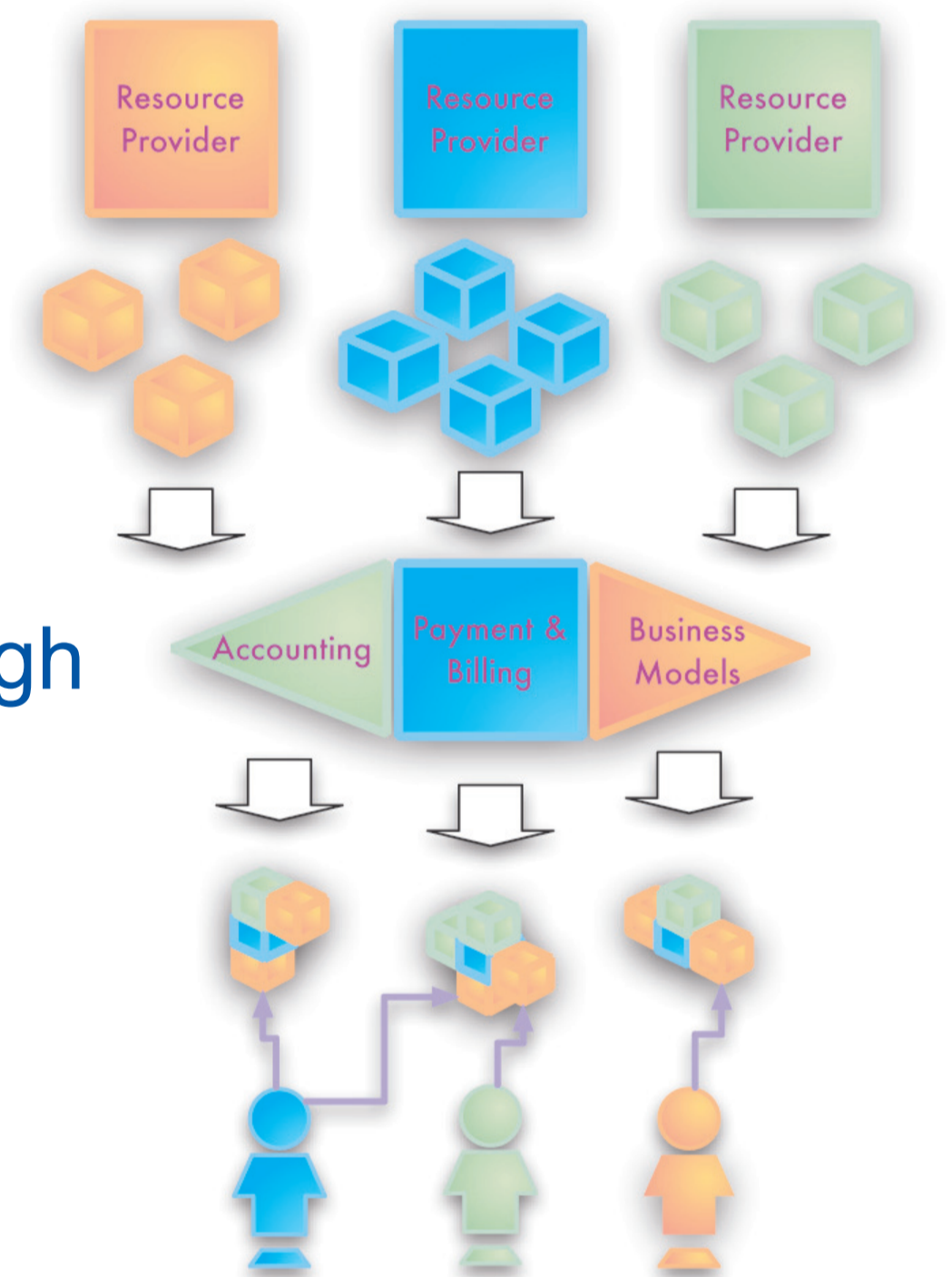


Resource Catalogue and Marketplace

Users can discover and select the resources that best fit their current needs

Resource providers can add new resources that cover all aspects related to the accounting, auditing payment and billing of the services.

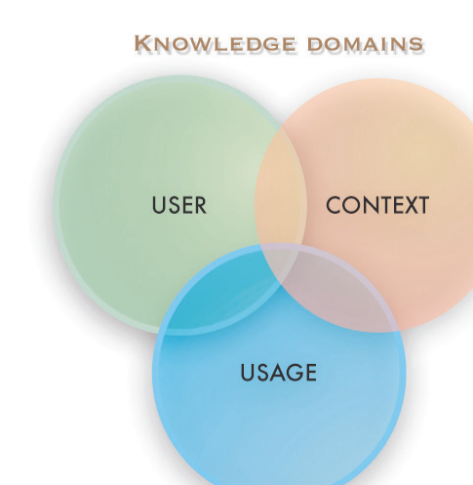
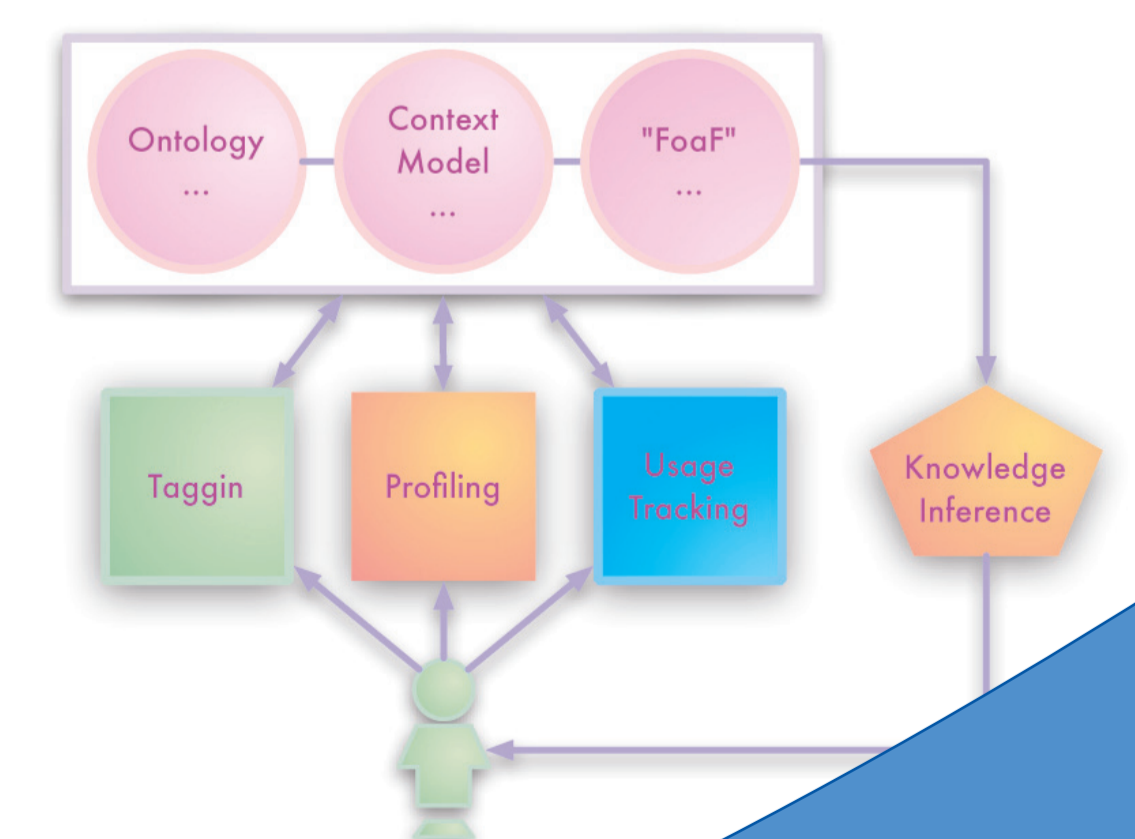
Ultimately the catalogue is a space where resources can evolve through combination with other resources



Knowledge capture and exploitation

Lightweight knowledge representation techniques are used: tagging, profiling...

Knowledge is leveraged through sharing and advanced inference mechanisms



Vision
 Creation of a new visual programming environment following a user-centric approach that 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.

Objectives

New visual programming environment for the development of complex front-end gadgets

- Involves the modeling and execution of relatively complex business processes
- Relies on back-end Semantic Web Services
- Agile software development from a top-down perspective

Exploitation of the full power of **semantics for integrating back-end with front-end** (user-centric)

- Developers will start from the front-end gadgets that the user will see and interact with
- Connection to back-end Web Services going through business execution flows if necessary

Plugin Tool for Enterprise Mashup Platforms

- Conceived as a tool that can be compatible with existing and future Enterprise Mashup Platforms
- Tool that enables to develop Mashup-able gadgets which rely on screen-flow Resources and Semantic Web Services stored in a (semantically enriched) catalogue

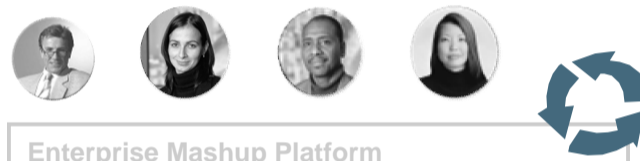
Consortium

	Telefónica Investigación y Desarrollo (Coordinator) Telecommunication Provider	
	SAP AG World's leading Enterprise Application Provider	
	University of Ireland, Galway Leading research institution in Semantic Web	
	University of Kassel Research experience in scenario-based methodologies	
	Cynetlix Corporation Semantic Web technology and collaboration platform	
	Universidad Politécnica de Madrid Semantic Web and Web 2.0-based technologies	

Architecture and Components

Conceptual Model

- Conceptualization of complex gadgets having an associated workflow and possibly human interaction
- Specification of a new ontology to conceptually model to specialise their development to solve concrete user problems
- Definition of screen flows and gadget templates in a visual way
- Definition of mechanisms generating automatically new gadgets from previously developed ones or parts of them

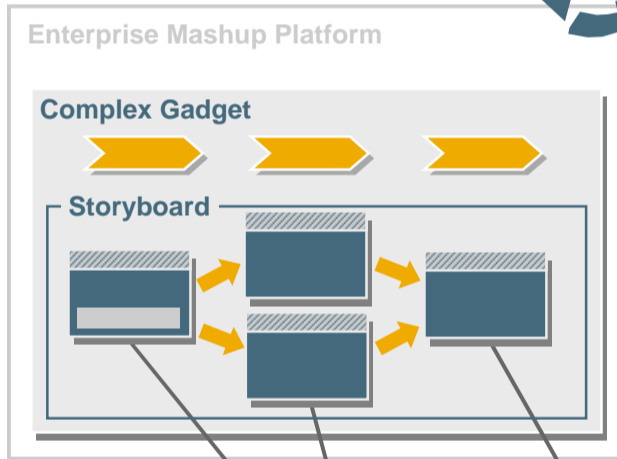
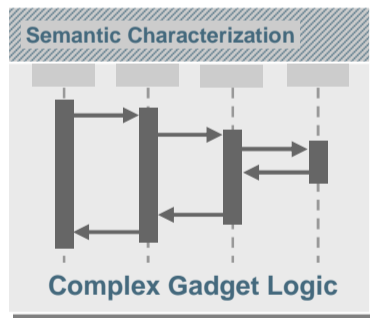


User-Centric Collaboration

- Agile software development from a top-down perspective
- Highly visual integrated development environment
- Continuous improvement

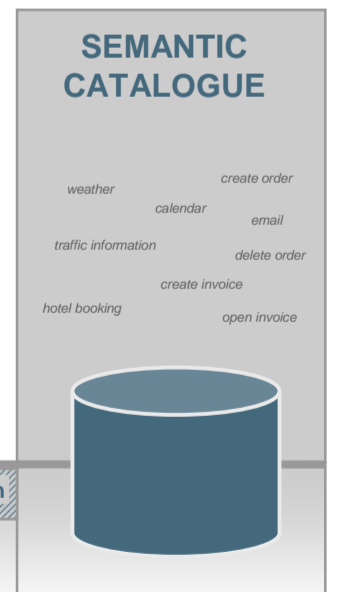
Visual Design of Complex front-end Gadgets

- Tools for the development of complex gadgets (ISO 9000 compatibility)
- Build the different screens composing the gadgets
- Design and implement the flow between these screens
- Define directives and specifications that assure the development and compatibility of the gadget developed in any „gadget enabled“ platform



Visual Composition of screen-flow Resources and Interoperability with back-end Web Services

- Bridge the gap between the ideas managed by user at a user-oriented level on the screen-flow and the underlying resource available in the back-end as Web Services and Resources
- Allow the automatic discovery of services and their friendly use in the visual tool



Benefits

Software industrialization

- Putting a visual face to Service-Oriented Architectures (SOA)
- Lightweight service consumption

Covering the profitable "Long Tail" in Enterprise Application

- User empowerment from service consumer to producer („prosumer“)
- Development of individual gadgets according to user requirements

Faster to market

Semantic Catalogue of screen-flow Resources and back-end Web Services

- Definition of mechanisms to connect gadgets to Semantic Web Services
- Mediation between gadgets ontology and other Semantic Web Services ontologies
- Discovery of Semantic Web Services based on user requirements and capabilities
- Automatically endow gadgets with desired data and functionality

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

Juan J. Hierro, Till Janner, David Lizcano, Marcos Reyes, Christoph Schroth, Javier Soriano

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 that makes up that *Web of Services* and shift the roles and characteristics of resource consumers, providers and intermediaries considerably. A novel platform architecture is presented that builds on all the key technical enablers and has the potential to drive the development of a global, user-centric SOA. Existing prototypes of this platform will be shown as a proof-of-concept.

Index Terms—SOA, Web Services, Enterprise Mash-up, Web 2.0, Global SOA, Internet of Services, user-centric SOA, Web of Services, Program by Composition, lightweight program models.

I. 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

Manuscript sent October 16, 2007. This work is supported in part by the European Commission under The Seventh Framework Program (Project FAST, INFSO-ICT-216048) and by the European Social Fund and UPM under their Research Personnel Training program.

Javier Soriano and David Lizcano are with Universidad Politécnica de Madrid, Campus Montegancedo s/n, 28660 Boadilla del Monte, Madrid, Spain (e-mail: jsoriano@fi.upm.es; dlizcano@fi.upm.es). Javier Soriano is an IEEE Member, and the contact author.

Juan J. Hierro and Marcos Reyes are with Telefónica I+D, Emilio Vargas 6, 28043, Madrid, Spain (e-mail: jhierro@tid.es; mru@tid.es).

Christoph Schroth and Till Janner are with SAP Research CEC St. Gallen, Switzerland (email: christoph.schroth@sap.com; till.janner@sap.com).

boundaries at present [18], and the global provision and consumption of services over the Internet is still at an early stage [7]. 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 choreographies, 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.

II. WEB SERVICES-BASED SOAs AND THEIR SHORTCOMINGS ON THE ROAD TOWARDS A REAL WEB OF SERVICES

Many definitions of SOA have been published since the late 1990s [1]. The widely accepted normative OASIS Reference Model for SOA defines SOA as "...a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations" [10]. According to this model, the major components of a basic SOA and their possible interactions are: a service provider publishes its service interface via a service registry where a service requester can find it and subsequently may bind to the service provider. The central concept of the SOA Reference Model is the existence of services that provide access to capabilities by well-defined interfaces to be exercised following a service contract with constraints and policies. This enables a loose coupling of services (thereby minimizing mutual dependencies) and complies with some of the probably best-known software engineering principles: information-hiding and modularization. Services are provided by entities, the service provider, and are to be used by others, the service consumers.

Services may be composed on the basis of other, existing services, thereby adhering to the principle of reuse. They are autonomous (solely control the logic they encapsulate), uniformly described and publicly retrievable via certain discovery mechanisms. Nowadays, IT and Internet evolution has stated that these services should be more easily deployed by

providers, as well as accessed and managed by end users than traditional packaged software [20]. Therefore, **services must be accessible for all users** (not only enterprise stakeholders), because these services should replace a previous wide range of enterprise software [7],[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, represents an architecture focused fundamentally on a B2B context 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.

III. 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 [9], 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.

A. *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 implementations. 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.

B. Key technological principles

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.

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.

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

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 [20]. 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 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 [20]. 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.

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 [8].

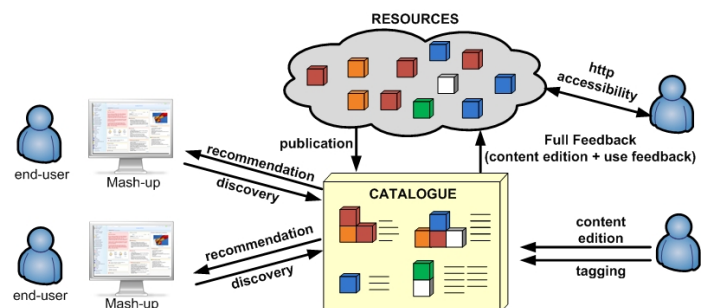


Fig. 1. Cataloguing Resources

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.

IV. 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.

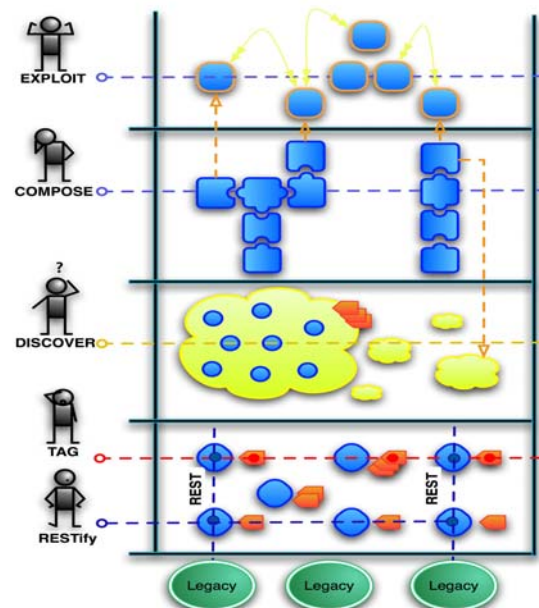


Fig. 2. User-centered SOA solution stack

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. This approach proposes the conceptualization of items like, for example:

- Interaction models: The user interface constitutes a bidirectional communication channel between an end user and a system or machine. The interaction model between these parties needs to be conceptualized to assure proper communication between the user and the machine. This is done by conceptualizing the different interaction models (direct action, event-based, publish and subscribe, etc.) present today and even proposing new ones stemming from the use of semantics in the front-end.
- Internal logic flow: A user interface is just appearance without connecting it with data, resources and functionality. One important aim would be to conceptualize the gadget's internal logic flow, as well as its interaction with semantic Web services. For example, once conceptualized, a gadget action could lead to the execution of a more or less complex logic flow, including invocations to existing semantic Web services to complete a task. 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.

