Oxford Brookes University

School of Technology

Department of Computing, Web Technologies

Master of Science Thesis

PortalLib
Open component library
for a vertical portal

by

Nicolas Cynober

(ID: 06013763 - Contact)

Supervisor: Professor David Duce

Oxford, 2007

Acknowledgements

I would like to warmly thank all the W3C members who are working and fighting for the web as a social and accessible media.

> My hope and faith that we are headed somewhere stem in part from the frequently proven observation that people seem to be naturally built to interact with others as part of a greater system … If we end up producing a system in hyperspace that allows us to work together harmoniously, that would be a metamorphosis. Though it would, I hope, happen incrementally, it would result in a huge restructuring of society. A society that could advance with intercreativity and group intuition, rather than conflict as the basic mechanism would be a major change. If we lay the groundwork right and try novel ways of interaction on the new Web, we may find a whole new set of financial, ethical, cultural and governing structures to which we can choose to belong , rather than having to pick ones we happen to physically live in. Bit by bit, those structures that work best would become important in the world, and democratic systems might take on different shapes.
>
> (Berners Lee 1999:224)

And of course David Duce, Jonathan Bishop, Bill Johnston, Daniel Dardailler, Frédéric Cavazza, Luc Cynober, Julien Rateau, Laurent Galais, Frédéric Rogel.

Abstract

My background research is focused on the understanding of the web, functionally and technically, in order to define the current and future needs of a vertical portal. A vertical portal is a web site that provides a gateway or portal to information related to a particular community. My research emphasizes the convergence of the social web and the future of web as main media. This thought focuses mainly on:

- Web history: past and future.
  What was the web ten, fifteen years ago? What is it nowadays?
  This dissertation gives you a taste of what the web should be in few years. This vision is both functional and technical.
- Social web and new mass media: web 2.0 features.
  Here we talk about "many-to-many", "web 2.0", "web 3.0", communities, long tails, multi-media, multi-devices, UI, accessibility, trust and many other topics related to vertical portals.
  Functional evolutions are always linked to technical evolutions: we talk about AJAX, Semantic Web, RSS, Widgets, Web Services, RIAs.
- Vertical portals: analysis and comparison.
  Here we study current online communities through their portal. Horizontal portal & mass information VS vertical portal & niches.
- General community needs.
  What are the common functional needs of all communities? Share? Discuss? Create?
  What kind of technical specifications? Level of accessibility? Semantic? Customizability?

All this reflection will point out a standardization of web portal concepts. This standardization is described in part 3; it contains both functional and technical specifications. Each point of the specification comes from a reflection in the research part but all the thoughts of my research part don't necessary create a new entry in the portal specifications. All along the dissertation we keep focused on vertical features and communities. So you won't find here an exhaustive review on web 2.0 implications or latest web technical products.

The technical aim is to create an open library named PortalLib implementing concepts found during my background research. PortalLib can be implemented easily in any kind of vertical portal. PortalLib is customizable, scalable, resource efficient and documented. PortalLib version 1.0 should provide:

- Import / Export profiles with FOAF
- External authentication with OpenID
- AJAX components for a customizable user interface
- Widgets handling based on W3C working draft
- Semantic Web tools

PortalLib is based on Prado, a php component-driven framework. The library contains descriptions of each component: why this component has been created and what is its usefulness. The documentation explains also how to use and implement each component. You can download PortalLib and its full documentation on http://www.portallib.net.

Contents

1. Introduction

1.1. Rationale

1.1.1. Web as new mass media

Despite the most sceptics, everyone is aware that the web is becoming the first mass media. Nowadays young generations are growing up on internet and are using less and less old medias. Newspapers, radio and TV are now replaced by blogs, online radios and video portals like Youtube. Economics and scientists compare the weaving of the web to the printing revolution [17]. A new era begins where costs to access and diffuse information are reduced to zero. In this completely new world everything has to be rethinked and recreated. The World Wide Web is only eighteen years old and like a teenager at this age, it's still growing and getting maturity. And only nowadays people are realising how the web can be useful and fun, a place to learn, to entertain, to keep in touch with your network, your community.

