

Report on Panel 2 – FIA Madrid on User generated services and user generated content: Similarities and Differences

Juanjo Hierro, Chief Software Technologist, Telefónica I+D

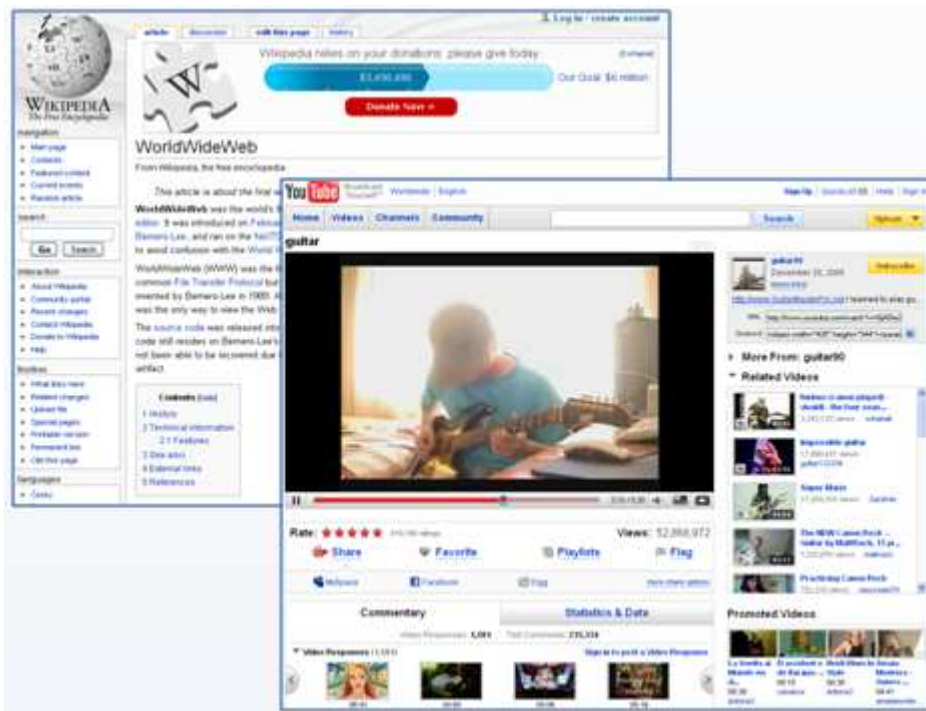
Andrew Oliphant, BBC Research & Innovation

1. Background and objectives of the panel

The Web 2.0 phenomenon has brought some lessons on what can be achieved by empowering end users to generate and share content. We indeed have learnt that:

- We shouldn't underestimate users ... if you give them the tools:
 - they will generate contents you had never figure out
 - they will deliver contents incredibly faster than you
 - They will innovate incredibly faster than you ... and in a collaborative manner!
- Visibility and recognition is a very powerful tool ... it fosters true engagement of users

Figure 1. Some well-known examples of User Generated Contents



As recognized in the ICT FP7 Work Programme 2009-2010, “‘third party generated service’ is emerging as a trend supporting the move towards user-centric services, as shown by the advances in Service-Oriented-Architectures and in service front-ends as the interface to users and communities” and, recognizing this fact, Service Front Ends have been identified as a major target outcome in Objective 1.2 “Internet of Services, Software and Virtualisation” particularly in the area of Service Architectures and Platforms for the Future Internet:

Objective ICT-2009.1.2: Internet of Services, Software and Virtualisation

Target outcomes

a) Service Architectures and Platforms for the Future Internet

- **Service front ends** enabling communities of networked users easily to compose, configure, share and use services and providing device and context aware service adaptations. They facilitate the development of, search for and interaction with services, cover the service life cycle and take account of social network users having different levels of expertise.

However ... Are lessons learnt in the “Content Space” experienced with the Web 2.0 phenomena really transferable to the Application space ? If so ... Wouldn’t this lead to the opportunity to establish an Architecture for Service Front Ends which may intermediate access to both application and content-related services in the Future Internet ? What will be the requirements for such architecture?

These three questions were the main questions to be addressed in the panel.