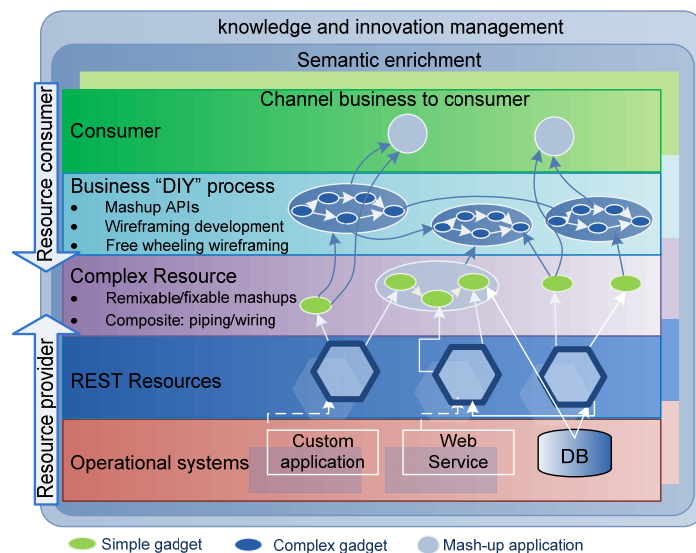


Fig. 3. Logical layers in the proposed approach

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 **automatically**

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.

V. 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.



Fig. 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

VI. 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).

VII. CONCLUSION AND FUTURE TRENDS

In this paper, we elaborated on the traditional Web Services-based approach to SOA as a traditional enabler for integrating distant and potentially heterogeneous Web-based resources. As Web Services have neither lived up to their promise of facilitating a global network of loosely interconnected services nor provided an appropriate front-end for people, new approaches are required to rise to this challenge of enabling a Web of Services. As a key component of the Web 2.0 era, novel technologies and design principles are now about to emerge that will allow human users to use, customize, combine, interconnect and finally display Web-based content or functionality as new resources on the Web. In this paper we have elaborated on the synergies between the Web 2.0 and the SOA worlds that can be exploited to rise to this important challenge and have proposed a generic model of a global user-centric SOA and a novel architecture for enterprise mash-up platforms.

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 [12] 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].

REFERENCES

- [1] Alonso, G., Casati, F., Kuno, H. & Machiraju, V.(2004). *Web Services Concepts, Architectures and Applications*. Springer, 2004
- [2] Anderson, C.(2006). *The Long Tail, Why the Future of Business is Selling Less of More*. Hyperion. July 2006.
- [3] Coveyduc, J., Huang, C., Ostdiek, L. & Reif, J.(2006). *IBM Innovation Factory: An integrated solution for accelerating innovation*. IBM, October 2006. Available at <http://ibm.com/websphere/developer/zones/hipods>.
- [4] Davenport, T. H.(2005). *Thinking for a Living: How to Get Better Performance and Results from Knowledge Workers*. Harvard Business School Press, Boston, MA, USA, 2005.
- [5] Evans, P. & Wurster, T. S.(2000). *Blown to Bits: How The New Economics of Information Transforms Strategy*. Harvard Business School Press, Boston, MA, USA, 2000
- [6] Fielding, R. T.(2000). *Architectural styles and the design of network-based software architectures*, Ph.D. thesis, University of California, Irvine, 2000
- [7] Gartner Inc. (2006). *Hype Cycle for Software as a Service*, Gartner Research, 10 August 2006.
- [8] Giles, J.(2005). *Internet encyclopedias go head to head*. *Nature*, Vol.438, No.7070 (pp.900-990), December 2005.
- [9] Högg, R., Meckel, M., Stanoevska-Slabeva, K. & Martignoni, R.(2006). *Overview of business models for Web 2.0 communities*, *Proceedings of GeNeMe 2006*, p. 23-37, Dresden, 2006.
- [10] MacKenzie, M. (2006) *OASIS - Reference Model for Service Oriented Architecture 1.0*, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm
- [11] McAfee, A.(2005). *Will Web Services Really Transform Collaboration*, *MIT Sloan Management Review*, Vol.46, No.2, 2005
- [12] McAfee, A.(2006). *Enterprise 2.0: The Dawn of Emergent Collaboration*. *MIT Sloan Management Review*, Vol.47, No.3 (pp. 21-28).Spring 2006.
- [13] North, D.C.(1990). *Institutions, Institutional change and economic performance*, Cambridge University Press, Cambridge, 1990
- [14] O'Reilly, T.(2005). *What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software*. www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html
- [15] O'Reilly, T. & Musser, J.(2006). *Web 2.0 Principles and Best Practices*. O'Reilly radar, November 2006.
- [16] Roman, D. (2005). *Web Service Modeling Ontology*, *Applied Ontology*, Vol.1, No.1 (pp. 77 - 106), 2005.
- [17] Schroth, C. & Christ, O. (2007). *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)* (pp. 8): IEEE Computer Society. Retrieved 2007-07-11, from <http://www.alexandria.unisg.ch/publications/37038>.
- [18] Schroth, C. & Janner, T. (2007). *Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services*. *IEEE IT Professional* Vol.9, No.3(pp.36-41) , June 2007.
- [19] Smith, R.(2006). *Enterprise Mashups: An Industry Case Study*. Keynote at the New York PHP Conference & Expo, Manhattan, New York, USA, 14-16 June 2006
- [20] Winewright, P. (2005) *Why Microsoft can't best Google*, *Software as a Service ZDNet editorial*, August 2005. Retrieved September 18 2007, from <http://blogs.zdnet.com/SAAS/?p=13>

Fostering Innovation in a Mashup-oriented Enterprise 2.0 Collaboration Environment

Javier Soriano¹, David Lizcano¹, Miguel A. Cañas², Marcos Reyes² and Juan J. Hierro²

¹ Universidad Politécnica de Madrid, Campus Montegancedo s/n
28660 Boadilla del Monte, Madrid, Spain

² Telefónica I+D, Emilio Vargas 6, 28043 Madrid, Spain
Email: {jsoriano; dlizcano}@fi.upm.es; {mac, mru, jhierro}@tid.es

Abstract: A number of Enterprise 2.0 Collaboration platforms are beginning to proliferate. By leveraging desirable Web 2.0 attributes, these platforms provide enterprises with an ecosystem of employees, partners, suppliers, and customers who collaborate to develop capabilities by collectively generating, sharing and refining business knowledge. This paper advocates the idea that enterprise collaboration must evolve towards a new paradigm in which knowledge workers 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. 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. This architecture will help to evolve the above ecosystem into one in which all the stakeholders will also be able to collaboratively develop capabilities and innovate operating procedures by remixing and integrating already available services through the core ideas of enterprise mashups and collaborative resource repositories.

Keywords: Enterprise Collaboration, Enterprise 2.0, Web 2.0, Knowledge Work, 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, Python, 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 were 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. Google IG 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.

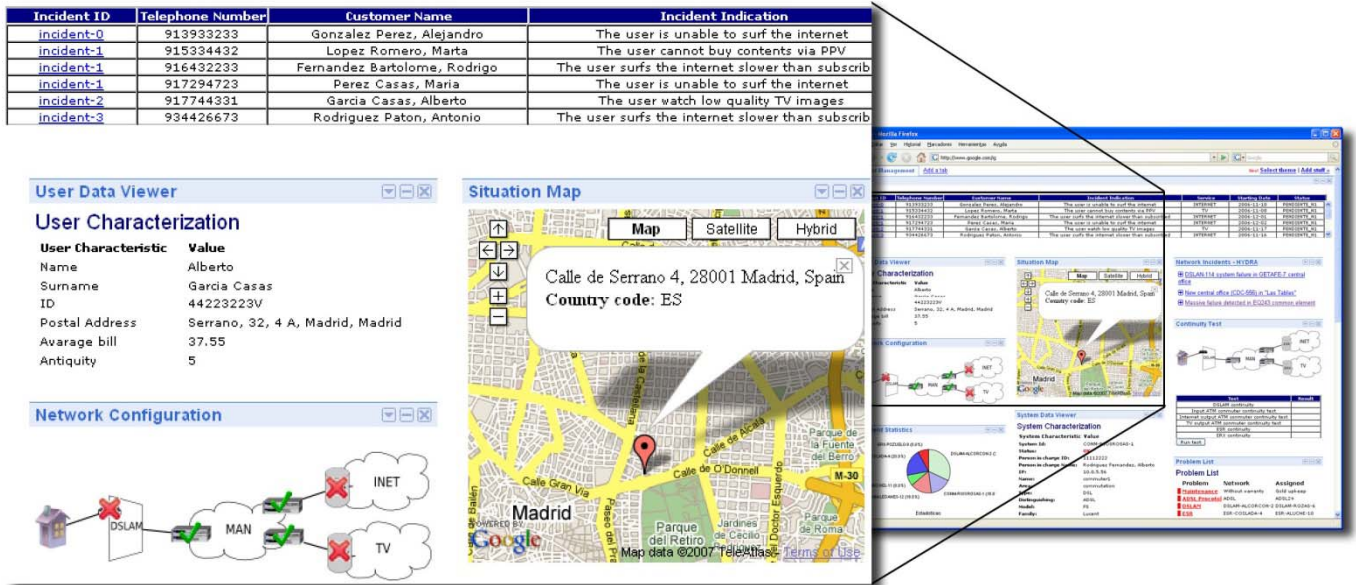


Fig. 1. A real world EzWeb enterprise mashup currently deployed at Telefónica OSS as a fully operational environment.

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 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. 2 and are further explained in the following subsections.

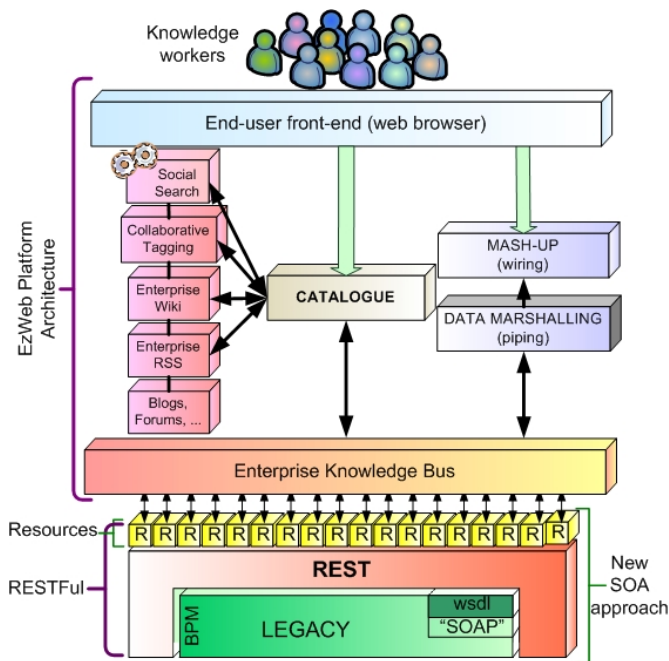


Fig. 2. EzWeb architecture.

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.

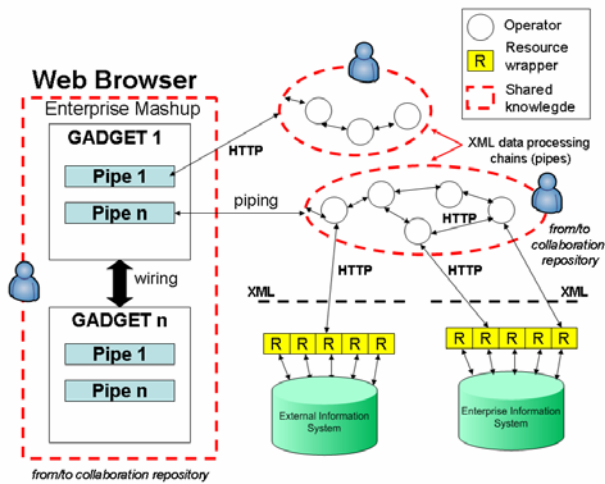


Fig. 3. Resource composition in enterprise mashups.

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.

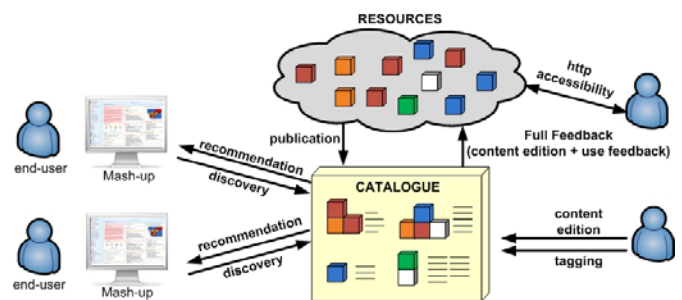


Fig. 4. Cataloguing resources.

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.

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.

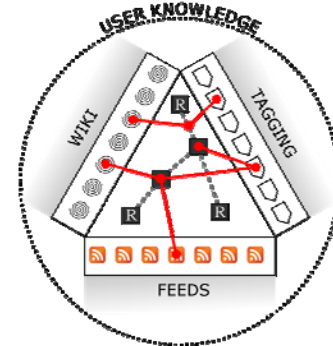


Fig. 5. Collective knowledge emergence through a catalogue.

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.

Acknowledgement

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.

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/yaho-o-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.

Notes

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.

Brave New Web: Emerging Design Principles and Technologies as Enablers of a Global SOA

Christoph Schroth^{1,2}, Oliver Christ²

¹ *University of St. Gallen, =mcm Institute, St.Gallen, Switzerland*

² *SAP Research CEC, St.Gallen, Switzerland*

{christoph.schroth, oliver.christ} @ {sap.com}

Abstract

Web Services have experienced great interest during the last years as they were expected to play a key role as enablers of seamless application-to-application integration both within company boundaries and on a global, cross-organizational scale. As a technical foundation for the realization of Service-Oriented Architectures (SOAs), Web Services encapsulate complexity inherent to individual applications and allow for their loose coupling. However, a truly global mesh of such services has not yet become reality due to various reasons. Novel technologies and design principles are currently about to emerge which allow human users to use, customize, combine, interconnect and finally expose Web-based content or functionality as new resources which are often referred to as Mash-ups. In this article, we provide an overview of existing Mash-ups as well as tools and platforms that empower users to build them in a highly efficient and intuitive fashion. Statistical data and case studies are leveraged to examine new ways of resource provision and consumption and also the relevance of upcoming intermediaries. Finally, we investigate remaining research challenges on the path to a truly global SOA.

1. Introduction

The number of enterprises that automate cross-organizational business transactions is constantly growing. Renowned scientists such as Malone cite the relentless march of improvements in the cost-performance ratio of information technology as main driver of this development [1]. Benefits of performing transactions electronically include extending market reach, saving time, cutting costs and responding to customer queries more agilely [2].

Web Services [3] have experienced great interest as they were expected to play a key role as enablers of this seamless application-to-application integration both within company boundaries and on a global, cross-organizational scale. They allow for encapsulating proprietary application functionality with the help of

uniformly defined interfaces and thus facilitate the loose and quick coupling of potentially heterogeneous applications.

As argued in [4], **Service-Oriented Architectures (SOAs)** [5] on the basis of Web Services mostly only exist within company boundaries and have not yet taken over a significant role in realizing cross-organizational collaboration. Various reasons such as high technical complexity, implementation and maintenance costs, inflexibility and the lack of widely accepted standards for defining service choreographies as well as message semantics have prevented the emergence of a global mesh of interoperable Web services.

Novel technologies and design principles are now about to experience increasing acceptance as they allow for using, customizing, interconnecting and finally exposing Web-based content or functionality again as new resources. **Web 2.0** is frequently used as an umbrella term to summarize the collection of programming languages and general principles that can be considered not as substitute, but as enrichment of SOA concepts and technologies.

Numerous various definitions of Web 2.0 exist that mostly only describe certain aspects of the overall concept. Tim O'Reilly, who originally coined the term, identifies seven major characteristics inherent to the Web 2.0 concept [6]. First, the Web is considered as a platform for building systems that do not necessarily have a specific owner and are "tied together by a set of protocols, open standards and agreements for cooperation". Harnessing collective intelligence of Web users represents the second major paradigm. The ownership of mission-critical data is regarded a further cornerstone of numerous Web 2.0 applications. Fourth, O'Reilly propagates the end of the software release cycle as another central paradigm. The use of lightweight programming models that allow for loosely coupled systems and applications, the provision of software above the level of a single device and the realization of rich user experience represent the last major paradigms inherent to the concept of Web 2.0. Besides such analyses that properly describe parts of the super-ordinate concept, there are only very few comprehensive scientific definitions available. An in-depth investigation of numerous different, successful Web

2.0 applications performed by Hoegg et. al. [7] condensed the respective characteristics in the following statement which works as underlying definition for this work: “Web 2.0 is defined as the philosophy of mutually maximizing collective intelligence and added value for each participant by formalized and dynamic information sharing and creation”.

Besides social communities (e.g., LinkedIn.com, GiveMeaning.com) or online collaboration tools, **Mash-ups** are considered as central to the Web 2.0 era. *The Economist* defines Mash-ups as follows [8]: “The term Mash-up is borrowed from the world of music, where it refers to the unauthorised combination of the vocal from one song with the musical backing of another, usually from a completely different genre. Web Mash-ups do the same sort of thing, combining websites to produce useful hybrid sites and illustrating the internet’s underlying philosophy: that open standards allow and promote unexpected forms of innovation.” The *BusinessWeek* [9] explains this novel form of using Web-based resources similarly: “Mash-ups portend big changes for software companies, Web sites, and everyone online. No longer just a collection of pages, the Web is morphing into a sort of global operating system, à la Microsoft [...] Windows. And now, people are learning to program Web 2.0 with much of the same innovative energy of the personal computer’s early days.”

We basically adhere to these explanations and take the following summary as foundation for this work: *A 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.*

Mash-ups can be setup by using resources such as the OpenStreetMap¹ service, which provides free geographical data such as street maps. By leveraging lightweight protocols and a simple interface, it can be seamlessly integrated into other Web-based applications and thus be enriched with additional, arbitrary features such as pictures of local buildings or tourist attractions. Platforms and tools are provided in the Web that empower users to loosely couple readily available applications, to enrich and compose them into a novel service, which can again be made publicly available. As a result, meshes of different Web-based resources emerge that can be considered rather ungoverned since there is no central control entity existing that imposes formal guidelines for reuse and syndication. The possibility to actively contribute and gain reputation by publishing resources on the one side and to harness the aggregated knowledge and intelligence of all other platform participants on the other side represent core benefits for users.

Recently, the relationship between Mash-ups and traditional SOAs has experienced considerable interest

[10, 11]. In this article, we investigate this interrelation and propose Mash-ups as key enablers of a truly global SOA that is freely developed by human users who are fascinated and empowered by a set of Web 2.0 technologies and design principles.

In the following sections, we first of all provide an exact definition of SOA and also elaborate on the shortcoming of conventional, Web Services-based implementations with regard to the emergence of a global mesh of interoperable services (Section two). We then use the key three elements of a SOA, namely the service provider, the service consumer and a registry [5] to systematically analyze success factors of the above mentioned Mash-ups on the basis of statistical data and case studies (Section three). As argued below, new ways of resource provision and consumption as well as certain intermediaries are characteristic for the emergence of Mash-ups which can be considered new and human-driven forms of SOAs.

2. Vision and shortcomings of the traditional Web Service-stack to enable a global SOA

2.1. SOA and the vision of loosely coupling heterogeneous applications worldwide

Since the late 1990s, many definitions of SOA have been published [3, 12]. The normative OASIS Reference Model for SOA [5] defines SOA as “...a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations”.