1.1.2. Communities along the tail

Communities are everywhere and have always been targeted by old medias and advertisers. For decades medias are focusing on the largest communities, creating programs for "teens", "women", "elders" but in fact people are belonging to thinner communities and groups. This concept has been raised by Chris Anderson in his book "The Long Tail", where he explains that the mass market is turning into a mass of niches. These niches have different sizes and different characteristics and most of them were not accessible because they were not profitable. Internet allows to finally reach these emerging communities, but where are they located? In fact most of them don't have a portal to interact, exchange and create. Here is the main objective of PortalLib, reducing costs, democratizing the tools of production and easing the creation of a vertical portal for a specific community.

1.1.3. New technologies

Functional innovations are often bound to technical ones. The web is young and its technologies are not mature as well. That is why a lot of companies are fighting to control web technologies. Hopefully the World Wide Web consortium (W3C) controls this evolution, safeguarding fundamental values like accessibility and operability. The inventor of the web, Tim Berners Lee, has perfectly understood this critical issue and decided to found the W3C in order to accomplish his vision: a gigantic information marketplace, where individuals and organisations buy, sell and freely exchange information and information services among one other. That is why my dissertation also deals with the future technical issues of vertical portals, following and sometimes discussing W3C recommendations.

1.2. Outline of the problem

1.2.1. Functional issues

The sentence "vertical portal for a community" points out two questions. What is a vertical portal? What are the common needs of communities? Along our reflexion we shall try to analyse users' behaviours and other portal tendencies.

It is very important to keep focused on what are the needs of the final customers when you are creating a library. In fact a library is a set of useful tools for webmasters and webmasters are always focusing on their users, so it is relevant that our library builds components resolving concrete needs. This reflexion around communities covers many ways of studies, we'll take a look especially at the socials links (e.g. relationships between profiles), different mediums of expression (e.g. blogs), different medias used (e.g. text, audio, audio & video). But we'll also work on portals functionalities, with talks around customization and widgets.

1.2.2. Technical issues

As I mentioned, following W3C technologies is an essential choice because it is the only way to guarantee operability and accessibility. The main issue here is to find technological solutions to functional problems. E.g. People are belonging to multiple communities and sub-communities, how to open social links with other networks and communities? A technical solution could be the implementation of FOAF (i.e. friend of a friend project using semantic with RDF) and OpenID (i.e. universal authentication). Creating an open library also means that we have to take care of creating well documented components, working with other developers and integrating their feedback.

1.3. Objectives

1.3.1. Standardization of web portal concepts

Our researches and reflexions should raise various assumptions and concepts around vertical portals. The main idea here is to create a document which lists conclusions and advises about the creation of vertical portals. These concepts are functional or technical.

1.3.2. Open component library

The PortalLib project is a set of concrete solutions and tools that will make the creation of vertical portals easier. All the components of the library respond to a specific concept. Creating a library responding to all the concepts can be an overwhelming work so I have decided to implement the most valuable components first. Other components could be built after the dissertation delivery date. This project is created as open as possible so it shouldn't be a big deal. All the specificities of the PortalLib project can be found at the following address: http://www.portallib.net.

2. Research

2.1. Web history: past and future

2.1.1 Functionally: How the web has been used?

There is a huge gap between "what the web has been created for" and "how the web has been used". The web has been truly built on one man's vision: Tim Berners Lee. We can't really talk about internet without talking about Tim. Reading Weaving the web, the book where Tim (yes, I'll keep naming him Tim) relates web's creation, has been a great experience for me, it shows how technology can serve ideology. This man fights for his ideas, a vision of what the web should be, and all his choices are focused to realise his dream. Unfortunately his dream of a web '*where we have limitless choice because we don't have to take what the TV producer has decided we should see next*' (Berners Lee 1999:182) wasn't really shared and understood. First web sites were really weak, copying old medias, creating one-to-many communications, without any interactivity. Try out archive.org and use "the take me back" button, you'll be in pain if you are looking before 2000 for blogs (2004 blogs became mainstream)(Wikipedia *Blog*), video sharing (Youtube founded in 2005) or social networking (LinkedIn, MySpace, Friendster are 2003). All the interactive tools and services have been founded more than ten years after web creation. But when you are reading "Weaving the web", wrote in 1999 with thoughts 1990 aged, you can read:

> My hope and faith that we are headed somewhere stem in part from the frequently proven observation that people seem to be naturally built to interact with others as part of a greater system... Computers help if we use them to create abstract social machines on the web: processes in which the people do the creative work and the machine does the administration... If intercreativity is not just sitting there passively in front of a display screen, then intercreativity is not just sitting there in front of something "interactive".
>
> (Berners Lee 1999:224)

Tim's ideas have passed the dot com bubble, waiting more than ten years, and finally web 2.0 came out. It shouldn't be called web 2.0 but simply "THE web". But don't be too much delighted at that news, like I said web is still a teenager and isn't mature yet. Nevertheless, THE web has already transformed our world, information is now free and everywhere, we are always connected (sooner or later at 100%) and everyone produces on internet. This figure shows the new structure of the Web:

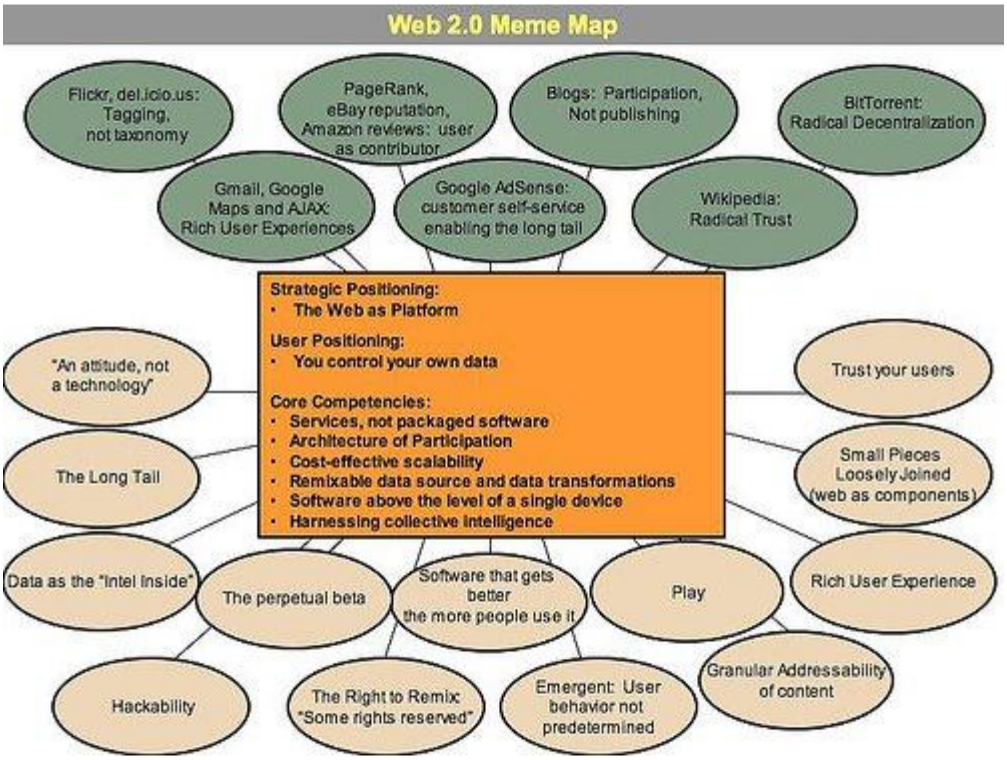

Web 2.0 Meme Map
Flickr, del.icio.us: Tagging, not taxonomy
PageRank, eBay reputation, Amazon reviews: user as contributor
Blogs: Participation, Not publishing
BitTorrent: Radical Decentralization
Gmail, Google Maps and AJAX: Rich User Experiences
Google AdSense: customer self-service enabling the long tail
Wikipedia: Radical Trust
Strategic Positioning:
• The Web as Platform
User Positioning:
• You control your own data
Core Competencies:
• Services, not packaged software
• Architecture of Participation
• Cost-effective scalability
• Remixable data source and data transformations
• Software above the level of a single device
• Harnessing collective intelligence
"An attitude, not a technology"
The Long Tail
Data as the "Intel Inside"
The perpetual beta
Software that gets better the more people use it
Play
Trust your users
Small Pieces Loosely Joined (web as components)
Rich User Experience
Hackability
The Right to Remix: "Some rights reserved"
Emergent: User behavior not predetermined
Granular Addressability of content

Figure 1. Web 2.0 Meme Map (O'Reilly 2005:1)

Ok, functional revolution is on the move but what about the web's core? This isn't the same deal...