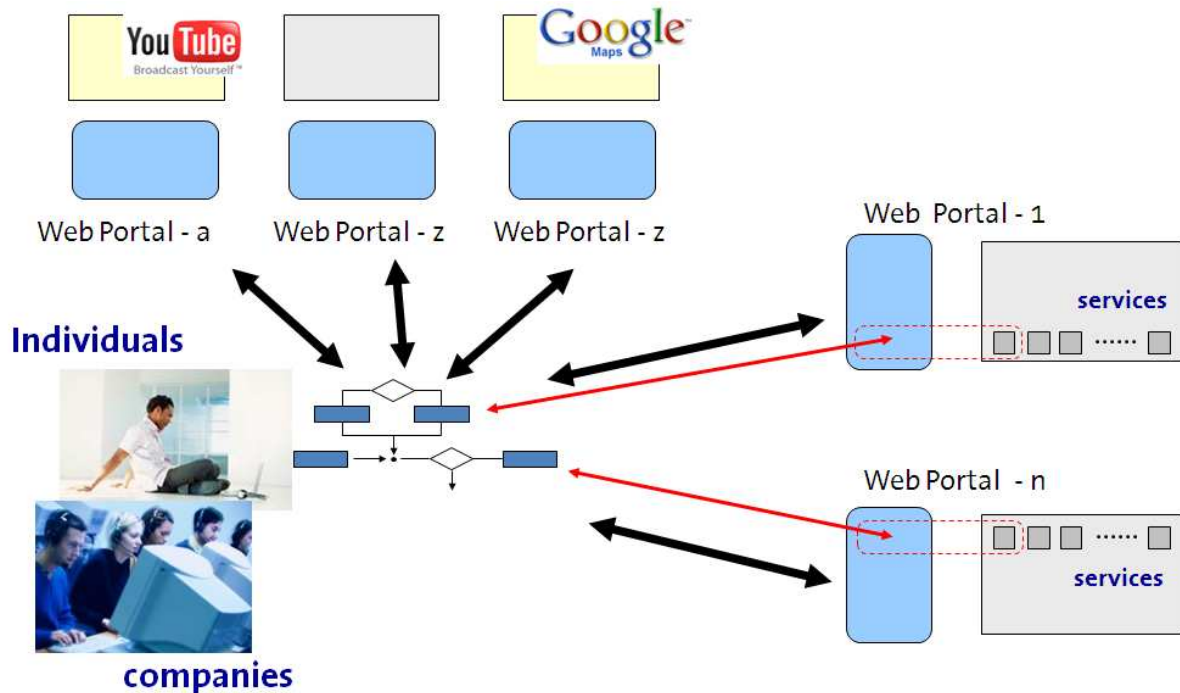
2. Position Presentation

Juanjo Hierro (Telefónica I+D) and Andrew Oliphant (representing BBC Research & Innovation) jointly prepared a position presentation to help situate the debate on questions raised for the panel.

Juanjo Hierro elaborated on the findings regarding empowerment of end users in the Application Space. One of the main peculiarities of this space has to do with the fact that end users intend to carry out **processes** which are not supported in existing Content-oriented Web 2.0 platforms today. However, is this an argument to justify that lessons learnt in Web 2.0 cannot be applied in the Application Space and a single Service Front End Architecture would not be feasible?

If we examine how end users interact with the Web today, we realize that end users have to orchestrate services on their own (they become Enterprise Application Integrators :-) manually searching, navigating and copying/pasting data from one web portal to another (see Figure 2). Besides, they are not able to save and automate orchestrated process or share what they have done with others who may face similar situations. This not only happens at the public Internet but inside large companies. Things even get worst in the latter case due to the fact that barriers sometimes impede to get access to services published on the Internet and orchestrate them together with services available on the Intranet. These Intranet/Internet barriers are artificially preventing innovation happening at a rapid rate on the Internet to be imported in SOA at companies.