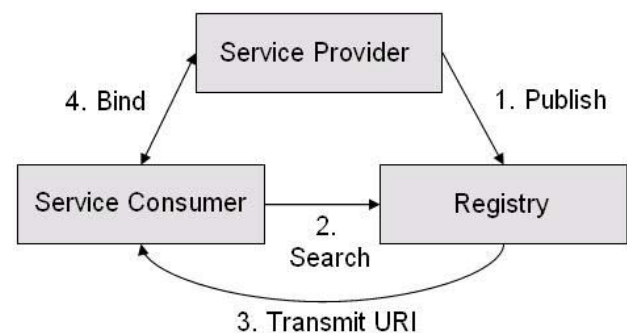


Figure 1: Basic SOA Architecture [5]

According to this model, the major components of a basic SOA and their possible interactions are (see Figure 1): a service provider publishes his service interface via a service registry where a service requester/ consumer can find it and subsequently may bind to the service provider. The central concept of the SOA Reference Model is the existence of services which provide access to capabilities

¹ <http://wiki.openstreetmap.org/index.php/REST>

by well-defined interfaces to be exercised following a service contract with constraints and policies. This enables a loose coupling of services (thereby minimizing mutual dependencies) and complies with some of the probably most-known principles in software-engineering, information-hiding and modularization [13]. Services are provided by entities, the service provider, and are to be used by others, the service consumers. Services may be composed on the basis of other, existing services, thereby adhering to the principle of reuse. They are autonomous (solely control the logic they encapsulate), uniformly described and publicly retrievable via certain discovery mechanisms.

The concept of supporting loosely coupled, business-aligned and networked services as introduced above can be realized with the help of numerous different technologies. As **WSDL**² and **SOAP**-based Web Services are the most widely spread standards used to setup SOAs, these are in the focus of this work [3]. The Web Service Description Language (WSDL) defines a uniform, machine-readable XML format for service interfaces. The Universal Description, Discovery, and Integration (**UDDI**³) standard specifies publicly available service registries (see Figure 1) that are needed for service search, identification and invocation. SOAP specifies the data format and an exchange protocol for the messages to be sent between service providers and requesters. While SOAP only defines formal aspects of data structure, the interpretation of information semantics has not yet been defined by a common standard. Machines may describe the same piece of information with different terms, thereby preventing from seamless system interoperability. The Business Process Execution Language (**BPEL**⁴) is widely accepted as a standard for imposing a choreography on different services. Also based on XML, it focuses on the description of a business process from one participant's point of view.

Many have generated great enthusiasm about Web Services, which are considered as "open, Internet-era standards for exchanging data between applications" [4]. With a SOA, tight couplings between the applications of different companies are expected to be replaced by loose couplings. As a consequence of common standards for data description and connection protocols, applications can be seamlessly connected to others without costly reprogramming efforts. By loosely coupling application functionality via commonly defined interfaces, companies are enabled to modify their operations more agilely in response to changing business requirements. Summing up, especially companies expected the dawn of Web Services to turn traditional supply chains inside out and to replace static cross-organizational business relations by a

flexible mesh of loosely coupled services which can be identified via publicly accessible registries.

2.1. Shortcomings of the conventional Web Services stack with regard to realizing a global SOA

Right now, Web services have not lived up to this promise as mostly people are still involved in loose couplings all over the world. Rather than automatically inter-connecting applications across company boundaries with the help of Web Services technologies, most loose couplings are based on "e-mail and instant-messaging applications, groupware such as Lotus Notes/ Domino and nascent "social software" such as wikis, Friendster.com and LinkedIn.com" [4]. As argued in [14], the "corporate household" problem prevents Web Services-based SOAs from being setup quickly and easily. Companies and their respective systems are subject to rigid governance mechanisms [15] and adhere to different conventions with respect to business documents and also internal business processes: "The applications used in different departments or different locations of a single company, for example, are virtually guaranteed to have inconsistency in customer lists, part-numbering schemes, approved suppliers, addresses for these suppliers and so on" [4].

Besides the heterogeneity of conventions with respect to data structures, semantics and processes, the lack of comprehensive, trustworthy and widely accepted service registries prevents global SOAs from quickly establishing. The few globally available UDDI-based registries are mostly only usable for technical experts, do not offer any additional functionality such as service performance monitoring and only feature a small number of references to different Web Services.

Apart from that, the traditional Web Services stack aims at supporting the setup of loosely coupled application interconnections especially in a professional context and assumes users to be technically sophisticated. This prevents normal Internet users from easily discovering and interacting with services available via the Web. WSDL-based service interfaces, but also the UDDI-compliant service registries are not designed to be interpreted and used by humans but rather by machines.

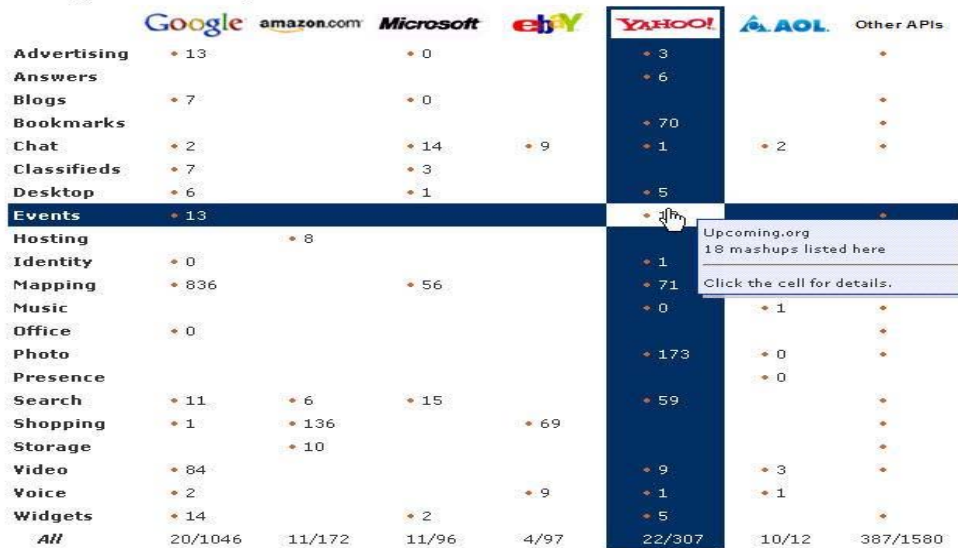
2. Web 2.0 design principles and technologies as enablers of a global SOA

² <http://www.w3.org/tr>

³ <http://www.oasis-open.org/>

⁴ <http://www-128.ibm.com/developerworks/>

Figure2: Mash-up API Scorecard



After elaborating on the actual vision and the promise of traditional Web Services to allow for the efficient setup of loosely coupled application interconnections, we present emerging design principles and technologies which enable a global Service-Oriented Architecture which is driven by individual Web users and their requirements. By introducing new ways for **resource provision and consumption**, but also as a consequence of the emergence of **new intermediaries**, the major obstacles which traditional Web Services struggled with can be overcome. In this way, Mash-ups can be freely designed by individuals on the basis of lightweight programming models and interfaces. The following sections will thoroughly investigate the prerequisites for the success of these Mash-ups and thereby leverage statistical data as well as case studies.

3.1. Resource provision

As mentioned above, traditional Web Services are provided as functionality described by uniform interfaces (mostly WSDL compliant) that define input and output messages as well as the supported functions of the services. These interfaces are not human-readable and do not facilitate the interaction of users with the underlying services. Resources in the Mash-up context do not aim at technical experts in the corporate context any more, but at the huge number of individuals, the “long tail” [16] of Internet users who require intuitive visual means for retrieving resources in the Web and for capturing their respective functionality.

The Website **ProgrammableWeb.com** acts as a major aggregator of numerous (currently more than 1600) references to diverse existing Mash-ups all over the

world. It provides statistics about used resources, a classification of Mash-ups into categories as well as user statistics to evaluate their popularity. Therefore it serves as one central source of empirical data used in this work.

To investigate the most frequently used resources and their respective Application Programming Interfaces (APIs), the Web Site exposes an “API-Scorecard” (Figure 2) which has six major API-providers on the X-axis and the most relevant categories of APIs offered on the Y-axis. In this context, the term API refers to the interface type, data formats and the protocol to be used for interconnecting to an application. The last line of the figure lists the number of APIs referenced on the Website and the number of Mash-ups using one of these APIs for each of the 6 major providers. For example, 20 Google-based APIs and 1046 Mash-ups that leverage one of these APIs are listed on ProgrammableWeb.Com.

Several facts can be inferred from this graphic: First, large companies such as Google, Amazon and Microsoft start to successfully provide Web-based resources which are leveraged by numerous users for creating Mash-ups on their basis. As can be seen in Figure 2, at least 836 Web-based applications have already integrated the “Google Maps”- resource that offers geographical data. The Website “Vuelta a California” can be cited⁵ as an excellent example for a Google Maps-based Mash-up. It uses information from a Web log that deals with road profiles for bikers in California and mashes it with road map data from Google Maps to achieve a rich visual representation.

The second important fact that can be inferred from the API scorecard is the shift away from the professional corporate context towards a private, end-user driven field

⁵ <http://www.berndstraehe.com/map.html>

of applications. As opposed to the focus of traditional Web Services, the resources used for building Mash-Ups aim at the long tail of Internet users and deal with media management, shopping functionality, entertainment or desktop applications. Heavy-weight enterprise applications for automating business transactions can only rarely be found in this context.

Third, the Website also provides information about the protocols and interface standards used by the different listed resources. The analysis of the provided data leads to the following results: Besides the **SOAP** protocol which is also used in the context of traditional Web Services, particularly “light-weight” protocols such as REST [17] and RSS are widely leveraged to allow for fast and seamless mashing of different resources.

The Representational State Transfer (**REST**) is an architectural style that has been proposed by Fielding in his doctoral dissertation. With the help of REST, Web clients can interact with arbitrary Web resources (sources of specific information) in a uniform way and thereby exchange XML-based messages. Resources are uniquely addressable by means of a global identifier (URI) and support a well defined and also constrained set of supported operations (so-called HTTP verbs such as GET, PUT). By using resource identifiers and HTTP verbs together, clients can invoke REST-based services and request the delivery of, for example, an XML file containing the desired specific information. The Simple Storage Service (S3⁶) provided by Amazon is one of many exemplary applications of REST technology: On the basis of a simple service interface, users are enabled to store any amount of data on Amazon servers and also retrieve it from arbitrary locations on the Web. REST incorporates the principles of complexity-hiding and uniformity since it significantly reduces programming efforts and forces both clients and recourses to adhere to a common set of supported operations and interface descriptions.

The Really Simple Syndication (**RSS**⁷) format represents a further technology that is affiliated with numerous resources in the Web 2.0 context. An XML based file format, RSS supports the easy aggregation of content from arbitrary sources in the Web. With the help of this standard, the information offered by frequently changing Web sites such as Web logs can be exported into XML files of a uniform structure and is thus made publicly available. The files’ structures are determined by the RSS specification which intentionally only allows for a very limited set of content categories such as title, Web link and textual description of a piece of information. Web applications that aim at distributing content via RSS technology must ensure that the respective data files always contain the latest information. Applications that

present the information obtained from different RSS-based sources in a combined manner are referred to as content aggregators. Users are thus enabled to gather information that has been created by a number of authors and is easily accessible via uniform interfaces, redesign it and also publish it.

Summing up, the Web increasingly can be considered as a comprehensive and global development platform containing numerous easily usable and mashable resources that are provided by large firms as well as by individual end-users. As argued above, the provision of resources that use lightweight protocols such as REST or RSS and also the focus on end-user requirements rather than Business-to-Business relationships represent core success factors for this new global SOA consisting of numerous Mash-ups. Figure 3 finally shows the currently most popular providers of resources leveraged for the creation of Mash-ups.

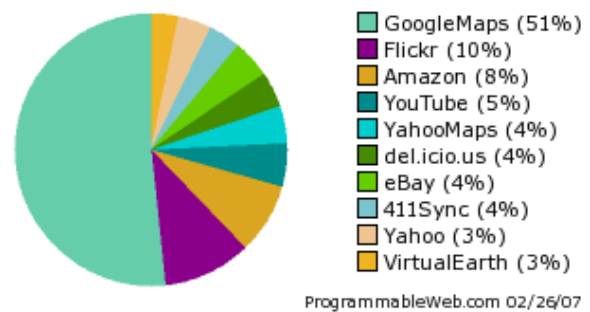


Figure 3: List of most popular Mash-up resource providers (according to ProgrammableWeb.com)

3.2. Emerging intermediaries

As presented above, a publicly available registry constitutes one of the three major elements of a SOA. UDDI is widely accepted as standard and offers a platform-independent, XML-based registry specification for stakeholders worldwide to list the services they provide via the Web. The key information published by such a registry contains white pages (address, contact, and known identifiers of the service providers), yellow pages (industrial categorizations based on standard taxonomies) and finally green pages (technical information about services exposed by the business).

Similar to the technologies used for describing services in traditional SOA implementations (e.g. WSDL), the UDDI standard aims at technical experts and only specifies very limited functionality. Merely UDDI-based registries are not adequate to serve the large number of normal Internet user who require intuitive, human-readable interfaces to identify and to evaluate existing services and to finally integrate and mash them into their applications.

⁶ <http://www.amazon.com/gp/browse.html?node=16427261>

⁷ <http://www.rssboard.org/rss-specification>

Novel forms of intermediaries are currently about to emerge which offer resource registry functionality and extend the role of the traditional UDDI-based implementations. As depicted in Figure 4, they do not only contain references to services, but also offer statistics about resource performance or information about evaluations conducted by previous users.

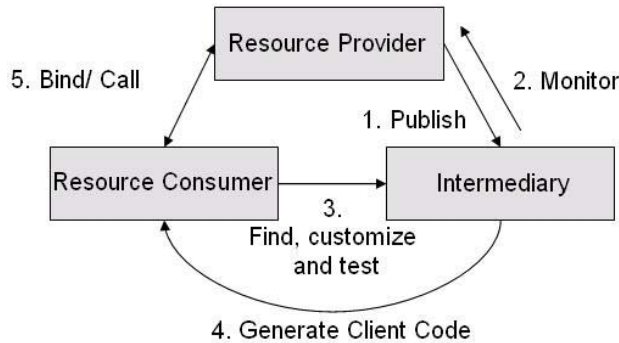


Figure 4: Advanced intermediaries facilitate retrieval and use of Web-based resources

The typical resource discovery and utilization process is therefore different from the one depicted in Figure 1. A resource provider first of all publishes a reference to his resource at the intermediary which stores it and subsequently starts to monitor parameters such as general resource availability and response latency. Potential consumers may then access the intermediary's Website and use the performance metrics and other evaluation results while searching for adequate resources. Some intermediary platforms even allow for building customized code for integrating selected resources into the users' Mash-ups in order to relief them from any programming efforts. The intermediary **WebRPC**⁸, for example, offers this functionality and uses the advertising slogan "Find, customize, monitor, and call SOAP, REST, or HTML services in minutes". After performing these four steps, the user may finally call the customized resource.

Besides such one-stop-shop resource registries, other kinds of intermediaries are about to emerge which all follow the goal to simplify user/ service interaction and thus facilitate the development of a global SOA. The platform **FeedBurner**⁹ represents an excellent example for such intermediaries: It allows users to register RSS-compliant pieces of information ("feeds") and provides users such as bloggers, podcast publishers and other Web-based content creators with several value-added services. After registering the URL of an RSS feed from an arbitrary source on the platform, FeedBurner offers free readership statistics, promotion services that make sure

feed search engines are notified about updates as well as a human-readable and harmonized feed address as a substitute for the mostly heterogeneously designed original addresses. In this way, users benefit from both flexibility and persistence: In case the actual source of a user's feed changes, he can edit the respective entry at the central Feedburner platform and thus ensures that the potentially high number of users who read or even mash the feed with other resources are not affected by this change. For them, the feed's address remains the same. According to the information provided by the platform, as of August 2006, FeedBurner hosts feeds for 250,000 publishers, and delivers feeds to 19 million subscribers.

The emergence of such intermediaries that act as central resource publishing and management platforms can be motivated with help of **institutional economics**. According to North [18], the reason for institutions to exist is the uncertainties involved in human interaction; institutions act as constraints and framework to structure that interaction and make it more reliable. The proper establishment and application of informal constraints, formal rules and enforcement are adequate to lower transaction costs and boost economic performance. Both the WebRPC and the Feedburner platform are recognized as trustworthy central points of access to different kinds of Web-based resources. By monitoring performance and by gathering user-based evaluations, they lower the risk for new users to select a bad performing resource for their Mash-up. They also act in a normative fashion as they provide common standards for naming and format and thus reduce uncertainty of resources to be unavailable as a consequence of a URL change, for example.

3.3. Use of resources

After investigating new principles of offering resources and after presenting novel forms of intermediaries which provide institutional frameworks, we deal with the changes regarding user-side consumption of resources in this section. Before presenting three major cases of tools and platforms that empower users to mash up almost arbitrary kinds of web-based content or application functionality, we first present a Top Ten list of Mash-up categories (Figure 5) as of February 2007 and according to the data gathered by the ProgrammableWeb.com platform which references more than 1600 different Mash-up use-cases. As can be seen in this figure, end-user centric applications which are related to geographical maps, media management or online shopping prevail, while enterprise class Mash-ups do not yet occur very frequently.

Many of the Mash-ups cited and examined above have been created by individuals who are still technically experienced to some degree. Light-weight programming models and protocols as well as intermediaries which support the retrieval and evaluation of resources facilitate

⁸ <http://www.webrpc.com/>

⁹ <http://www.feedburner.com/>

the Mash-up setup procedure. However, some programming effort is still required in those cases. The following three major cases of Mash-up-platforms highlight a rather recent way of even empowering users without any technical sophistication to create and change Mash-ups and to thus foster the emergence of a user-driven, global SOA.

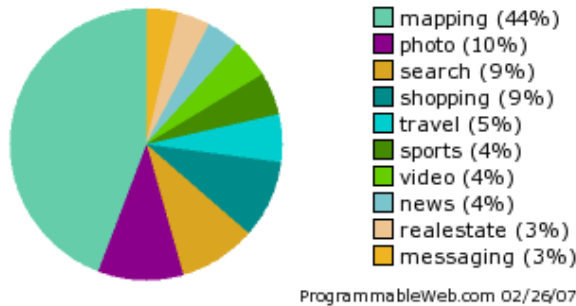


Figure 5: Top ten list of Mash-up categories

3.3.1. Case studies

The **Google Website**¹⁰ represents a first intuitive example for a Mash-up platform as it allows users to aggregate and arrange different “gadgets” which represent either mere content (e.g., pictures, text, weather forecasts, Web cams) or certain application functionality (e.g., calculators). An intuitively usable gadget repository can be accessed for free and enables users to select and subsequently drag and drop some of the gadgets on their individual user interfaces. Finally, individuals are also empowered to create and register gadgets on their own by leveraging freely available templates and by complying with the Google de-facto standard. This kind of Mash-up-platform facilitates the mere aggregation of resources (which is also often referred to as syndication) but does not foresee any kind of interaction between them. Gadget interoperability issues are thus completely avoided since the resulting Mash-ups solely reside on the presentation layer. The gadgets offered by Spring Widgets¹¹ also belong to this category of presentation-layer syndication Mash-ups.