2.1.2. Technically: How the web has been built?

When you are looking at the web's history you realise that internet has always been a field of war. Every company has tried to control the web, ISPs controlling navigation and browsers defining their own standards. In the middle of the field, referee Tim (always him), and the W3C crew have tried to organize this mess, pretty successfully and in the end nobody has won this battle. But the war isn't finished yet. The result of these fights is that the web has not really changed technically since its creation. HTML became XHTML and CSS is now used quite everywhere, so now content and presentation are separated. Nothing really exciting indeed... What about RSS? One of the most valuable creations of the web 2.0! It has been created in 1995 and it also took ten years to be widely implemented and used (Wikipedia *RSS*). And Ajax? This technology uses only HTML+DOM+XMLHttpRequest (Javascript) and all of them have been created before the dot-com bubble. The good thing is that all these technologies are now pretty mature and the web is ready to be upgraded, but what for?

2.1.3. A taste of what the web will be

No doubt that the web called 2.0 is on the good way, web has never been so democratized, and surfers' behaviours are closer to Tim's vision than ever. This dissertation will not attempt to predict the future but there is clear evidence that the web is entering into a new phase, structuring its communities and services. It seems that the next big step would be the appearance of a new technology. No it won't be RIAs, or 3Ds, or any complex and weird user interface. Internet still needs more accessibility and operability before becoming more complex. Web pages are rich enough; we mustn't widen the gap separating healthy people from people with disabilities (20% of population). So what is it? And if Tim had the answer (damn Tim, always him)... The Semantic Web (W3C 2001) could bring us in a new era, a web of data, an unending set of databases which are connected, a web machine-understandable, a better web for a better world, Tim's web.

2.2. Social web and new mass media: Web 2.0 features

What is a vertical portal? The role of vertical portals is to gather people and information belonging to the same community.

But there are already communities on blogs and on social networking sites?! Blogs ease information production and SNSs (social networking services) maintain links between people but a community still need its home, his own channel with specific services and tools.

A wine community and a basket ball community are sharing common tools on internet but not all of them. The goal of our reflexion here is to identify what these common needs are.

2.2.1. Web is social: many-to-many

2.2.1.1. The blog phenomenon

Web is different from old medias by the nature of the information flow. Old medias have channels containing shows which give information to people. The new media has always channels but now everyone is free to produce in it and to deliver the information. Nowadays some people have their own information channel, most common are blogs (text) but they are sometimes vlogs (video). These private channels are closely connected to their community, exchanging thoughts through comments and sometimes publishing the best comment as an article. Blog is becoming the first tool for producing content on the web because it's simple and thus used by the crowd (Sifry 2007).

In the same way, any member of a community should have a producing tool in order to deliver information with the rest of his community (functional concept n°1). Blogs are popular because anything can be posted simply whenever you want, maybe your post will be popular, maybe it won't. And if your post makes some buzz, it just means that you have produced something interesting for your community. A lot of people often said that you had to produce one topic per day on your blog, otherwise people won't come on it anymore, I have some concerns with this idea. If your blog is specific to a community (e.g. Blog about your hobby: wind surfing) and aggregated by the community's portal (e.g. French wind surfers), then your article will be promoted by portal mechanisms (using tags and quality markers) and maybe it will become popular, known by peers and recognised (Wikipedia *Virtual Community*) (functional concept n°2). People won't come on your blog every day to check if a new interesting article has been posted because the portal will do it for them, portal is used here as an aggregator. All information is now able to move from the end of the tail to the head (Anderson 2006:119).