Figure 2. How users interact with the Web today



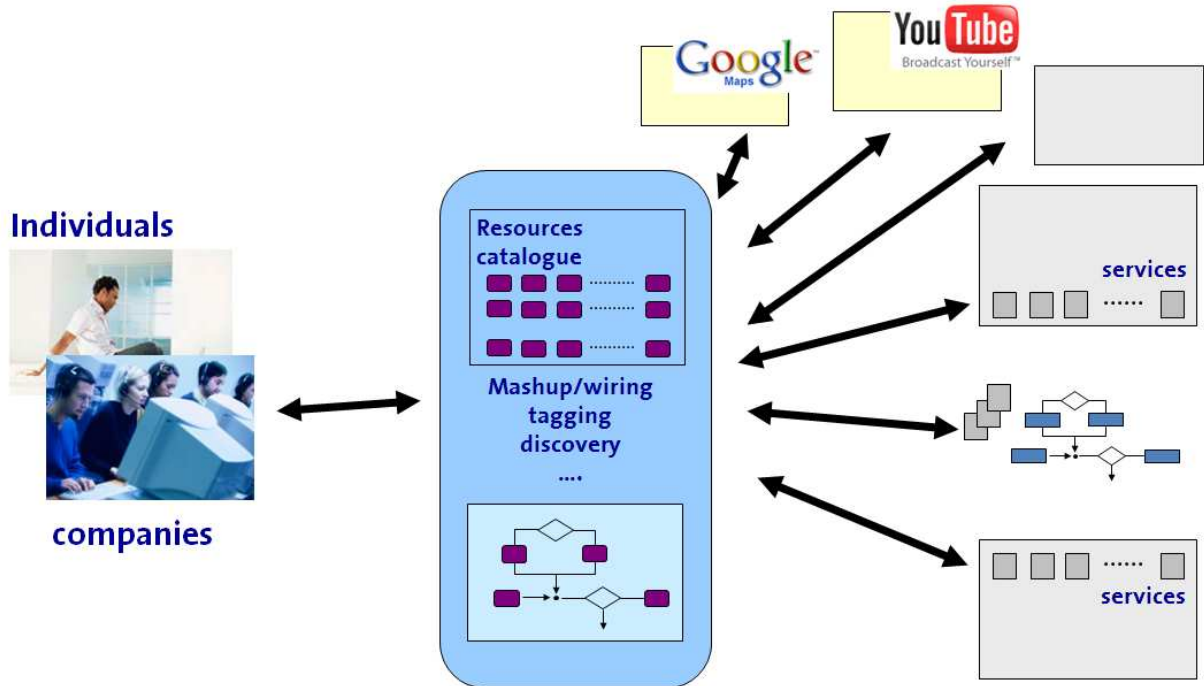
Some research projects in the European space (e.g., EzWeb Strategic NESSI Project funded by Spanish Avanza program, EU FP6 IP project OPUCE or EU FP7 STREP project FAST, CELTIC MyMobileWeb) have developed and experimented with tools that enable end users to design their own web access to the Internet of Services and share results with other end users. They are showing us that, same as with contents:

- We shouldn't underestimate users ... if you give them the tools:
 - they will generate **applications** you had never figure out
 - they will deliver **applications** incredibly faster than you
 - They will innovate incredibly faster than you ... and in a collaborative manner !
- Visibility and recognition is a very powerful tool ... it fosters true engagement of users

As Alberto León (project coordinator in OPUCE) stated in the session on Service Front Ends that took place in the ICT Lyon event, "the user is the killer application".

These projects decided to join efforts and recently created the Service Front Ends (SFE) Open Alliance which is aimed to integrate results of projects that share a common vision on the architecture that next-generation Web Front Ends may adopt in the future Internet of Services. Its aim is to produce open specifications as well as reference implementations of components in the envisioned architecture (some being open source.) Figure 3 outlines the architectural vision established as basis for work in this Open Alliance which probably is one of the first attempts to integrate results coming out from R&D projects since the Future of Internet initiative was launched.

Figure 3. Sketched Architecture for Service Front Ends



In the envisioned architecture, the concept of web resources/gadgets (REST-based) is introduced. Web resources enable to “put a face” on services and make them tangible and meaningful to end users. They are able to build their personalized web access point to services, by means of picking gadgets they find in a public catalogue and combining them to configure their operating workspace. Some web resources are developed and published by service providers (putting a face on services they want to export) while others are developed and also published by end users, using visual tools which help them to orchestrate existing services.