Yahoo pipes¹² represent an example for a second category of Mash-up platforms which are not limited to mere resource syndication. Figure 6 depicts an exemplary use-case for this platform: The “pipe” (which has been designed by a user) reads in an RSS-based feed provided by the New York Times homepage, passes it through a customizable content analysis (marked orange) and uses filtered keywords to find related photos at the widely known media repository Flickr¹³. The pipe’s output issues

links to the photos in a RSS-compliant way. Mash-up platforms like this are different from the ones assigned to the first category as they involve the processing of resources rather than their mere syndication. Content from the New York Times website is filtered and parts of another feed are selected according to the result of the filtering at the time the user triggers the Mash-up application. This advanced category assumes the resources to comply with certain standards to allow for their seamless connection.

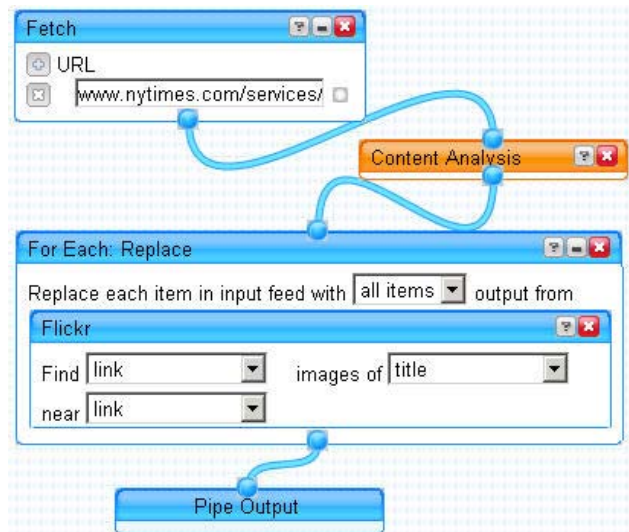


Figure 6: Yahoo Pipes facilitate mashing of content from arbitrary sources

Still, resources are only loaded and processed at one point of time and do not interact in a stateful fashion similar to Web Services that are called according to a previously defined choreography. Only very few examples exist that try to automate the mashing of application functionality rather than their simple aggregation. The firm **Kapow Technologies**¹⁴ has recently published its vision of so-called Enterprise Mashups. It focuses on empowering users of their Web Integration Platform to integrate Web-based resources on several different levels and thus efficiently realize rapid portal creation, content migration, synchronization and syndication, composite application integration and SOA enablement. According to the firms publications, logic-based Mash-ups can be setup on the basis of a merely visual, technology- and complexity hiding modeling interface. Enterprise applications which are encapsulated as, for example, Web Services or by REST-based interfaces can be mashed up to build new functionality.

3.3.2. Results

¹⁰ <http://www.google.com/>

¹¹ <http://www.springwidgets.com/>

¹² <http://pipes.yahoo.com/>

¹³ <http://flickr.com/>

¹⁴ <http://www.kapowtech.com/>

The analysis and categorization presented above shows that mainly three types of platforms have emerged which all aim at empowering individuals to mash up resources in an as quick and intuitive manner as possible. First, platforms such as Google facilitate the mere syndication of gadgets which are not interlinked at all. Second, environments such as Yahoo Pipes allow for processing and interconnecting resources that adhere to a certain standard (in this case RSS) and expose the result of this processing as a new resource. The mashing of resources on an application logic-level represents the third and last category investigated in the course of this work. Only very few examples exist which are capable to partly realize this sort of Mash-up (e.g., Kapow technologies).

Projects are under way and start-up firms are created that tackle this remaining challenge on the path to a global SOA that is driven by end-users and facilitated by lightweight programming models and intuitive Mash-up platforms and tools.

3. Conclusion

In this article, we elaborated on the Web Services stack as traditional enabler for coupling distant and potentially heterogeneous Web-based resources. As Web Services have not lived up to their promise to facilitate a global network of loosely interconnected services, new approaches are required to cope with this challenge. As a key component of the Web 2.0 era, novel technologies and design principles are now about to emerge which allow humans to use, customize, combine, interconnect and finally expose Web-based content or functionality as new resources on the Web. The results of reusing and composing existing resources are referred to as Mash-ups.

In this work, we provided an overview of already existing Mash-ups as well as different tools and platforms that empower users to build them in a highly efficient and intuitive fashion. Statistical data and case studies were leveraged to examine new ways of resource provision and consumption and also the relevance of upcoming intermediaries. Finally, we investigated remaining research challenges on the path to a truly global SOA: The development of platforms and tools for the automated mashing of application functionality rather than the mere syndication of resources will represent a key milestone which has to be achieved in future work.

Acknowledgments

The term “Brave New Web” has been brought up in the Web 2.0 context by Kenneth P. Morse, Managing Director, MIT Entrepreneurship Center

5. References

- [1]T. Malone: *The Future of E-Business*, Sloan Management Review, Vol. 43, No. 1, p. 104, 2001
- [2]M. Porter, *Strategy and the Internet*, Harvard Business Review, 2001
- [3] G. Alonso, F. Casati, H. Kuno, V. Machiraju: *Web Services Concepts, Architectures and Applications*, in: Springer, 2004
- [4] A. McAfee: Will Web Services Really Transform Collaboration, in: MIT Sloan Management Review, Vol.46, No.2, 2005
- [5]M. MacKenzie, K. Laskey, F. McCabe, P. F. Brown, R. Metz: OASIS - Reference Model for Service Oriented Architecture 1.0, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm, 2006
- [6]O'Reilly: What is Web 2.0, <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>, last seen: February 23, 2007
- [7]R. Högg, M. Meckel, K. Stanoevska-Slabeva, R. Martignoni: Overview of business models for Web 2.0 communities, Proceedings of GeNeMe 2006, p. 23-37, Dresden, 2006
- [8]Mashing the Web, *The Economist* 376 (8444), Special section, p. 4, 2005
- [9]R. D. Hof: Mix, Match, And Mutate, *Business Week Magazine*, 2005, available online at: http://www.businessweek.com/@@76IH*ocQ34AvyQMA/magazine/content/05_30/b3944108_mz063.htm, last seen: February 23, 2007
- [10]D. Hinchcliffe: i-Technology Viewpoint: Is Web 2.0 the Global SOA?, *SOA Web Services Journal*, <http://webservices.sys-con.com/read/164532.htm>, last seen: February 23, 2007
- [11]A. McAfee: Enterprise 2.0 vs. SOA, in: Harvard Business School Faculty Blog, http://blog.hbs.edu/faculty/amcafee/index.php/faculty_amcafee_v3/enterprise_20_vs_soa/, last seen February 28, 2007
- [12]P. Frost, S. Frost: *Component-based development for enterprise systems: applying the SELECT perspective*, in: Cambridge University Press, 1998
- [13]D. L. Parnas: On the criteria to be used in decomposing systems into modules, in: *Commun. ACM*, Vol.15, No.2, ACM Press, 1972
- [14]X. Chen, J. Funk, S. Madnick, R. Wang: *Corporate Household Data: Research Directions*, Proceedings of the Americas Conference on Information Systems, Boston, August 2001
- [15]P. Weill and J.W. Ross: *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results*, Harvard Business School Press, May 2004
- [16]C. Anderson: *The Long Tail: Why the Future of Business Is Selling Less of More*, Hyperion Books, 2006
- [17]R. T. Fielding: *Architectural styles and the design of network-based software architectures*, Ph.D. thesis, University of California, Irvine, 2000
- [18]D. C. North: *Institutions, Institutional change and economic performance*, Cambridge University Press, Cambridge, 1990

By examining and contrasting Web 2.0 and SOA, the authors envision a new Internet of Services that incorporates the best of both.

Christoph Schroth and Till Janner



Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services

Recently, the relationship between Web 2.0 and service-oriented architectures (SOAs) has received an enormous amount of coverage because of the notion of complexity-hiding and reuse, along with the concept of loosely coupling services (see <http://blog.hbs.edu/faculty/amcafee/index.php>). Some argue that Web 2.0 and SOAs have significantly different elements and thus can not be regarded as parallel philosophies (see http://edgeperspectives.typepad.com/edge_perspectives/2006/04/soa_versus_web_.html). Others, however, consider the two concepts as complementary and regard Web 2.0 as the global SOA. In this article, we investigate these two philosophies and their respective applications from both a technological and business perspective.

DEFINITION OF TERMS AND RESEARCH APPROACH

Tim O'Reilly coined the term Web 2.0 to describe a quickly growing set of Web-based applications. SOA is considered the philosophy of encapsulating application logic in services with a uniformly defined interface and making these publicly available via discovery mechanisms (C.M. MacKenzie et al., "Oasis—Reference Model for Service Oriented Architecture 1.0," 2006; <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf> and

A. McAfee, "Will Web Services Really Transform Collaboration," *MIT Sloan Management Review*, vol. 46, no. 2, 2005, pp. 78-84).

O'Reilly identifies seven major characteristics inherent to the Web 2.0 philosophy: first, the Web is considered a platform for building systems that are "tied together by a set of protocols, open standards, and agreements for cooperation" (T. O'Reilly, 2005, <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>). The exploitation of collective intelligence of Web users, ownership of mission-critical data, and the end of the software release cycle are quoted as central characteristics as well. The use of lightweight programming models that allow for loosely coupled applications, the use of diverse media and devices for the consumption of Internet-based applications, and the realization of rich user experiences represent further paradigms inherent to the concept of Web 2.0.

In 2006, Högg et al. conducted an in-depth investigation of 40 successful Web 2.0 applications (R. Högg et al., "Overview of Business Models for Web 2.0 Communities," *Proc. Gemeinschaften in Neuen Medien*, Technische Universität Dresden, 2006, pp. 23-37). They condensed their respective characteristics into the following statement, which works as an underlying definition for this article: "Web 2.0 is defined as the philosophy of mutually

maximizing collective intelligence and added value for each participant by formalized and dynamic information sharing and creation.”

Since the late 1990s, many definitions of SOA have been published (P. Frost and S. Frost, *Component-Based Development for Enterprise Systems: Applying the Select Perspective*, Cambridge Univ. Press, 1998 and G. Alonso et al., *Web Services Concepts, Architectures, and Applications*, Springer, 2004). The Oasis Reference Model for SOA (MacKenzie et al., 2006) defines SOA as

... a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with, and use capabilities to produce desired effects consistent with measurable preconditions and expectations.

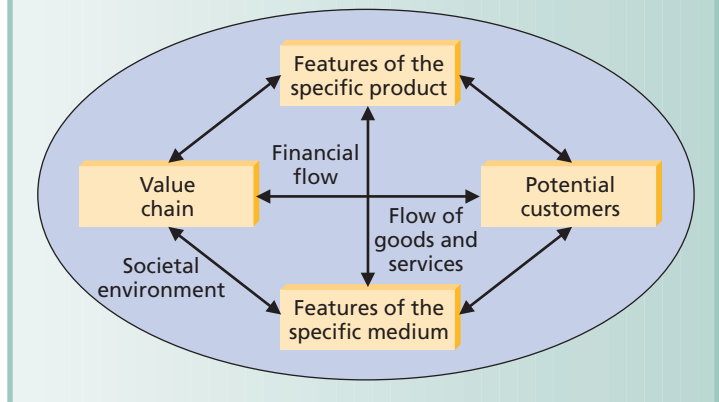
According to this model, a service provider might publish a well-defined interface on a registry that enables other stakeholders to retrieve and loosely couple the offered service with their own services.

The research approach applied for this comparative analysis is based on the examination of 40 real-world use cases of both Web 2.0 and SOA. In the case of Web 2.0, 40 popular applications (Högg et al., 2006) were selected, while case studies provided by Systeme Anwendungen und Produkte in der Datenverarbeitung (SAP), IBM, and Gartner were leveraged to investigate SOA solutions.

The Institute for Media and Communications Management (MCM) at the University of St. Gallen’s business model framework (see Figure 1) was leveraged to structure the analysis along seven major components (Högg et al., 2006):

1. *Features of the specific product* comprise the design of a product or service and its value for the customer.
2. *Features of the specific medium* define the technological foundation of a product.
3. The *customers* component refers to the target groups of an offered product or service and explains their respective business needs.
4. The *value chain* is devoted to reflecting all players that are involved in the production and delivery of a product and their respective interrelationships.
5. *Financial flow* identifies revenue models and explains the roles that different stakeholders play.
6. *Flow of goods and services* describes the stakeholders’ activities that are essential for the product’s or service’s creation.
7. Last, the *societal environment* reflects relevant outside influences on a business model (such as legal and social aspects and competitive situations).

Figure 1. The Institute for Media Communications Management’s (MCM’s) business model framework.



WEB 2.0 VS. SOA

In comparing each product or service, it’s worthwhile to consider Web 2.0’s and SOA’s pros and cons more thoroughly. In the following paragraphs, we analyze Web 2.0 and SOA based on the MCM business model framework’s seven major components.

Features of the specific product or service

According to Högg et al. (2006), we can classify Web 2.0 applications as follows:

- Communities that aim to unify their users by means of a common ideal such as social networking or knowledge sharing.
- Platforms or tools that help users create and share content with a broad audience (for example, Web logs and online directories). Mashup platforms let users retrieve content or functionality from arbitrary sources, mix it with other resources, and expose it for further reuse by other applications.
- Online collaboration tools support users in collaboratively performing certain tasks, such as maintaining time schedules or processing text online.

In comparison, the following are ways we can differentiate SOA use cases:

- First, SOAs allow for a cross-organizational integration of services. By adhering to common standards for the description of their service interfaces, corporations are enabled to setup loosely coupled electronic business transactions with other companies and thus automate business transactions in a quickly changeable fashion.
- Second, SOAs facilitate the intraorganizational integration of disparate services. On the basis of a central integration layer (often referred to as an Enterprise

Service Bus, or ESB), heterogeneous applications can be encapsulated and composed to a seamlessly integrated IT landscape (R.W. Schulte, "Predicts 2003: Enterprise Service Buses Emerge," Gartner Research Note, 2002; http://www.gartner.com/DisplayDocument?doc_cd=111977).

- Third, SOA-based application development significantly reduces development time thanks to the availability of reusable application building blocks.

The first major analogy between product design in the fields of Web 2.0 and SOA is the notion of reusing and composing existing resources. Both concepts let users reuse, remix, and enrich existing resources and components to new and potentially higher-level applications. The second commonness is the affinity to collaboration and coupling of remote resources or services. Both Web 2.0 and SOA applications enable the loose coupling of distant and possibly heterogeneous resources. A third apparent resemblance between Web 2.0 and SOA is the shared principle of agility and the support of permanent structural change.

Web 2.0 and SOA also have divergent elements. First of all, many Web 2.0 applications incorporate a social aspect, as they facilitate human interaction and also mainly deal with human-readable content (such as text and pictures). In contrast, conventional SOAs merely aim at interconnecting dispersed business functionality and facilitating seamless machine-machine collaboration. Second, Web 2.0 is clearly about presentation and user interface integration, whereas SOA deployments are more abstract and less visible to its users. Third, and last, the degree of ex-ante determination and involved governance (McAfee, 2005) is a key differentiator between Web 2.0 and SOA. Because of their frequent implementation in the corporate context, SOAs are subject to requirements that don't exist in the case of most Web 2.0 applications and thus underlie governance mechanisms (P. Weill and J.W. Ross, *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results*, Harvard Bus. School Press, 2004).

Features of the specific medium

As argued in Kolbitsch and Maurer, Web 2.0 doesn't stride along with a fundamental technological innovation, but is facilitated by a number of technologies (J. Kolbitsch and H. Maurer, "The Transformation of the Web: How Engineering Communities Shape the Information We Consume," *J. Universal Computer Science*, vol. 12, no. 2, 2006, pp. 187-213). First of all, the Representational State

Transfer (REST) is an architectural style that enables Web clients to interact with arbitrary Web resources in a uniform way (R.T. Fielding, "Architectural Styles and the Design of Network-Based Software Architectures," doctoral dissertation, Information and Computer Science Dept., Univ. of California, Irvine, 2000). The Really Simple Syndication (RSS) format supports the easy aggregation of content from arbitrary sources in the Web. AJAX represents a composite of several other technologies that together allow for rich user experiences, which is one of the key paradigms of Web 2.0 applications.

As opposed to lightweight Web 2.0 technologies (Hagel, 2006), SOAs rely on a set of more complex standards. As the Web Service Description Language (WSDL) and SOAP-based Web services are the most widely spread standards used to set up SOAs, these are in the focus of our technical analysis.

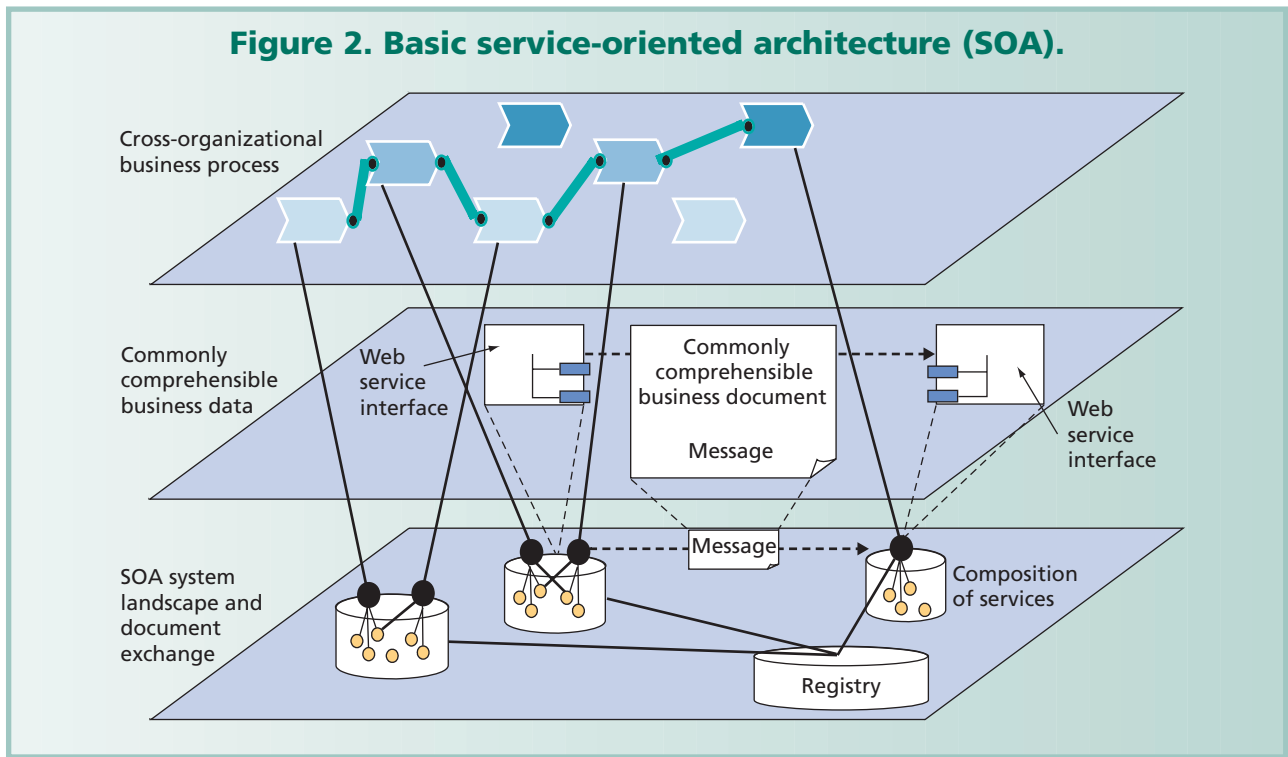
As Figure 2 shows, possibly heterogeneous applications (visualized as orange circles) are encapsulated as services and can be composed to new, aggregated functionality (black circles). WSDL is used for uniform service interface descriptions, while the Universal Description, Discovery, and Integration (UDDI) standard allows for open-service discovery. SOAP specifies a protocol for message exchange between services, while no widely accepted standard exists for data semantics (see the middle layer of Figure 2). The Business Process Execution Language is frequently used as a standard for orchestrating different services into one process choreography (see the upper layer in Figure 2).