By the word blog, you don't have to think about a place where someone talks about his day-to-day life. Here we are focusing on the format; you have to imagine community's blog as a unique producing tool where all the information produced by someone is referenced. When implementing this functionality into a vertical portal it might be a good idea to find a new name for it. In fact the word "Blog" is now too much linked to a personal journal. If we keep going on the French wind surfing portal, we could name it something like "Waves logbook". We will keep user behaviours, using a blog presentation but we will integrate attributes which are specific to this community. Indeed the information produced should use a specialized semantic. This will help the portal engine to classify the information produced. Shared folksonomies (tags) could be a good solution because it is widely democratized in the blogosphere (Sifry 2007) and these tags could be enhanced with more semantic (Spivack 2006:17). If you want to recover more semantic from a message you can also create templates for specific information types. Wikipedia already uses templates for a lot of different topics (Wikipedia *Infobox*). In our wind surfing case, the "Waves logbook" functionality could add a specific message type called "I found a new beach". If the user decides to use this template, then the portal will suggest a number of information to integrate in his message like the location, type of waves, wind, etc... Semantic would be added to each peace of information using RDFa and a beach ontology (Spivack 2006:16) (technical concept n°1). The portal engine is now able to understand this semantic and we can maintain a list of all the beaches referenced by our users.

2.2.1.2. Comments, because everyone has an opinion on everything

We were treating about information spontaneously produced by community's members but this information can also be a reaction to another article, video or any other piece of information. Conversations are starting around pretty anything on the web, so community's members have to be able to react everywhere. These reaction have to be ranked as well and using a proper format. Discussions have to be in live, using instant messaging and recording conversations for future users. People participating to a discussion could know when a new response has been added (technical concept n°2). (E.g. a new discussion service has just opened up and responds perfectly to that problematic www.tangler.com). Because comments are often as important as the original message commented, the portal should automatically add an entry on the user producing tool (expressed upper). Comments are too often downgraded compared to the initial information whereas people are proud about their comments. I've already seen a lot of time someone making a comment to a blog article directly on his own blog and then simply post a link like: "Hey check out my comment on my blog: http://ww...". Because he knows that his information is more valuable as a message than as a comment.

2.2.1.3. When one-to-'a few' is replacing one-to-many

Vertical portals have to provide tools in order to let the community living and growing by itself but they can also provide one-to-many information flow. We will keep our French wind surfing portal example. If that portal has been created by the institutional French federation of wind surfing then this institution may want to inform all the users about a new regulation or a new law concerning wind surfing. This example shows exactly the edge of our reflection on common functionalities for vertical portal; all these one-to-many functionalities will have to be developed beside the library. The library can still be used to implement one-to-many and many-to-many information into the portal design. We will develop this thought later in this dissertation.

2.2.4. Web is social: we create profiles

Profiles are used on every web site. They allow authentication and then identification. User's profile is linked to one unique ID in the website database; it makes the use of several functionalities possible around the user identity: functionalities customization, link between information produced and authors, relationship between profiles, etc... The main problem nowadays is that profile is already created on every web site with more or less details. Even relationships between profiles have to be recreated. The issue here is to share user profiles in order to ease authentication and the first steps of the registering processes. Nowadays some solutions are growing up but aren't widely used yet. The two well known systems are OpenID for user authentication and FOAF files for handling profiles. OpenID provides a way to be authenticated only one time on all the web sites you will visit. You will just have to give your OpenID URL which will be used to authenticate you on the current web site.

Concerning your basic information, like age, gender or friend list, semantic web can provide an easy way to share a common configuration file for all the sites. This profile configuration file could be FOAF. In this RDF file you will find your basic description and a list of your friends. The creation of a unique identifier on internet is critical; it could avoid identity violations (e.g. fake posts) and it will increase the efficiency of researches with more semantic. Nowadays it is very difficult to perform a research on articles written by a specific author because of the lack of semantic in the page. OpenID+FOAF+RDFa provide a quite good solution to that problematic and there are already systems managing both authentication and remote profile (e.g. videntity.org storing remote profiles and the PeopleAggregator framework for managing them). Every portal should implement OpenID authentication and import/export FOAF files (technical concept n°3). We will discuss later how the semantic web can be generally implemented in vertical portals.