Front-end Web mashup features in the envisioned architecture go far beyond what current content-oriented mashup platforms on the Internet (iGoogle, NetVibes) offer today since several features have to be supported such as gadgets wiring, context-aware discovery and adaptation, ability to support a wide diversity of user profiles and access rights, multiple revenue models, etc. Feedback derived from experimentation in some of the projects already mentioned lead to the conclusion that some of these features would be also very useful in content-oriented mashup platforms. As an example, wiring content-oriented gadgets together seem to be useful (e.g., I may want to select a video title from a video library I can consult from a gadget and then visualize it on a video player gadget I have configured in my workspace, or I may want to select a photo from a gadget which offers a view on a number of user accounts in flickr and send it via MMS through another gadget).

Are there additional issues in common which could confirm that a single Service Front End platform to access both application and content-related services is feasible?

Juanjo Hierro and Andrew Oliphant explained that they tried to perform a simple exercise. Andrew identified scenarios in the content/multimedia space where we are currently facing some short of problem and trying to find a solution. Juanjo then tried to check whether such scenario could be mapped into the application/service space following the proposed architecture. Results of this exercise (see appendix at the end of this report) seemed to indicate that such a mapping is feasible and that a common set of requirements could be defined but, of course, require further research and experimentation. In a very first approach, a common set of functionalities/requirements seem to arise:

- Public Catalogue where gadgets wrapping access to both content delivery services and application services get published
- (Quite sophisticated) Management of users and users groups privileges, combined with social networking functionalities and gadget data, to handle:
 - Visibility over the catalogue
 - Rights for publication in catalogue of new gadgets
 - Rights to design mashups and publish them as items in the catalogue
 - Modification of published gadgets/mashups (redistribution)
 - ...
- Classification of gadgets:
 - Explicit/implicit tagging (sources: providers and users)
 - Maybe more sophisticated semantic-based mechanisms
 - Compensation mechanisms (to reward end users collaboration)
- Trust and authentication issues on:
 - gadget providers (how to ensure the provider is who is claiming to be, how to avoid malicious gadgets)
 - users (how to ensure the user is who is claiming to be, ...)
- Digital Right and Licensing Right Management issues between backend service providers, gadget providers, and gadget mashup platform providers
- Rich brokering functions enabling gadget providers to control how their gadgets get published in several catalogues and can be:
 - Tested by end users free of charge
 - Used (being able to restrict visibility to desired group of users)
 - Monetized (payment for actual usage, revenue sharing on incomes derived from advertising, ...)

3. Open Debate

Once the position presentation finished, an open debate took place. Pete Bramhall from HP Labs, UK, joined the panelist to elaborate on security aspects.

The debate was structured around the following questions:

- Do you find a significant set of common functionalities for both, contents and services, at the Front End? ... Can you enumerate some?
- Can these common functionalities rely on common technical solutions? ... Is a Common Service Front-End Architecture feasible?
- What research challenges do you foreseen in the Future Internet with respect to Service Front Ends?

There was a rich debate on these matters. Following is a list of major topics addressed during the debate:

- What kind of users do we envision in this picture? This was a topic that led to considerable discussion. Some people in the audience argued that generation of content is affordable for most of users (this is why the Web 2.0 phenomena on content has been so fruitful) but the same users would not be able to mashup applications together which seems to be a more complex task. Some people replied back that those who keep this statement are underestimating end users, the same way some people before the Web 2.0 phenomena stated that no one were going to produce videos and upload them on the Internet to share with other users. They stated that experimentation performed so far shows that users can do rather interesting application compositions if they are armed with the proper tools. It was also mentioned that there might be some advanced users who may be able to mashup applications on their own and rest of users just need to use what others have already designed and published.
- Is any kind of process going to be supported under this schema? What role is led to backend service orchestration? Some people argue that in existing Web 2.0 mashups and collaborative spaces, users do not handle processes. There is an inherent complexity in process modeling which seems to indicate that mashup technologies would not be suitable. Some people replied that some very simple processes can clearly be designed today based on mashup/wiring principles in a rather straightforward manner, much simpler than with proposed BPM/BPEL-based tools which impose models that are far from what end users can easily understand. They also stated that, anyway, the fact that there are still some issues not yet resolved is not a problem but an opportunity for research.
- Pete Bramhall elaborated on the fact that security aspects become huge if we give end users full empowerment. We all see how far we are with fraudulent usage of contents today ... with processes, damage may be even worse, so security issues have to be tackled rather carefully. Some argued that, again, this may be seem as an opportunity/challenges that require research

4. Next steps

The following steps for progress are envisaged:

- Identification of requirements linked to a common Service Front End Architecture that may provide access to both application services and content-related services
- Proposals for Service Front End Architectures and integration of results coming from FP and other European/National funding program towards materialization of each architectural vision.