As a first major technical similarity, standards applied in both Web 2.0 (such as RSS and REST) and SOA applications (such as WSDL) support the loose coupling of remote applications via uniform interfaces. Second, Web 2.0 and SOA technologies also share the notion of complexity hiding and reduced programming effort with the help of uniform descriptions of interfaces and data structures.

We can identify two major differences between the technological basis of Web 2.0 and SOA: First, we can make a distinction between the terms *syndication* and *coordination*. Many technologies used in the Web 2.0 context focus on static syndication of content and services, while SOAs often incorporate service coordination protocols (different services are invoked in a predefined sequence). The second difference refers to semantic interoperability. Web 2.0 applications directly involve human beings who are fault tolerant with respect to the information they're provided, whereas SOAs typically are limited to the mere interaction of machines, which aren't flexible to formal errors or semantic differences.

Because of their frequent implementation in the corporate context, SOAs are subject to requirements that don't exist in the case of most Web 2.0 applications.

Figure 2. Basic service-oriented architecture (SOA).



Potential customers

In the Web 2.0 context, basically every Web user can be regarded as a potential customer. Instead of heading for a small number of huge customers, Web 2.0 applications involve the bulk of private users or small businesses, also known as “the long tail” (C. Anderson, *The Long Tail: Why the Future of Business Is Selling Less of More*, Hyperion Books, 2006). As argued by O’Reilly, Web 2.0 is about leveraging customer self-service and thus is able to “reach out to the entire Web, to the long tail and not just to the head” (O’Reilly, 2005). In the examined SOA cases, medium-sized or larger corporations are the customers of choice. These aim to introduce SOA as a software design principle, as a possibility to streamline and harmonize internal IT landscapes or to set up cross-organizational business relationships. Because of their substantially different application domains, Web 2.0 and SOA applications serve different customer needs and requirements.

Value chain

In the Web 2.0 context, traditional value chains are broken up to a large extent and substituted by loose networks of providers and consumers (Högg et al., 2006). Every user may publish his or her own content or functionality on the Web and thus become a platform operator, consuming resources or reusing them to compose new applications and make them publicly available. In contrast, the high technical complexity inherent to SOAs requires the existence of one or several expert players within the value

chain who 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), Web 2.0 value chains are mostly loosely coupled (many-to-many) networks of self-managed users who can offer and consume resources via the Web.

Financial flow

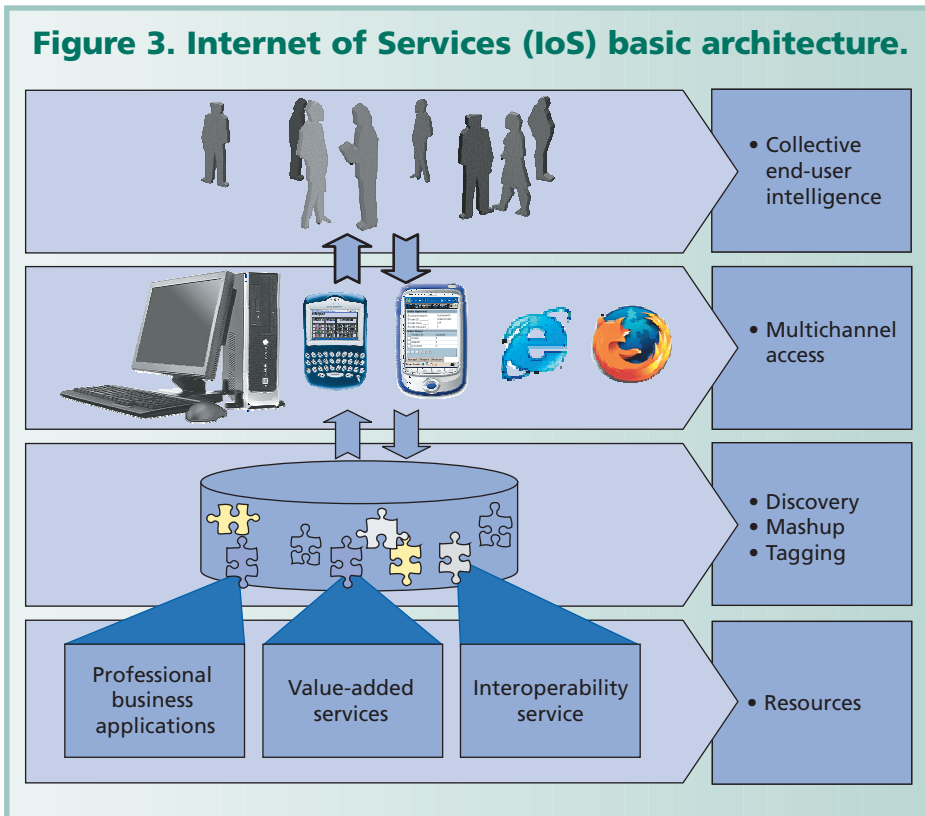
Revenue models of Web 2.0 applications strongly differ from traditional models in the software industry. First, software applications aren’t packaged and sold as over-the-counter products anymore, but provided as services (or the “end of the software release cycle,” Högg et al., 2006). For a significant share of Web 2.0 applications, the number of users is part of their central value, so providers generally don’t introduce service fees. They take this approach to avoid limiting the amount of users, but then this approach leads to challenges with monetizing their products.

The integration of third-party advertisements represents one possible solution to this revenue model dilemma. SOA revenue models in most cases follow a more traditional approach, as license fees are charged for using the respective solutions. This could change in the next few years as enterprise services become easily retrievable and usable via the Web.

Flow of goods and services

The decisive difference between typical product creation and provision in the fields of both Web 2.0 and SOA results

Figure 3. Internet of Services (IoS) basic architecture.



CONVERGING WEB 2.0 AND SOA CONCEPTS

In general, the philosophies of Web 2.0 and SOA serve different user needs and thus expose differences with respect to the design and used technologies of real-world applications. However, recently numerous novel use cases demonstrate the great potential of combining the technologies and principles of Web 2.0 and SOA.

One major example of the convergence of the two philosophies is the emergence of a global SOA that we refer to as Internet of Services (IoS). Up to now, normal Internet users with little IT sophistication haven't been able to easily retrieve and use certain services. This is because these services mostly reside within company boundaries and are only accessed for professional use in a corporate

context. However, the provision of easily accessible services for end users might drive a novel generation of Internet use and allow for the ad-hoc setup and configuration of IT-supported business models. Without an intuitive "face" toward human users, Web-based services will remain fairly unusable by the common Web surfer, because currently these interfaces (such as those described in WSDL) are designed to be processed by machines.

Figure 3 visualizes a basic IoS architecture that's based on a combination of principles and technologies from both Web 2.0 and SOA. Resources that are accessible via the Web are registered in platforms and can thus be discovered, tagged (to collect user evaluation and to allow for folksonomies), and also mashed up (composed and interlinked with the goal of designing new resources) according to the users' requirements (O'Reilly, 2005).

Societal environment

Arbitrary stakeholders can provide and host services, thereby leading to a global market with decentrally organized platforms that act as brokers. The actual users can access these platforms for discovery purposes via intuitive interfaces that facilitate tagging and mashing activities without requiring any coding effort. Then we could leverage all of the different channels (such as PCs or mobile devices) to enter and use the platforms. The establishment of an open architecture that comprises possibilities for human beings to use and interact with Web-based resources has the potential to drive the development of a global mesh of services.

As we previously noted, Web 2.0 use cases considerably benefit from the lack of formal guidelines and governance mechanisms with which SOAs typically must cope. However, cases exist where providers of Web-based platforms have been enforced to review the content they publish, thereby limiting the growth and dynamics of the offered solutions (Högg et al., 2006). As opposed to most Web 2.0 applications, SOAs are subject to clearly defined regulatory frameworks (such as the Sarbanes-Oxley Act), because they mostly exist in the corporate context. The design, provision, maintenance, and coupling of services must be compliant with legal frameworks and thus they do not allow for flexibility as observed within the Web 2.0 context.

Also, numerous Web 2.0 application providers don't rely on traditional marketing and sales activities but aim at *viral marketing*—that is, recommendations autonomously propagating from one user to another. In the SOA context, large solution providers such as SAP and IBM mostly treat SOA products as software artifacts that are packaged and then sold to customers.

Professional use in a corporate context. However, the provision of easily accessible services for end users might drive a novel generation of Internet use and allow for the ad-hoc setup and configuration of IT-supported business models. Without an intuitive "face" toward human users, Web-based services will remain fairly unusable by the common Web surfer, because currently these interfaces (such as those described in WSDL) are designed to be processed by machines.

Figure 3 visualizes a basic IoS architecture that's based on a combination of principles and technologies from both Web 2.0 and SOA. Resources that are accessible via the Web are registered in platforms and can thus be discovered, tagged (to collect user evaluation and to allow for folksonomies), and also mashed up (composed and interlinked with the goal of designing new resources) according to the users' requirements (O'Reilly, 2005).

A combination of principles from both Web 2.0 (user self-service and collective end-user intelligence) and SOA (a composition of reusable building blocks) can facilitate the wide dissemination of many resources. Examples include professional business applications, value-added services (including location-based services), and interoperability services (for example, applications that can be leveraged by trading partners to initiate business-to-business transactions).

Enterprise mashups represent one specific use case of this type of architecture that could easily be situated at the interstice of Web 2.0 and SOA (A. Mulholland et al., *Mashup Corporations: The End of Business As Usual*, Evolved Technologist Press, 2006). Technically unsophisticated business experts would be empowered to model and deploy business models in an extremely quick and efficient fashion. The interconnection of presentation-layer-focused Web applications to internal SOA implementations could be of significant value for enterprises, as this could extend their services' reach to the Web for further use and composition by their business partners and customers.

The firm Kapow Technologies (see <http://www.kapowtech.com>) recently announced the release of a technical solution for these kinds of enterprise mashups. It focuses on empowering users of their Web integration platform to integrate resources available via the Web on several different levels.

First, this helps to seamlessly integrate and expose user-interface content and functionality from arbitrary sources as a new service on the Web. Second, it helps establish application mashups by composing and coupling applications that are accessible via REST or WSDL interfaces into new services.

The key goal of such platforms for creating enterprise mashups is to provide businesses with speed, flexibility, and agility in creating and changing cross-enterprise applications on the basis of a merely visual, complexity-hiding modeling interface. The envisioned result of this approach that combines Web 2.0 and SOA technologies (for example, AJAX, SOAP-based Web services, RSS, and REST) is achieving fast and business requirements-driven content and application integration.

The United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) provides another excellent example of the combination of Web 2.0 and SOA (T. Janner et al., "From EDI to UN/CEFACT: An Evolutionary Path Towards a Next Generation e-Business Framework," *Proc. 5th Int'l Conf. e-Business*, 2006, King Mongkut's Univ. of Technology; <http://www.alexandria.unisg.ch/Publikationen/30346>). It proposes a novel approach for the standardization of business processes. Instead of prescribing yet another fixed standard for the establishment of cross-organizational SOAs, the UN/CEFACT envisions establishing a publicly accessible


repository featuring a basic set of modeling building blocks that can be used, extended, and tagged by the users according to their actual business requirements.

In this work, we thoroughly contrasted the two philosophies of Web 2.0 and SOA from a technical and economic perspective. We identified numerous similarities—but also clear differences—in each of the seven criteria for comparison.

Recent applications show the paramount importance of unifying the two philosophies to drive the next wave of value creation within and across enterprises. Web 2.0 incorporates a social philosophy that we consider complementary to the technology-focused SOA philosophy, as it provides techniques and design principles that strongly facilitate the active consumption of Web-based resources. By integrating the long tail of Web users (Anderson, 2006) into application design via all relevant media channels and on the basis of easily usable platforms that allow for discovering, mashing, and tagging diverse resources, we can realize a comprehensive IoS. ■

Christoph Schroth is a Research Associate and PhD candidate in information management at the University of St. Gallen and Systeme Anwendungen und Produkte in der Datenverarbeitung (SAP) Research in Switzerland. Contact him at christoph.schroth@sap.com.

Till Janner is a Researcher Associate at SAP and is working toward his PhD in information management at the University of St. Gallen. Contact him at till.janner@sap.com.



**The IEEE
Computer
Society**

**publishes over 150 conference
publications a year.**

**For a preview of the latest
papers in your field, visit**

www.computer.org/publications/

Enterprise Mashup

Putting a face on the next generation global SOA



SYSTEMATIC THOUGHT LEADERSHIP FOR INNOVATIVE BUSINESS

Miguel A. Cañas², Juan J. Hierro², Volker Hoyer¹, Till Janner¹,
David Lizcano³, Marcos Reyes², Christoph Schroth¹, Javier Soriano³

¹SAP Research (Switzerland)

²Telefónica I+D (Spain)

³Universidad Politécnica de Madrid, (Spain)

The 8th International Conference on Web Information Systems Engineering
3-7th December, 2007, Nancy, France

WISE
Web Information Systems Engineering

SAP RESEARCH

THE BEST-RUN BUSINESSES RUN SAP™



Time Schedule (14:00-17:00 CET)



Time	Topic
14:00-14:10	Opening, welcome, and introduction
14:10-14:30	General motivation for Enterprise Mashup, related work, and context of the tutorial
14:30-14:50	Enterprise Mashup foundations – terms and concepts
14:50-15:20	Enterprise Mashup foundations – key technologies
15:20-15:30	Overview (demo), and comparison of existing platforms (part 1)
15:30-16:00	Coffee Break (scheduled for all workshops/tutorials)
16:00-16:15	overview (demo), and comparison of existing platforms (part 2)
16:15-16:45	Live-demos of Telefónica I+D prototypes
16:45-17:00	Q&A, discussion, tutorial closing

SAP RESEARCH

© SAP 2007 / Page 2

Executive Summary



The Situation

- SOA and Web Service technologies lack of end-user integration and human-/ service interaction
- Web2.0 technologies provide new concepts to improve this situation
- Enterprise Mashup Platforms and tools are gaining momentum in research and industry

Theory & Practice

- Introducing the theoretical and technical foundation of Enterprise Mashup
- Market overview and comparison of current Mashup Platforms
- Presentation of live-demos of first prototypes in the Enterprise Mashup context

Enterprise Mashup

- Enterprise2.0: combining Web2.0 and Enterprise SOA
- Enterprise Mashups empower end-users, enabling them to create programs and applications that exactly meet their daily needs
- Enterprise Mashup - putting a face on the next generation global SOA

SAP RESEARCH

© SAP 2007 / Page 3

Agenda



1. Opening, welcome, and introduction
2. General Motivation for Enterprise Mashup
 - 2.1. Related work
 - 2.2. Related projects
3. (Enterprise) Mashup foundations
 - 3.1. Terms and concepts
 - 3.2. Key technologies
4. Overview and comparison of current mashup platforms
 - 4.1. Overview and demo of current mashup platforms
 - 4.2. Comparison and shortcomings
5. Live Demos from Telefónica I+D
 - 5.1. Telefónica I+D Seed Project: EzWeb
 - 5.2. MORFEO-EzWeb project
 - 5.3. Building MORFEO-EzWeb gadgets
6. Q&A, discussion and tutorial closing

SAP RESEARCH

© SAP 2007 / Page 4

Speakers introduction



Name	Till Janner
Affiliation	Institute for Media and Communications Management, University of St. Gallen, Switzerland SAP Research CEC St. Gallen, Switzerland
Position	Research Assistant and Doctoral Candidate at the Institute of Media and Communication Management (MCM) Research Associate at the SAP Research CEC St. Gallen
Fields of research	Enterprise Service-oriented Architectures, Enterprise Interoperability, Enterprise 2.0
Short CV	Till Janner holds a diploma-degree on Information Systems ("Diplom-Wirtschaftsinformatiker") of the University of Hamburg, Germany. Before joining University of St. Gallen and SAP Research he gained experience in Enterprise Software Development at industrial projects in the utilities sector.
Website	http://www.alexandria.unisg.ch/persons/Till_Janner



SAP RESEARCH



Contact:

Phone +41 71 224 7713

Mobile +41 76 220 7713

e-mail till.janner@unisg.ch

© SAP 2007 / Page 5

Speakers introduction



Name	David Lizcano
Affiliation	CoNWeT Lab, Department of Computer Science, Universidad Politécnica de Madrid, Spain
Position	Research associate and Ph.D. candidate at the Computer Networks and Web Technologies Lab (CoNWeT Lab), Universidad Politécnica de Madrid
Fields of research	Web 2.0, Enterprise 2.0, Service-oriented Architectures
Short CV	David Lizcano holds a M.Sc. degree with honors in Computer Science (2006) from the Universidad Politécnica de Madrid and holds a research grant from the UPM and the European Social Fund under their Research Personnel Training program. He has gained experience in Web Software Development at several IT European projects
Website	http://hydra.ls.fi.upm.es/research/



CoNWeT Lab
Computer Networks & Web Technologies Laboratory



Contact:

Phone +34 91 336 7394

e-mail dlizcano@fi.upm.es

SAP RESEARCH

© SAP 2007 / Page 6

Speakers introduction



Name	Miguel Angel Cañas Vaz
Affiliation	Telefónica I+D, Spain, Madrid
Position	PhD Student at the Universidad Politécnica de Madrid (UPM) IT Engineer at Telefónica I+T
Fields of research	Enterprise Software Architectures, Enterprise Collaboration, Enterprise 2.0
Short CV	Mr. Miguel Angel Cañas Vaz holds a B.Sc in Computing (2005) at the UPM. Miguel has gained professional experience at Telefónica I+D, getting involved in some R&D projects dealing with Workforce Management, Service-Oriented Architectures and Enterprise Collaboration applied to big enterprises.
Website	www.tid.es



Telefónica I+D

Contact:

Phone +34 91 337 39 09

Mobile +34 626 39 52 83

e-mail mac@tid.es

SAP RESEARCH

© SAP 2007 / Page 7

Agenda



1. Opening, welcome, and introduction
- 2. General Motivation for Enterprise Mashup**
 - 2.1. Related work
 - 2.2. Related projects
3. (Enterprise) Mashup foundations
 - 3.1. Terms and concepts
 - 3.2. Key technologies
4. Overview and comparison of current mashup platforms
 - 4.1. Overview and demo of current mashup platforms
 - 4.2. Comparison and shortcomings
5. Live Demos from Telefónica I+D
 - 5.1. Telefónica I+D Seed Project: EzWeb
 - 5.2. MORFEO-EzWeb project
 - 5.3. Building MORFEO-EzWeb gadgets
6. Q&A, discussion and tutorial closing

SAP RESEARCH

© SAP 2007 / Page 8

Empowering the end-user to overcome several shortcomings of SOA



SOA shortcomings

- As Gartner says in recent paper, the SOA paradigm has several shortcomings due to the complexity of SOAP, WSDL, UDDI, BPEL, and so on.
- The conventional Web Services stack does not suffice to enable a global SOA (interoperability issues, lack of global service registries)
- Users that are not expert programmers cannot afford to create their own SOA-oriented solutions.