2.2.5. Web is social: profiles are linked

A vertical portal should of course implement social networking functionalities. Objective here is not to recreate a full SNS but to create an architecture allowing people to maintain their social links. This should include at least:

2.2.5.1. Private and public messages/conversations

Public and private messages can become live conversations because they use instant notification thanks to Ajax (see 2.2.1.2.). Public messages are displayed on the user's profile, this system already exists on MySpace (comments list) and Facebook (The Wall) but these messages can't turn in a live conversation (Functional concept n°3).

2.2.5.2. Friend of a friend links

It's important to let the user builds his own network thanks to invitation mechanisms. Thanks to FOAF, not only profiles could be shared between different systems but also his list of friends and even groups.

2.2.5.3. Groups and sub-communities

Danah Boyd writes:

> In 1980s and 1990s researchers argued that the Internet would make race, class, gender, etc. extinct. There was a huge assumption that geography and language would no longer matter, that social organization would be based on some higher function. Guess what? When the masses adopted social media, they replicated the same social structures present in the offline world. Hell, take a look at how people from India are organizing themselves by caste on Orkut. Nothing gets erased because it's all connected to the offline bodies that are heavily regulated on a daily basis.

(Boyd 2006)

A vertical portal is created for one specific community and the portal has to let the community manages its sub-communities (Functional concept n°4) .Specific functionalities could be allowed to sub-communities like a private board and tools for sharing various contents (videos, photos, widgets), etc ...

2.2.6. Mass media means multi-media

Because internet is replacing traditional newspapers, radio and TV, vertical portals should attempt to broadcast all these different kind of media, text, audio and video (functional concept n°5). The first type of content is textual. Text is everywhere on internet, so it's quite easy to understand why newspapers felt the web's wave first (Fadner 2005).

Vertical portals should also broadcast audio, maybe creating a user driven radio linked to the vertical portal? (e.g. Last.fm). Anyway, many softwares allow to generate an audio version of an article. These audio versions are really useful for people doing something else at the same time (e.g. driving) or people with disabilities. It can be even possible to imagine reading articles on the portal radio.

Of course every vertical portal should include video. The main problem around videos on internet is accessibility for people with disabilities. Special versions for deaf are already available on TV, why not on internet? Videos have also a lack of semantic. It's very difficult to know what a video exactly contains, despite folksonomies which describe the global subject, we don't know exactly when the information you are looking for in the video is. Indeed it is a technical problem; most of the video on the web are running under a proprietary technology: Flash. Flash hasn't been created for broadcasting videos but for creating vector and raster graphics (Malone 2007). SMIL, a Synchronized Multimedia Integration Language, seams to be the best alternative (W3C 2007a) but unfortunately a lot of people are not concerned about accessibility and non-proprietary issues (Axistive 2007) (e.g. Firefox won't implement last versions of SVG and SMIL before the end of 2008, they are currently focused on security issues). Thus we will have to wait 2008 for a full implementation of SMIL 3.0. Considering this major issue, nowadays it is very difficult to state on a specific technology for broadcasting video+audio+text. I still consider SMIL implementation as a priority and so video handling in PortalLib will be built on two different layers: one uploading and storing videos without any format modification and one for the broadcasting (technical concept n°5).

What about a richer media? People are more and more talking about richer interfaces, with technologies like Silverlight (Microsoft), Flex (Adobe), full 3D interfaces, etc... But what exactly is the benefit except decreasing accessibility down to zero, spreading proprietary technologies on the web and finally shackling the semantic web? War continues, first victims are users (see 2.1.2.).

2.2.7. Mass media means multi-devices

We just said that vertical portals use multiple forms of content at the same time. You can find on the same page, text, embedded videos and audios and this rich content might be difficult to render on very different devices. We will focus here on mobile devices because we are entering in a new era of mobile devices. Indeed the next generations of mobile devices are merging all the existing mobile services. Nowadays you can find mp3 player, photo, video recorder, video player, radio, internet access and all powered with more and more storage capacity, processing power, larger bandwidth and wider screens. So I don't think we should create specific web technologies for mobile devices; we don't need anymore old mobile technologies like Wap or Imode, we just have to work on the user experience, surfing on a vertical portal using his mobile device. W3C standards hopefully let us separate content and form. Content is stored with a markup language like XHTML, RDF or any XML based technology while the form is designed thanks to stylesheets (CSS).