5. Contacts

- Juanjo Hierro
Chief Software Technologist
jhierro atnospam tid.es
Telefónica I+D
- Andrew Oliphant
NEM vice-Chair for Liaison
andrew.oliphant@btopenworld.com
representing BBC Research & Innovation

6. Appendix: Mapping UGC examples into the services space

6.1. Use case 1

- **Description**

A musician wants to make short samples of her work available free to the general public. She also wants to be able to sell complete items to the public with the reservation that they may be handed on to not more than two other people and they may not be re-used or adapted without her consent. Items will need to be linked to other elements such as scores, other performances of the same kind of music, ...
- **Mapping to application service space:**
 - music performances (complete songs) = application gadgets
 - Samples = trial versions of application gadgets
 - Performances of the same kind = application gadgets of the same kind
- **Common requirements:**
 - Social features (scoring), semantics (to establish links to related gadgets)
 - Public catalogues, probably several, potentially federated
 - Management of different levels of permissions for end users

6.2. Use case 2

- **Description**

The same musician also wants to make complete items available free to Agents (who might be members of a trade association, for example) with the reservation that she is informed of those who access them and anyone they send the items to.

- **Mapping to application service space:**

- music (complete items) = application gadgets
- Agents = Broker of a given application catalogue

- **Common requirements:**

- Deployment of gadgets in multiple catalogues
- Usage accounting

6.3. Use case 3

- **Description**

A news provider wants to be able to accept still pictures, video and/or audio of news events and to make them available to the public. It must be able to authenticate the material submitted by checking the place and time where it was recorded; it must also get a certificate that the person submitting the material has the rights to submit it. It might wish to allow the person submitting to name a price (the news provider does not pay, but another organization might do so) and to perform an automatic negotiation. It must not be possible for the material to be re-used by others.

- **Mapping to application service space:**

- News providers = application gadget mashup platform provider
- Pictures, etc. about news events = application gadgets

- **Common requirements:**

- Certification of gadgets in public catalogue, trustworthiness
- Brokering between gadget provider and mashup platform provider

6.4. Use case 4

- **Description**

A sports fan would like to tag material made available by others to list, for example, goals scored by David Beckham, races won by a horse called Red Rum, crashes in which Lewis Hamilton was involved In each case, he must check that he can acquire the rights to tag the material and publish the list to a fan club. The members of the club might wish to offer a small payment, from their subscriptions, to anyone adding new material to the lists.

- **Mapping to application service space:**

- Material = application gadgets

- Tagging of material = tagging of gadgets in gadgets catalogue
- **Common requirements:**
 - Tagging of gadgets (either wrapping access to contents or applications)
 - Tagging permissions, payment in return for tagging & recommendation
 - Social networking aspects

6.5. Use case 5

- **Description**

A school Parents' Association wants to collect or tag so that its members can view pictures of a school event (concert, school play, sports match, school journey, ...). Because the pictures involve children, consent of the parents involved must be obtained, even if the pictures are submitted by students. Access to the pictures must be limited to school staff and those involved in the event.
- **Mapping to application service space:**
 - Pictures = application gadgets
 - School Parents' Association = group of users
- **Common requirements:**
 - Protection of hosted content / application-data against non authorized usage, user privileges on access to content / application gadgets
 - Grouping of users with different privileges

6.6. Use case 6

- **Description**

The webmaster of the website of a village church sees an attractive picture of her church on another, unrelated website (perhaps the local tourist board). She doesn't know this, but the picture belongs to a commercial picture library on a third website. She selects the picture and puts it on her website; her website contacts the picture library and exchanges a certificate from a trusted third party confirming that her website is not-for-profit, so she is told that for a nominal fee she can use the picture with an appropriate attribution to the picture library.
- **Mapping to application service space:**
 - Picture = application gadget
- **Common requirements:**
 - Trustworthiness on users
 - Different billing mechanisms, depending on users