Empowerment of the end-user

- Enterprise mashups empower end-users, enabling them to create programs and applications that exactly meet their daily needs
- Mashup supports the DIY (Do it yourself) ideology, putting a user-friendly face on traditional SOA, solving several problems related to user-service interaction
- End-users apply their expertise during the creation of the mashup, fostering innovation and knowledge emergence in enterprises.

SAP RESEARCH

© SAP 2007 / Page 9

Empowering Business Users Trends Today Are Re-Shaping the Workplace

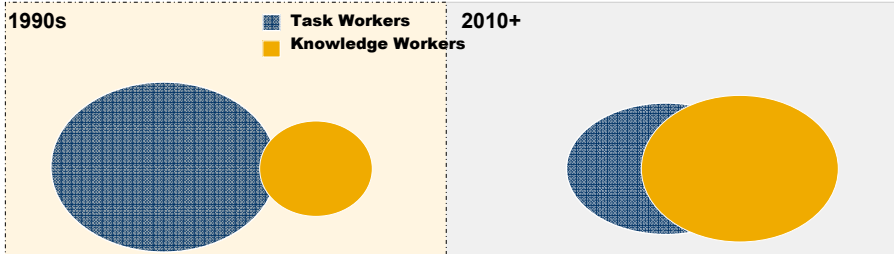


Web 2.0

- Consumer-like expectations
- Internet as lean consumption platform
- Advanced, exception-focused analytics

Nature of Work

- Increasing array of exceptions
- Collaboration across boundaries
- Needs context not information



SAP RESEARCH

* Source: IDC, The Yankee Group, Datamonitor, Unstrung News

© SAP 2007 / Page 10

Related work (part of the tutorial handout)



- **Fostering Innovation in a Mashup-oriented Enterprise 2.0 Collaboration Environment**, Javier Soriano, David Lizcano, Miguel A. Cañas, Marcos Reyes and Juanjo J.Hierro, System and Information Science Notes, Vol 1 Num 1, pp.62 - 69, July 2007. ISSN 1753-2310. SIWN International Conference on Adaptive Business Systems (ICABS'2007) Chengdu, China, 22-24 July 2007
- **Enhancing User-Service Interaction Through a Global User-Centric Approach to SOA**, Juan J. Hierro, Till Janner, David Lizcano, Marcos Reyes, Christoph Schroth, Javier Soriano, The Fourth International Conference on Networking and Services, ICNS'08. Gosier, Guadalupe, 16-21 March 2008. Published by IEEE Computer Society Press.
- Schroth, C., & Janner, T. (2007). **Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services**. IEEE IT Professional (9/3), 36-41.
- Schroth, C., & Christ, O. (2007). **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) (pp. 8): IEEE Computer Society.

SAP RESEARCH

© SAP 2007 / Page 11

Agenda



1. Opening, welcome, and introduction
2. **General Motivation for Enterprise Mashup**
 - 2.1. Related work
 - 2.2. **Related projects**
3. (Enterprise) Mashup foundations
 - 3.1. Terms and concepts
 - 3.2. Key technologies
4. Overview and comparison of current mashup platforms
 - 4.1. Overview and demo of current mashup platforms
 - 4.2. Comparison and shortcomings
5. Live Demos from Telefónica I+D
 - 5.1. Telefónica I+D Seed Project: EzWeb
 - 5.2. MORFEO-EzWeb project
 - 5.3. Building MORFEO-EzWeb gadgets
6. Q&A, discussion and tutorial closing

SAP RESEARCH

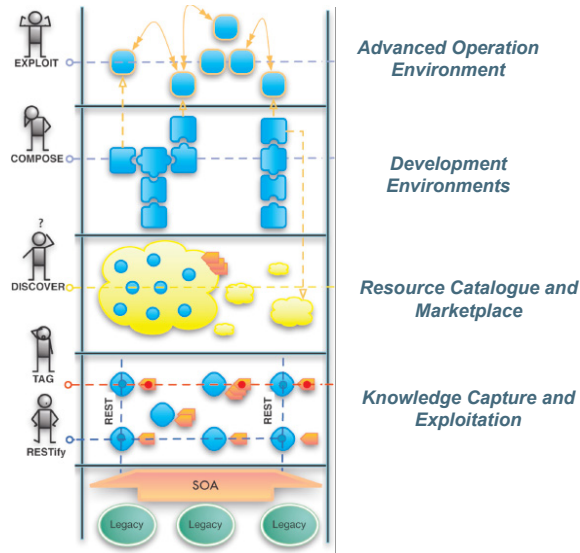
© SAP 2007 / Page 12

EzWeb Project by Telefonica I+D



EzWeb - Putting a Face on SOA

„Target reference architecture and implementation of an open standard platform which manages the front-end access in the next-generation web and supports the economics of a resource marketplace.“



SAP RESEARCH

© SAP 2007 / Page 13

The race for dominating the front-end in the future global SOA



The Internet is about to change...

- **The Internet of Services is about to emerge:** The Web is morphing into a sort of global operating system similar to Microsoft Windows
- **People may leverage resources** available on a global scale and mash them to solutions that meet their individual requirements
- **First examples already exist:** Mash-up platforms (Google, Yahoo pipes), REST-ful interfaces provided by enterprises and more

...EzWeb will act as pace-setter

- **EzWeb will drive the race** for the front-end in a future global SOA
- **Key principles: User-centrism** (user empowerment, ability to share information and context-aware interaction) as well as **Business-Centrism** (creation of a trustworthy marketplace that facilitates sustainable business ecosystems)
- **Private end-users, SMEs and large enterprises** will benefit

Who dominates the front-end access to the web will dominate the future

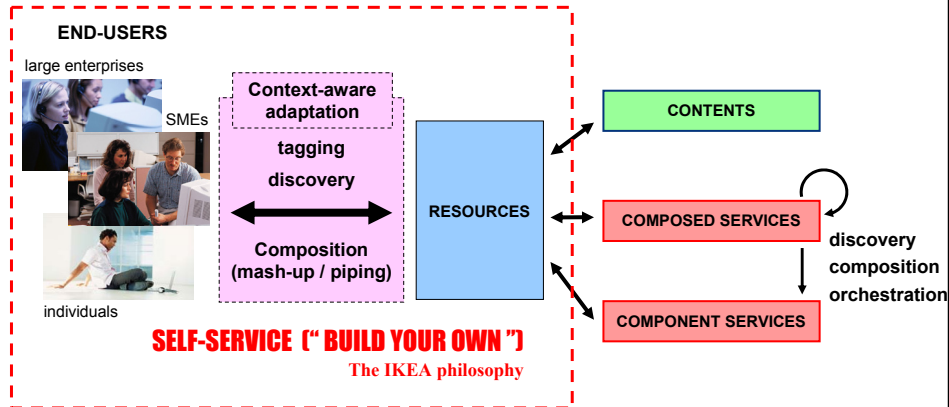
SAP RESEARCH

© SAP 2007 / Page 14

EzWeb: General concept and vision



The EzWeb project aims at providing the **target reference architecture** and reference implementation of an open standard platform which will support definition of the **front-end layer in the next-generation global SOA** (Service Oriented Architecture).



SAP RESEARCH

© SAP 2007 / Page 15

FAST AND ADVANCED STORYBOARD TOOLS

Collaborative Project, Small or medium-scale focused research project (STREP)
7TH FRAMEWORK PROGRAMME – THEME ICT-2007.1.2



TELEFÓNICA I+D, SAP AG, NATIONAL UNIVERSITY OF IRELAND – GALWAY,
UNIVERSITÄT KASSEL, CYNTELIX CORPORATION AND UNIVERSIDAD POLITÉCNICA DE MADRID



SAP RESEARCH

© SAP 2007 / Page 16

FAST Vision



Fast and Advanced Storyboard Tool




„FAST will create a new **visual programming environment** following a **user-centric approach** that 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”

SAP RESEARCH

© SAP 2007 / Page 17


FAST Project Consortium



 **Teléfono Investigación y Desarrollo (Coordinator)**
Telecommunication Provider

 **SAP AG**
World's leading Enterprise Application Provider

 **University of Ireland, Galway**
Leading research institution in Semantic Web

 **University of Kassel**
Strong research experience in scenario-based methodologies

 **Cyntelix Corporation**
Semantic Web technology and collaboration platform

 **Universidad Politécnica de Madrid**
Semantic Web and Web 2.0-based technologies

EU funded Project (FP7)

- March 2008 – February 2011
- Project Budget: 5,6 Mio €
- 6 Partners from academia and industry

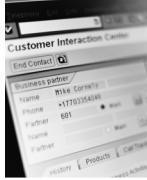
Project Consortium



SAP RESEARCH

© SAP 2007 / Page 18

FAST Objectives



New visual programming environment for the development of complex front-end gadgets

- Involves the modeling and execution of relatively complex business processes
- Relies on back-end Semantic Web Services
- Agile software development from a top-down perspective



Exploitation of the full power of semantics for integrating back-end with front-end (user-centric)

- Developers will start from the front-end gadgets
- Connection to back-end Web Services through business process execution flows if necessary



Plugin Tool for Enterprise Mashup Platforms

- Conceived as a tool that can be compatible with existing and future Enterprise Mashup Platforms
- Tool that enables to develop Mashup-able gadgets which rely on screen-flow Resources and Semantic Web Services stored in a (semantically enriched) catalogue

SAP RESEARCH

© SAP 2007 / Page 19

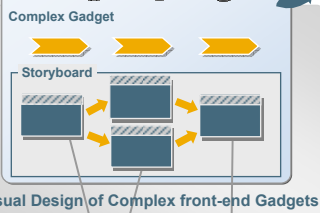
Architecture and Components Overview



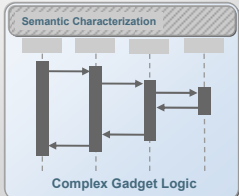
User-Centric Collaboration



Information Worker



Visual Composition of screen-flow Resources and Interoperability with back-end Web Services



Semantic Catalogue of screen-flow Resources and Back-end Web Services

SEMANTIC CATALOGUE

weather create order
calendar email
traffic information delete order
create invoice
hotel booking open invoice

BACK-END



SAP RESEARCH

© SAP 2007 / Page 20

Agenda



1. Opening, welcome, and introduction
2. General Motivation for Enterprise Mashup
 - 2.1. Related work
 - 2.2. Related projects
3. **(Enterprise) Mashup foundations**
 - 3.1. **Terms and concepts**
 - 3.2. Key technologies
4. Overview and comparison of current mashup platforms
 - 4.1. Overview and demo of current mashup platforms
 - 4.2. Comparison and shortcomings
5. Live Demos from Telefónica I+D
 - 5.1. Telefónica I+D Seed Project: EzWeb
 - 5.2. MORFEO-EzWeb project
 - 5.3. Building MORFEO-EzWeb gadgets
6. Q&A, discussion and tutorial closing

SAP RESEARCH

© SAP 2007 / Page 21

Web 2.0 vs. SOA



Web 2.0

“Web 2.0 is defined as the philosophy of mutually maximizing collective intelligence and added value for each participant by formalized and dynamic information sharing and creation.”

vs.

SOA

“... a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with, and use capabilities to produce desired effects consistent with measurable preconditions and expectations.”

SAP RESEARCH

© SAP 2007 / Page 22

Characteristics of Web2.0 (by Tim O'Reilly)



User Focus

- The exploitation of collective intelligence of Web users
- Realization of rich user experiences

Technology and Platform

- Considering the Web a platform for building systems that are "tied together by a set of protocols, open standards, and agreements for cooperation"
- Use of diverse media and devices for the consumption of Internet based applications

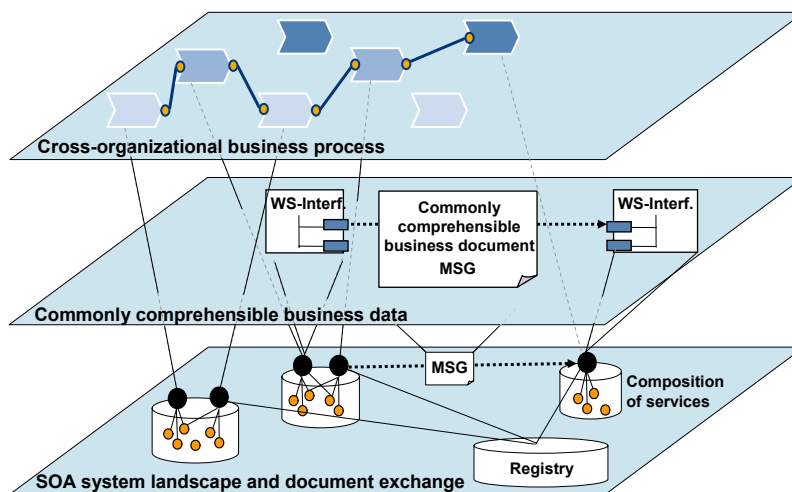
Development Model

- The end of the software release cycle
- The use of lightweight programming models that allow for loosely coupled applications

SAP RESEARCH

© SAP 2007 / Page 23

Cross-organizational Service-Oriented Architecture*



*C. Schroth, T. Janner: Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services, IEEE IT Professional Magazine, Vol. 9, No. 3, 2007

SAP RESEARCH

© SAP 2007 / Page 24

The concept of Mashup and Resources



Mashup ...

- 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.
- A **Mashup Platform** will facilitate flexible, useful, and effective user interaction and management with all kind of **resources**.

Resources

- Comply with **REST** architectural style
- Basically everything on the Web identified by an **URI**
- **Content or service** wrapped by HTTP verbs

Resource categories:

1. **Data sources** (feed the mashup app)
2. **Operators** (transform data sources)
3. **Gadgets** (provide the UI)

SAP RESEARCH

© SAP 2007 / Page 25

From “Traditional” to “Enterprise” Mashup



Traditional Mash-up ...

- **Based on content mash-up**, integrating contents published over the internet
- Users can easily **aggregate** data, contents and isolated functionalities to a web site
- Users can't create powerful applications or solutions only **collecting contents** in a page without “assembling” them

... Enterprise Mash-up

- **Based on Content-driven mashup-oriented programming**, with intercommunication between gadgets
- Users can easily **create** new web applications in a fully visual manner, **composing** solutions from resource orchestration
- Foster knowledge emergency and innovation: users **create applications** that supports their real needs during daily knowledge work

Translating mash-ups into enterprise-class applications is different story

SAP RESEARCH

© SAP 2007 / Page 26

Novel Approaches to Resource Provision



Wide take-up of resource provision

- Large companies such as Google, Amazon, Microsoft, ... start to successfully provide Web-based resources which are leveraged by numerous users

Resource provision in private and corporate context

- Professional services as well as applications that aim at the long tail of private Internet users are emerging

Light-weight, highly standardized protocols and styles

- Light-weight and highly standardized messaging protocols such as RSS, SOAP as well as architectural styles such as REST are quickly gaining momentum on a global scale

SAP RESEARCH

© SAP 2007 / Page 27

Rich Intermediaries Improve Transparency, Navigation and Governance



Transparency

- Intermediaries gather information about published resources over their whole lifecycles and thus improve transparency in a global resource market

Navigation

- Novel intermediaries provide navigation functionality and help user find desired services and content in the global mesh of resources

Governance*

- The reduction of uncertainty plays a relevant role in achieving a reliable, global SOA which is adopted by enterprises to execute mission-critical services over the Web**

* P. Weill and J.W. Ross: IT Governance: How Top Performers Manage IT Decision Rights for Superior Results, Harvard Business School Press, May 2004

** Networked European Software & Services Initiative (NESSI). NESSI Homepage. available online at: <http://www.nessi-europe.com/Nessi/>

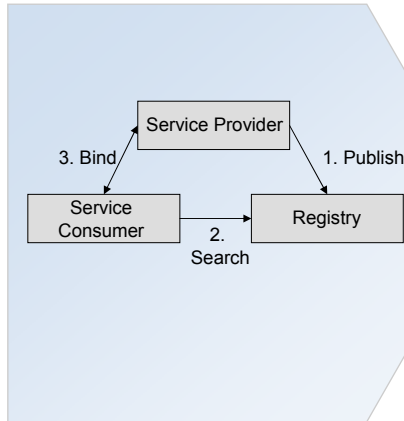
SAP RESEARCH

© SAP 2007 / Page 28

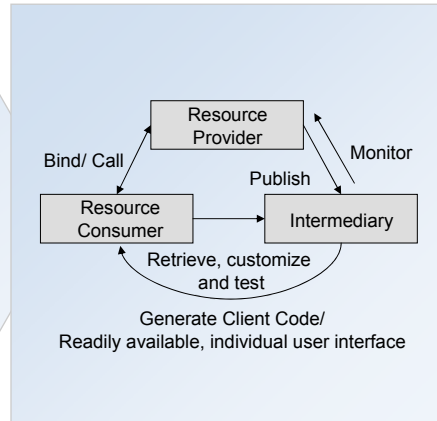
From Conventional Service Registries to Rich Intermediaries



From conventional service registries...



...to rich intermediaries*



*Example: WebRPC.com

SAP RESEARCH

© SAP 2007 / Page 29

New Ways of Resource Consumption enable "User Self-Service"



Google Gadgets

- Aggregation/ Syndication of content/ simple functionality
- Gadget repository
- Template for building own gadgets ("De-facto standard")
- No interaction between the resources (mere syndication)

Yahoo! Pipes

- Pipes can be defined that read in certain RSS-based "feeds"
- Filters can be applied on the pipes (to sort out information that is relevant for the individual user)
- The pipes' outputs can be aggregated and displayed to the user

Kapow Technologies

- Mash-ups can be setup on the basis of a visual modeling interface
- "Stateful" interaction between different resources can be defined and published

SAP RESEARCH

© SAP 2007 / Page 30

The Web Services Stack Does not Suffice to Support a “Global SOA”



Lack of interoperability

- The “corporate household problem”^{**} prevents from seamless interoperability across company boundaries (data structure as well as semantics differ significantly)

Lack of service registries

- Lack of comprehensive, trustworthy and widely accepted service registries prevents global SOAs

„Mute and autistic“

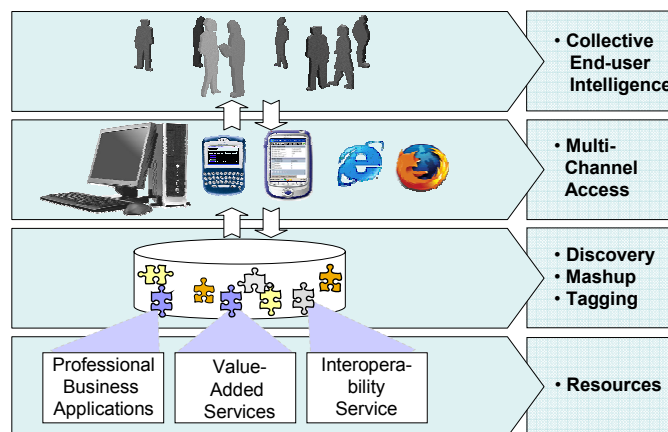
- The Web Services stack aims at supporting the setup of loosely coupled application interconnections especially in a professional context and **assumes users to be technically sophisticated**

* X. Chen, J. Funk, S. Madnick, R. Wang: Corporate Household Data: Research Directions, Proceedings of the Americas Conference on Information Systems, Boston, August 2001

SAP RESEARCH

© SAP 2007 / Page 31

The Internet of Services: Global infrastructure facilitating a networked services economy

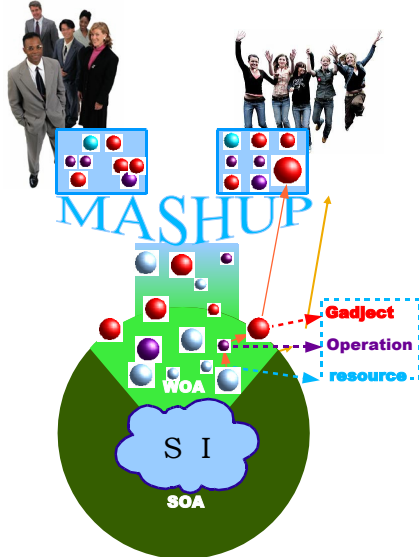


*C. Schroth, T. Janner: Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services, IEEE IT Professional Magazine, Vol. 9, No. 3, 2007

SAP RESEARCH

© SAP 2007 / Page 32

Web Oriented Architecture: WOA



- “Resources” (an URI, exporting HTTP get | post | put | delete operations returning interpretable XML) constitute the set of basic building blocks upon which to create end-user applications
- Web Services will be mainly used in Enterprise Application Integration and will be exposed to the end user through web resources
- Resources may be used and combined in a very simple, straightforward manner, based on preferences and needs from each user or user group

SAP RESEARCH

© SAP 2007 / Page 33

Agenda



1. Opening, welcome, and introduction
2. General Motivation for Enterprise Mashup
 - 2.1. Related work
 - 2.2. Related projects
3. (Enterprise) Mashup foundations
 - 3.1. Terms and concepts
 - 3.2. Key technologies
4. Overview and comparison of current mashup platforms
 - 4.1. Overview and demo of current mashup platforms
 - 4.2. Comparison and shortcomings
5. Live Demos from Telefónica I+D
 - 5.1. Telefónica I+D Seed Project: EzWeb
 - 5.2. MORFEO-EzWeb project
 - 5.3. Building MORFEO-EzWeb gadgets
6. Q&A, discussion and tutorial closing

SAP RESEARCH

© SAP 2007 / Page 34

REST vs SOAP



SOAP

- Service Oriented Architecture is an architectural style based on services, managed/invoked with SOAP
- Services are defined through WSDL, managed with SOAP, discovered with UDDI and orchestrated with BPEL. Each service has its own complex interface
- End-users can't manage these services easily: they need to know exactly how are their interfaces, and then invoke them using SOAP

vs.

REST

- Representational State Transfer architectural style is based on resources
- All modules, services or contents are modeled as resources
- A resource owns an 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 through a uniform interface that end-users can "see and touch"
- Users are empowered by allowing them to choose the best resources to meet their needs

SAP RESEARCH

© SAP 2007 / Page 35

REST Code example



Request:

GET www.parts-depot.com/parts?flavor=xml

Answer:

```
<?xml version="1.0"?>
<p:Parts xmlns:p="http://www.parts-depot.com"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=
"http://www.parts-depot.com
http://www.parts-depot.com/parts.xsd">
<Part id="00345" xlink:href="http://www.parts-depot.com/parts/00345"/>
<Part id="00346" xlink:href="http://www.parts-depot.com/parts/00346"/>
<Part id="00347" xlink:href="http://www.parts-depot.com/parts/00347"/>
<Part id="00348" xlink:href="http://www.parts-depot.com/parts/00348"/>
</p:Parts>
```

SAP RESEARCH

© SAP 2007 / Page 36

SOAP Code example



Request:

POST <http://www.parts-depot.com/soap/servlet/messagerouter>

with xml parameters:

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<p:getPartsList xmlns:p="http://www.parts-depot.com"/>
</soap:Body>
</soap:Envelope>
```

Answer

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<p:getPartsListResponse xmlns:p="http://www.parts-depot.com">
<Parts>
<Part-ID>00345<Part-ID>
<Part-ID>00346<Part-ID>
<Part-ID>00347<Part-ID>
<Part-ID>00348<Part-ID>
</Parts>
<p:getPartsListResponse>
</soap:Body>
</soap:Envelope>
```

SAP RESEARCH

© SAP 2007 / Page 37

Information feeds



- Information feeds are a family of Web formats used to publish frequently updated content
- Information feeds make it possible for people to keep up with their favorite web sites in an automated manner that's easier than checking them manually.
- Content can be read using software called a "feed reader" or an "aggregator." The user subscribes to a feed by entering the feed's link into the reader. The reader checks the user's subscribed feeds regularly for new content, downloading any updates that it finds
- JSON (JavaScript Object Notation) is a lightweight computer data interchange format. It is a text-based, human-readable format for representing simple data
- Important information feeds: RSS, Atom, JSON...



SAP RESEARCH

© SAP 2007 / Page 38

RSS Feed Code example



```
<?xml version="1.0"?>
<rss version="2.0">
  <channel>
    <title>Liftoff News</title>
    <link>http://liftoff.msfc.nasa.gov</link>
    <description>Liftoff to Space Exploration.</description>
    <language>en-us</language>
    <pubDate>Tue, 10 Jun 2003 04:00:00 GMT</pubDate>
    <lastBuildDate>Tue, 10 Jun 2003 09:41:01 GMT</lastBuildDate>
    <docs>http://blogs.law.harvard.edu/tech/rss</docs>
    <generator>Weblog Editor 2.0</generator>
    <managingEditor>editor@example.com</managingEditor>
    <webMaster>webmaster@example.com</webMaster>
    <item>
      <title>Star City</title>
      <link>http://liftoff.msfc.nasa.gov/news/2003/news-starcity.asp</link>
      <description>How do Americans get ready to work with Russians aboard the
      International Space Station? They take a crash course in culture, language and
      protocol at Russia's Star City.</description> <pubDate>Tue, 03 Jun 2003 09:39:21
      GMT</pubDate> <guid>http://liftoff.msfc.nasa.gov/2003/06/03.html#item573</guid>
    </item>
    <item>
      <title>Space Exploration</title>
      <link>http://liftoff.msfc.nasa.gov</link>
      <description>Sky watchers in Europe, Asia, and parts of Alaska and Canada will
      experience a partial eclipse of the Sun on Saturday, May 31st.</description>
      <pubDate>Fri, 30 May 2003 11:06:42 GMT</pubDate>
      <guid>http://liftoff.msfc.nasa.gov/2003/05/30.html#item572</guid>
    </item>
  </channel>
</rss>
```

SAP RESEARCH

© SAP 2007 / Page 39

JSON string Code example (EzWeb gadget representation)



```
{
  "vendor": "Morfeo",
  "description": "This gadget presents information about communication systems
failures",
  "name": "Bandeja_de_Averias",
  "tags": [],
  "image": "http://europa.ls.fi.upm.es/ezweb/gadgets/RedLightBulb.gif",
  "uri": "/user/admin/gadgets/Morfeo/Bandeja_de_Averias/1.0",
  "version": "1.0",
  "user": "userLogin",
  "template": {
    "variables": [
      { "type": "S", "name": "phoneNumber", "aspect": "EVEN" },
      { "type": "S", "name": "serviceHired", "aspect": "EVEN" }
    ],
    "size": { "width": 3, "height": 11 }
  },
  "mail": "miglesia@pegaso.ls.fi.upm.es",
  "shared": false,
  "xhtml": { "uri": "/user/admin/gadgets/Morfeo/Bandeja_de_Averias/1.0/xhtml1" }
}
```

SAP RESEARCH

© SAP 2007 / Page 40

Resource composition: Piping and wiring



- wiring composition
 - intercommunicate gadgets
 - this resolves their interdependences to achieve an interconnected graphical mash-up solution to complex functional requirements as a traditional desktop application

- piping composition
 - interconnect a number of heterogeneous data sources to create gadgets
 - this generates composed processing data chains/graphs by concatenating successive operators on data sources

- Data Marshalling and parameterization
 - to improve reusability and thus foster collaboration, both piping and wiring composition activities support marshalling (e.g. XSLT) and parameterization

SAP RESEARCH

© SAP 2007 / Page 41

Gadgets or Widgets



- Is a portable chunk of code that can be installed and executed (commonly within any separate HTML-based web page) by an end user without requiring additional compilation.
- They are derived from the idea of reusable code that has existed for years.
- Gadgets represent the basic building blocks 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 applications in a very short time

- Other terms: badge, module, capsule, snippet, mini and flake

SAP RESEARCH

© SAP 2007 / Page 42

Gadget Code example (a Gadget template)



```
<?xml version="1.0" encoding="UTF-8" ?>
<Template schemaLocation="http://morfeo-project.org/2007/Template">
  <!-- Meta tags define gadgets properties -->
  <Catalog.ResourceDescription>
    <Vendor>Morfeo</Vendor>
    <Name>Configuracion de Red</Name>
    <Version>1.0</Version>
    <Author>miglesia</Author>
    <Mail>miglesia@pegaso.ls.fi.upm.es</Mail>
    <Description>this gadget...</Description>
    <ImageURI>http://europa.ls.fi.upm.es/ezweb/gadgets/red.jpg</ImageURI>
    <WikiURI>http://www.image.com</WikiURI>
  </Catalog.ResourceDescription>
  <Platform.Wiring>
    <Wire name="serviceHired" type="text" label="ServiceHired"
      friendcode="service" wiring="out" />
    <Wire name="deviceId" type="text" label="deviceId"
      friendcode="deviceId" wiring="in" />
    <Wire name="deviceStatus" type="text" label="deviceStatus"
      friendcode="deviceStatus" wiring="in" />
  </Platform.Wiring>
  <Platform.Link>
    <XHTML
      href="http://europa.ls.fi.upm.es/ezweb/gadgets/configuracionRed.html" />
  </Platform.Link>
  <Platform.Rendering width="1" height="3" />
</Template>
```

SAP RESEARCH

© SAP 2007 / Page 43

Agenda



1. Opening, welcome, and introduction
2. General Motivation for Enterprise Mashup
 - 2.1. Related work
 - 2.2. Related projects
3. (Enterprise) Mashup foundations
 - 3.1. Terms and concepts
 - 3.2. Key technologies
4. **Overview and comparison of current mashup platforms**
 - 4.1. Overview and demo of current mashup platforms
 - 4.2. Comparison and shortcomings
5. Live Demos from Telefónica I+D
 - 5.1. Telefónica I+D Seed Project: EzWeb
 - 5.2. MORFEO-EzWeb project
 - 5.3. Building MORFEO-EzWeb gadgets
6. Q&A, discussion and tutorial closing

SAP RESEARCH

© SAP 2007 / Page 44

iGoogle



iGoogle is a mash-up platform that aims to centralize all personal information in a portal (or web page)

iGoogle is based on:

- Free wheel composition of gadgets in a web page, considered as a personal desktop from where a web users' experience starts every day
- Users can choose gadgets from:
 - A catalogue of ready-made components
 - Gadgets created by end-users

SAP RESEARCH

© SAP 2007 / Page 45

iGoogle - Demo



iGoogle

Google Search | I'm Feeling Lucky

Advanced Search
Enterprise
Language Tools

Hangman 2.0 Beta has been removed. Undo | Close

Home | Web 2.0 | Add a tab

Select theme | Add stuff »

Web 2.0 Technology Feeds

TechCrunch | TechMeme | ScoopWizer | GlobalTechnology

TechCrunch
TechCrunch profiles the companies, products and events that are defining and transforming the new web. TechCrunch is written by Michael Arrington.

TripIt Brings a Fresh New Look at Video Guides for Hotels

TV trip | TripIt is offering a multimedia hotel guide with videos professi
[link 14 nov 2007 15:17:52 CET]

Inbox 2.0 Makes Me Sad

Saul Hansel at The New York Times is

[link 14 nov 2007 09:24:39 CET]

VibeAgent Steps Out of Beta, Wants To Defeat TripAdvisor With People Power

Later today | VibeAgent

[link 14 nov 2007 09:00:24 CET]

Ads by Google | Tech RSS Feeds | RSS XML | Blogger RSS

Colombi Spring Feeders

Spring Detangling Feeder SemiAuto. Feeds up to 5 Stations / Operators
www.colombi.com

PeefFinance

The professional and secure way for your payments to Switzerland.
www.peefinance.ch

SAP RESEARCH

© SAP 2007 / Page 46

IGoogle - Advantages



- Catalogue has a lot of existing gadgets to choose from
- You have the option of creating new gadgets to use in the mashup
- IGoogle users can accumulate a lot of useful information on their home page
- End-users can fix the mashup, replacing/resizing gadgets on the page
- Users can create complex applications, located on their home page

SAP RESEARCH

© SAP 2007 / Page 47

IGoogle - Disadvantages



- End-users will find it difficult to create complex mashups, because:
 - Heavy programmatic skills are required
 - There is a poor variety of data inputs
 - There is high coupling between mashups and the platform (a middleware gadget is required to allow gadgets to communicate)
- Created mash-ups are not compatible with any other platform
- There is no inherent communication between gadgets

SAP RESEARCH

© SAP 2007 / Page 48

IGoogle - Links



- User's manual
 - <http://www.google.es/support/?ctx=web>
- Gadgets Development Api
 - http://code.google.com/apis/gadgets/docs/dev_guide.html
- Gadgets Development IDE
 - <http://desktop.google.com/dev/designer.html>
- YouTube Video of the platform
 - <http://www.youtube.com/watch?v=JorJOBFBKWM>

SAP RESEARCH

© SAP 2007 / Page 49

Yahoo Pipes



Visual development environment for generating data-oriented mashups

- Development is based on:
 - Dragging gadgets (pipes) from a toolbox and dropping them in work zone
 - Specifying data input
 - Interconnecting gadgets through pipes
 - Specifying data output format

SAP RESEARCH

© SAP 2007 / Page 50

Yahoo Pipes - Demo



SAP RESEARCH

© SAP 2007 / Page 51

Yahoo Pipes - Demo



SAP RESEARCH

© SAP 2007 / Page 52

Yahoo Pipes - Advantages



- Entirely graphic and intuitive IDE; not one line of code is required
- Inherent internal communication between gadgets
- RSS, CSV and JSON support
- Portability of mashups to other platforms (i.e. QedWiki)

SAP RESEARCH

© SAP 2007 / Page 53

Yahoo Pipes - Disadvantages



- Yahoo pipes does not generate XML from the mashup (limiting hacking and remixing)
- Data inputs are limited to a prefixed set
- It is very difficult to create complex mashups with really powerful functionalities early on
- Other platforms are usually required to create enterprise mashups

SAP RESEARCH

© SAP 2007 / Page 54

Yahoo Pipes - Links



- Yahoo Pipes Blog
 - <http://blog.pipes.yahoo.com/>
- Video tutorial
 - <http://www.jumpcut.com/fullscreen?id=594F555C568011DC9D24000423CEF5B0&type=movie>
- Forum
 - <http://discuss.pipes.yahoo.com/>
- Catalogue
 - <http://pipes.yahoo.com/pipes/pipes.popular>
- Tutorials
 - <http://pipes.yahoo.com/pipes/docs?doc=tutorials>
- FeedBack and suggestions
 - <http://suggestions.yahoo.com/?prop=Pipes>

SAP RESEARCH

© SAP 2007 / Page 55

OpenKapow

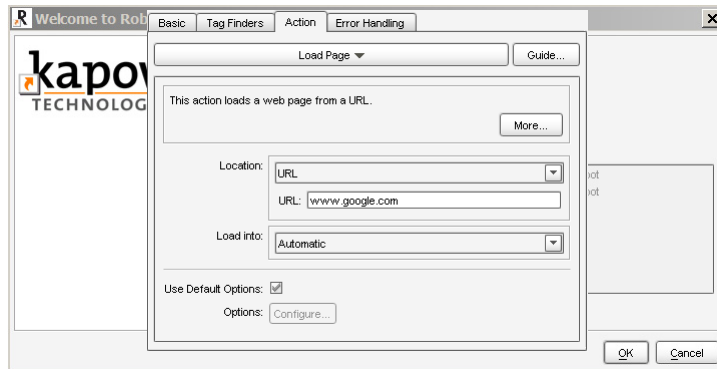


- Mashup platform for creating and remixing gadgets called “robots”. Robots have the following features:
 - Each robot is like a little components mashup
 - Robot inputs:
 - XML
 - HTML through screen scraping
 - Robots can manage a wide variety of data outputs
 - Robots can communicate with each other
 - These robots are created and managed with a desktop IDE, RoboMaker

SAP RESEARCH

© SAP 2007 / Page 56

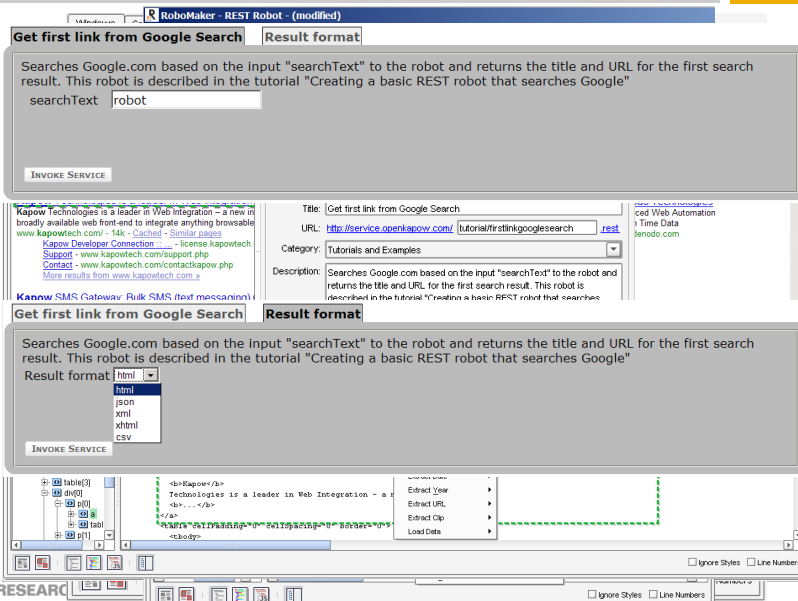
OpenKapow - Demo



SAP RESEARCH

© SAP 2007 / Page 57

OpenKapow - Demo



SAP RESEARCH

© SAP 2007 / Page 58

OpenKapow - Advantages



- This platform has a complete catalogue of prefabricated resources
- There is inherent communication between gadgets
- RoboMaker is a powerful and visual IDE
- It has a wide range of data inputs (XML, HTML, ...)
- It has a wide range of data outputs (XML, HTML, CSV, JSON, XHTML)
- It includes several ways to interact with end-users in final mashups (forms, REST calls, ...)
- Platform users can have their own robots server
- There is a lot of information, user guides, tutorials and support

SAP RESEARCH

© SAP 2007 / Page 59

OpenKapow - Disadvantages



- The platform is focused entirely on development, instead of exploitation
- You have to download an additional desktop IDE
- This platform targets heavy-tech users, with previous programming skills
- Robots can't be easily hacked or remixed
- The OpenKapow mashup server is not free
- The platform is overly centered on HTML and screen scraping

SAP RESEARCH

© SAP 2007 / Page 60

OpenKapow - Links



- Catalogue
 - <http://openkapow.com/files/robothome.aspx>
- Demos
 - <http://openkapow.com/blogs/demos/default.aspx>
- Tutorials
 - <http://openkapow.com/blogs/learn.aspx>
- Forums
 - <http://openkapow.com/forums/default.aspx8>
- Blog
 - <http://openkapow.com/blogs/news/default.aspx>

SAP RESEARCH

© SAP 2007 / Page 61

IBM QedWiki



- Mashup platform that combines two perspectives (data-oriented mashups and presentation mashups)
- Features:
 - Data inputs:
 - Databases
 - Feed RSS
 - Yahoo Pipes mashups
 - Spread sheets
 - XML
 - ...
 - Mashup process based on drag & drop operations, using gadgets from a toolbox, or further searching from a wiki of resources
 - Internal communication between gadgets is supported
 - Mashup results can be entered in a wiki, tagged, and refined by other users

SAP RESEARCH

© SAP 2007 / Page 62

IBM QedWiki - Demo



The screenshot shows a web-based workspace with a sidebar on the left containing a file tree with items like 'DemoOne', 'DemoTree', 'Test', 'Welcome', 'Wiki', and 'WikiPages'. The main area displays a news feed from 'scotsman.com' dated Monday, 11th December 2006. The feed includes a 'Show Data' section with a table of news items and a 'Feed Content Viewer' showing the full text of the selected article: 'EU agrees partial freeze of Turkey talks - Scotsman'. A red dashed arrow points from the 'EU agrees partial freeze of Turkey talks' entry in the 'Show Data' table to the corresponding article in the 'Feed Content Viewer'.

SAP RESEARCH

© SAP 2007 / Page 63

IBM QedWiki - Advantages



- QedWiki combines several tendencies; it supports data mashups, like Yahoo Pipes, and creates complex presentation mashups, like OpenKapow
- It supports internal communication between gadgets
- It supports a wide range of data inputs, managing databases directly
- Web-based IDE, no additional software has to be downloaded
- It can manage the internal code of each gadget and even of each complete mashup (according to its own XML-based standards)
- Simple documentation support oriented to inexperienced users

SAP RESEARCH

© SAP 2007 / Page 64

IBM QedWiki - Disadvantages



- You need to know code standards and the gadget programming language to build complex solutions because of limitations in the supplied visual IDE
- Screen scraping is not supported
- There is little, poorly detailed and slowly updated documentation. The QedWiki Project is not hot today.

SAP RESEARCH

© SAP 2007 / Page 65

IBM QedWiki - Links



- QEDWiki Help
 - <http://services.alphaworks.ibm.com/qedwiki/learn.html>
- Tutorial
 - <http://www.youtube.com/watch?v=63qlq9t9Gqs>
- Blog
 - http://www.mashboards.net/index.php?option=com_weblinks&catid=27&Itemid=23
- Forums
 - <http://services.alphaworks.ibm.com/qedwiki/forum.html>
- FAQ
 - <http://services.alphaworks.ibm.com/qedwiki/faq.html>

SAP RESEARCH

© SAP 2007 / Page 66

Agenda



1. Opening, welcome, and introduction
2. General Motivation for Enterprise Mashup
 - 2.1. Related work
 - 2.2. Related projects
3. (Enterprise) Mashup foundations
 - 3.1. Terms and concepts
 - 3.2. Key technologies
- 4. Overview and comparison of current mashup platforms**
 - 4.1. Overview and demo of current mashup platforms
 - 4.2. Comparison and shortcomings**
5. Live Demos from Telefónica I+D
 - 5.1. Telefónica I+D Seed Project: EzWeb
 - 5.2. MORFEO-EzWeb project
 - 5.3. Building MORFEO-EzWeb gadgets
6. Q&A, discussion and tutorial closing

SAP RESEARCH

© SAP 2007 / Page 67

Platforms Comparison – General features



Features	iGoogle	pipes	openkapow	QEDWiki
Wiki-based mashup catalogue	✗	✗	✗	✓
Free space to publish mashups	✓	✓	✗	✓
Visual composition (drag & drop)	✓	✓	✗	✓
Gadgets IDE	✗	✗	✓	✓
Internal communication	✗	✓	✓	✓
Mashups reuse between platforms enabled	✗	✓	✓	✓
REST support	✗	✗	✓	only REST inputs
Creation of Web-services based on mashups	✗	✗	✓	✗

© SAP 2007 / Page 68

SAP RESEARCH

© SAP 2007 / Page 68

Platforms Comparison – Supported inputs



Inputs	iGoogle	pipes ^{beta}	openkapow ^{beta}	QEDWiki
RSS	✓	✓	✓	✓
JSON	✓	✓	✓	✓
XML	✓	✓	✓	✓
HTML	✓	✓	✓	✓
Screen Scraping	✗	✗	✓	✗
CSV	✗	✓	✗	✓
Spread Sheets	✗	✗	✗	✓

© SAP 2007 / Page 69

SAP RESEARCH

© SAP 2007 / Page 69

Platforms Comparison – Supported outputs



Outputs	iGoogle	pipes ^{beta}	openkapow ^{beta}	QEDWiki
RSS	✗	✓	✓	✓
JSON	✗	✗	✓	✓
XML	✓	✓	✓	✓
HTML	✓	✗	✓	✗
CSV	✗	✓	✓	✓
XHTML	✗	✗	✓	✗
Spread Sheets	✗	✗	✗	✓

© SAP 2007 / Page 70

SAP RESEARCH

© SAP 2007 / Page 70

Platforms Comparison – Documentation



Documentation	iGoogle	pipes ^{beta}	openkapow ^{beta}	QEDWiki
Help, user guide, tutorials, demo	✓	✓	✓	✓
Blog / Forums	✗	✓	✓	✓
FAQ	✗	✗	✓	✗
User feedback	✗	✓	✗	✗
Detailed mashup documentation & support	✓	✗	✗	✗

© SAP 2007 / Page 71

SAP RESEARCH

© SAP 2007 / Page 71

Platforms Comparison – IDEs



IDE features	pipes ^{beta}	openkapow ^{beta}	QEDWiki
IDE as a service	✓	✗	✓
No previous configuration required	✓	✗	✗
Completely visual development	✓	✗	✓
Given items can be modified	✗	✓	✓
Low-level programming supported	✗	✓	✓
Required Programming level	LOW	HIGH	MEDIUM/HIGH

© SAP 2007 / Page 72

SAP RESEARCH

© SAP 2007 / Page 72

A Feature List For An “Ideal” Enterprise Mashup Platform



In general, all the platforms have several weak points and major shortcomings. An ideal enterprise mashup platform should support all the features below:

- A powerful gadget catalogue, based on an “architecture of participation” (i.e. taggable wiki) users add value
- Free mashup composition, remix and management
- A wide range of inputs/outputs (XML, HTML, DDBB, Spread Sheets, ...)
- Mashup creation as REST-oriented Web Services
- Internal communication between gadgets to create real complex mashups
- Mashups should follow a shared standard to enable compatibility between platforms
- Easy-to-use Web and visual IDE with low-level programming support and a high level of abstraction
- Detailed documentation and user guides:
 - Blogs and forums
 - Feedback
 - FAQ

SAP RESEARCH

© SAP 2007 / Page 73

Agenda



1. Opening, welcome, and introduction
2. General Motivation for Enterprise Mashup
 - 2.1. Related work
 - 2.2. Related projects
3. (Enterprise) Mashup foundations
 - 3.1. Terms and concepts
 - 3.2. Key technologies
4. Overview and comparison of current mashup platforms
 - 4.1. Overview and demo of current mashup platforms
 - 4.2. Comparison and shortcomings
5. **Live Demos from Telefónica I+D**
 - 5.1. Telefónica I+D Seed Project: EzWeb
 - 5.2. MORFEO-EzWeb project
 - 5.3. Building MORFEO-EzWeb gadgets
6. Q&A, discussion and tutorial closing

SAP RESEARCH

© SAP 2007 / Page 74

Telefónica I+D Seed Project: EzWeb



- EzWeb was developed under the scope of a internal innovation process at Telefónica I+D
- It was a first approach in order to extrapolate Mashup and Web2.0 technologies to an enterprise environment
- Simple objective:
 - Validate available Mashup tools in the scope of enterprise needs
 - Validate utility of Mashup related technologies from an enterprise perspective
- At Q4 2006, there were not many Mashup Platforms
 - IGoogle was selected as the target platform
- Structured on two phases:
 - Enterprise Services Wrap-up by REST resources
 - Gadget construction
- Visually Mashing-up the developed gadgets, a complete powerful operational environment can be easily built.
- The operational environment manages typical activities for solving Telco failures:
 - Presenting all kind of user and network info
 - Evolving incidences on an enterprise workflow system
 - Testing Telco devices

SAP RESEARCH

© SAP 2007 / Page 75

MORFEO-EzWeb project



- MORFEO (<http://www.morfeo-project.org>) is a free source community founded by Telefónica I+D and other partners
- MORFEO-EzWeb is a innovation project funded by the Spanish Administration
- The whole project results will be published in the MORFEO community in order to promote its public access
- The project is still under construction but an initial phase release is currently available
- New features (of the initial phase release):
 - Template concept
 - A flexible intergadget communication solution (wiring)
 - A simplified API
 - Advanced rendering capabilities
 - Social software tools integration (WIKI)

SAP RESEARCH

© SAP 2007 / Page 76

Building MORFEO-EzWeb gadgets



- Gadget construction is a three-phase process:
 - Declarative definition of gadget template
 - Writing gadget web interface
 - Writing communication source code via MORFEO-EzWeb API
- Once a set of gadgets are loaded into platform, user can customize their connections freely by the Wiring Manager with no IT expertise

SAP RESEARCH

© SAP 2007 / Page 77

Building MORFEO-EzWeb gadgets Gadget Template



```
<?xml version="1.0" encoding="UTF-8" ?>
<Template schemaLocation="http://morfeo-project.org/2007/Template">
  <!-- Meta tags define gadgets properties -->
  <Catalog:ResourceDescription>
    <Vendor>Morfeo</Vendor>
    <Name>Configuracion_de_Red</Name>
    <Version>1.0</Version>
    <Author>miglesia</Author>
    <Mail>miglesia@pegaso.ls.fi.upm.es</Mail>
    <Description>this gadget...</Description>
    <ImageURI>http://europa.ls.fi.upm.es/ezweb/gadgets/red.jpg</ImageURI>
    <WikiURI>http://www.image.com</WikiURI>
  </Catalog:ResourceDescription>
  <!-- EzWeb Gadget Data Wiring -->
  <Platform:Wiring>
    <Wire name="serviceHired" type="text" friendcode="service" wiring="out" />
    <Wire name="deviceId" type="text" label="deviceId" friendcode="deviceId" wiring="in" />
  </Platform:Wiring>
  <Platform:Link>
    <XHTML href="http://europa.ls.fi.upm.es/ezweb/gadgets/configuracionRed.html" />
  </Platform:Link>
  <Platform:Rendering width="1" height="3" />
</Template>
```

SAP RESEARCH

© SAP 2007 / Page 78

Building MORFEO-EzWeb gadgets Gadget API



Wiring API:

```
# RW variable definition
EzWebAPI.createRWGadgetVariable = function(varName);
# R variable definition
EzWebAPI.createRGadgetVariable = function(varName, callbackFunction);
# Set a variable value
EzWebAPI.RWGadgetVariable.set = function(value);
```

Piping-enabler API:

```
#Traditional AJAX-based REST verbs
EzWebAPI.sendGet = function (url, context, successHandler, errorHandler);
EzWebAPI.sendPost = function (url, params, context,
                              successHandler, errorHandler);
EzWebAPI.sendPut = function ( url, params, context,
                              successHandler, errorHandler);
EzWebAPI.sendDelete = function (url, params, context,
                                 successHandler, errorHandler);
```

SAP RESEARCH

© SAP 2007 / Page 79

Building MORFEO-EzWeb gadgets Gadget Source Code (1/2)



```
<html>
...
<script language="javascript" src="/ezweb/js/EzWebAPI/EzWebAPI.js"></script>
...
<script>
var address = new EzWebAPI.createRGadgetVariable("address",_addressHandler);

function _initialize() {
  if (GBrowserIsCompatible()) {
    map = new GMap2(document.getElementById("map_canvas"));
    map.addControl(new GMapTypeControl());
    map.addControl(new GLargeMapControl());
    map.setCenter(new GLatLng(34, 0), 1);
    geocoder = new GClientGeocoder();
  }
}

function _addressHandler(address) {
  if (address == null || address == undefined) {
    map.setCenter(new GLatLng(34, 0), 1);
    return;
  }
}
</script>

<body style="padding: 0px; margin: 0px" onload="_initialize;">
<div id="map_canvas" style="width:500px; height: 400px;"></div>
</body>
</html>
```

SAP RESEARCH

© SAP 2007 / Page 80

Building MORFEO-EzWeb gadgets Gadget Source Code (2/2)



```
<html>
...
<script language="javascript" src="/ezweb/js/EzWebAPI/EzWebAPI.js"></script>
...
<script>
var phoneNumber = EzWebAPI.createRGadgetVariable("phoneNumber",_phoneHandler);
var address = EzWebAPI.createRWGadgetVariable("address");

function _phoneHandler(phone){
if (phone==null && phone!=undefined){
var content="";
content+="
```

SAP RESEARCH

© SAP 2007 / Page 81

Agenda



1. Opening, welcome, and introduction
2. General Motivation for Enterprise Mashup
 - 2.1. Related work
 - 2.2. Related projects
3. (Enterprise) Mashup foundations
 - 3.1. Terms and concepts
 - 3.2. Key technologies
4. Overview and comparison of current mashup platforms
 - 4.1. Overview and demo of current mashup platforms
 - 4.2. Comparison and shortcomings
5. Live Demos from Telefónica I+D
 - 5.1. Telefónica I+D Seed Project: EzWeb
 - 5.2. MORFEO-EzWeb project
 - 5.3. Building MORFEO-EzWeb gadgets
- 6. Q&A, discussion and tutorial closing**

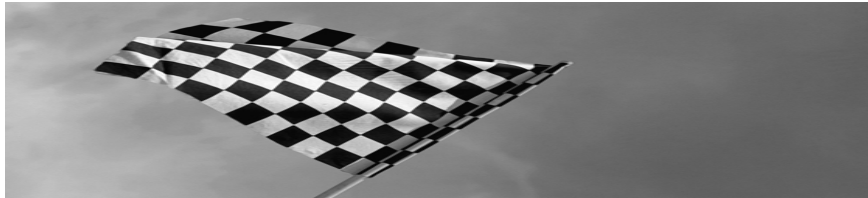
SAP RESEARCH

© SAP 2007 / Page 82

Summary and Outlook

Enterprise Mashup

Putting a face on the next generation global SOA



- Enterprise Mashups combine Enterprise-level services and Web2.0 technologies to **foster end-user innovation and knowledge emergence** in enterprises.
- **FAST**: Visual programming environment following a user-centric approach that will facilitate the development of complex front-end gadgets. **EzWeb**: reference architecture and implementation of an open standard platform which manages the front-end access in the next-generation web.
- Enterprise Mashup will a critical part of the **front-end layer in the next-generation global SOA**.

SAP RESEARCH

© SAP 2007 / Page 83

Q&A, discussion

Thank you!

Till Janner

SAP RESEARCH

SAP (Schweiz) AG
Blumenbergplatz 9
9000 St. Gallen
Switzerland

T +41/71/224-7713
M +41/76/220-7713

till.janner@sap.com
www.sap.com

David Lizcano



Universidad Politécnica de Madrid
Boadilla del Monte
28660 Madrid
Spain

T +34/91/336-7394

dizcano@fi.upm.es
www.fi.upm.es

Miguel Angel Cañas



Telefónica i+D
Emilio Vargas, 6
28043 Madrid
Spain

T +34/91/337-3909
M +34/626/39 52 83

mac@tid.es
www.tid.es

THE BEST-RUN BUSINESSES RUN SAP™



Copyright 2007 SAP AG All rights reserved



No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, Duet, Business ByDesign, ByDesign, PartnerEdge and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned and associated logos displayed are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

The information in this document is proprietary to SAP. This document is a preliminary version and not subject to your license agreement or any other agreement with SAP. This document contains only intended strategies, developments, and functionalities of the SAP® product and is not intended to be binding upon SAP to any particular course of business, product strategy, and/or development. SAP assumes no responsibility for errors or omissions in this document. SAP does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within this material. This document is provided without a warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. This limitation shall not apply in cases of intent or gross negligence.

The statutory liability for personal injury and defective products is not affected. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third-party Web pages nor provide any warranty whatsoever relating to third-party Web pages

Weitergabe und Vervielfältigung dieser Publikation oder von Teilen daraus sind, zu welchem Zweck und in welcher Form auch immer, ohne die ausdrückliche schriftliche Genehmigung durch SAP AG nicht gestattet. In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden.

Einige von der SAP AG und deren Vertriebspartnern vertriebene Softwareprodukte können Softwarekomponenten umfassen, die Eigentum anderer Softwarehersteller sind.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, Duet, Business ByDesign, ByDesign, PartnerEdge und andere in diesem Dokument erwähnte SAP-Produkte und Services sowie die dazugehörigen Logos sind Marken oder eingetragene Marken der SAP AG in Deutschland und in mehreren anderen Ländern weltweit. Alle anderen in diesem Dokument erwähnten Namen von Produkten und Services sowie die damit verbundenen Firmenlogos sind Marken der jeweiligen Unternehmen. Die Angaben im Text sind unverbindlich und dienen lediglich zu Informationszwecken. Produkte können länderspezifische Unterschiede aufweisen.

Die in diesem Dokument enthaltenen Informationen sind Eigentum von SAP. Dieses Dokument ist eine Vorabversion und unterliegt nicht Ihrer Lizenzvereinbarung oder einer anderen Vereinbarung mit SAP. Dieses Dokument enthält nur vorgesehene Strategien, Entwicklungen und Funktionen des SAP®-Produkts und ist für SAP nicht bindend. Einem bestimmten Geschäftsweg, eine Produktstrategie bzw. -entwicklung einzuschlagen, SAP übernimmt keine Verantwortung für Fehler oder Auslassungen in diesen Materialien. SAP garantiert nicht die Richtigkeit oder Vollständigkeit der Informationen, Texte, Grafiken, Links oder anderer in diesen Materialien enthaltenen Elemente. Diese Publikation wird ohne jegliche Gewähr, weder ausdrücklich noch stillschweigend, bereitgestellt. Dies gilt u. a., aber nicht ausschließlich, hinsichtlich der Gewährleistung der Marktgängigkeit und der Eignung für einen bestimmten Zweck sowie für die Gewährleistung der Nichtverletzung geltenden Rechts.

SAP übernimmt keine Haftung für Schäden jeglicher Art, einschließlich und ohne Einschränkung für direkte, spezielle, indirekte oder Folgeschäden im Zusammenhang mit der Verwendung dieser Unterlagen. Diese Einschränkung gilt nicht bei Vorsatz oder grober Fahrlässigkeit.

Die gesetzliche Haftung bei Personenschäden oder die Produkthaftung bleibt unberührt. Die Informationen, auf die Sie möglicherweise über die in diesem Material enthaltenen Hotlinks zugreifen, unterliegen nicht dem Einfluss von SAP, und SAP unterstützt nicht die Nutzung von Internetseiten Dritter durch Sie und gibt keinerlei Gewährleistungen oder Zusagen über Internetseiten Dritter ab.

Alle Rechte vorbehalten.

SAP RESEARCH

© SAP 2007 / Page 85