So do we just have to create a new CSS stylesheet for mobile devices? I don't think so. In fact mobile devices won't replace our 17 inches screen, our large keypad and our lovely mouse. User needs are different because he is on the move, input device is worse than his home keypad, his bandwidth isn't such good and his screen even with a good resolution has not been made for reading content on web sites. I think that vertical portals and even all web services should provide a specific version of their systems for mobile devices (e.g. the excellent: mobile.yahoo.com). A vertical portal should summarize its services into a user-friendly way designed specifically for mobile devices and maybe creating specific functionalities (technical concept n°4). It appears relevant to focus on how we can ease information access while information production will be more reserved for the 17 inches interface.

2.2.8. Vertical portals are open

In his well known paper 'What Is Web 2.0', Tim O'Reilly wrote:

> Much as the rise of proprietary software led to the Free Software movement, we expect the rise of proprietary databases to result in a Free Data movement within the next decade. One can see early signs of this countervailing trend in open data projects such as Wikipedia, the Creative Commons, and in software projects like Greasemonkey, which allow users to take control of how data is displayed on their computer.

(O'Reilly 2005:3)

In fact current social network sites are not social because all the information is bound to their technical edge. MySpace and Facebook aren't really pushing RSS. MySpace limits his site to specific HTML based widgets while Facebook creates a proprietary widget technology. None of those of course share their network through the web semantic but Facebook has at least an API (but only for embedded applications...).

The team who has written a framework for web science says *'On the Web, the relevant information is likely to be highly distributed and dynamic, personalisation is expected to be one of the big gains of the Semantic Web, which is pre-eminently a structure that allows reasoning over multiple and distributed data sources'* (*A framework for web science* 2006:47). In order to be distributed and dynamic a vertical portal should be open, that is why a vertical portal has to implement RSS (RDF Site Summary) everywhere an information is likely to be aggregated (technical concept n°6). Web services can also provide a good way to share processed information. Creating widgets for your vertical portal is also a critical action because widgets are a piece of your site included in another portal (functional concept n°6). Widgets are now everywhere: on client side (e.g. windows vista), on aggregators (e.g. netvibes), on mobile devices (e.g. iphone) and even on other portals (e.g. MySpace). Widgets have to be developed as a mini-site, so you should identify services that could be embedded externally. Our wind surfing portal could create various customized widgets like "last wind surfing photos", "wind on the beach", etc... Vertical portal is a community's home, when you are moving in a new home, you are coming with your stuff and then you personalise the place. So vertical portals have to be highly customizable, enabling widget inclusions, multiple skins and more generally a flexible user interface.

Semantic web is also *"an attempt to bring together data across the Web so as to create a vast database transcending its components, which makes possible applications that infer across heterogeneous data"* (*A framework for web science* 2006:19). The first rule when you create a database is to avoid duplicated information. Because the PortalLib project is about common tools for vertical portals, these tools could have a common database thanks to a web more semantic (technical concept n°7).

- Profiles are authenticated outside thanks to OpenID.
- Profiles and networks can be imported/exported thanks to FOAF
- Videos could be embedded in another site (we are waiting for SMIL).
- Photos can be imported/exported (e.g. FlickR)
- RDF and ontologies should be created and shared for verticals information.

2.2.9. User interface

The user interface has always been a tricky issue. How to increase usability? How to include ads? What level of user customization? PortalLib is a library which doesn't provide any graphical template, so webmasters are free to implement components as they wish. In all cases user-friendliness and ads depend of the nature of your vertical portal and your business model, so we won't argue here. Nevertheless customization is a critical point (*A framework for web science* 2006, Anderson 2006:169) and there are many different ways to enhance customization. The first thing is to allow users to choose a graphical themes (e.g. MySpace, Netvibes), because a portal should be your place, you can decorate it (CSS eases that process) (Functional concept n°7)(Technical concept n°8). Moreover these services could also be removed and added; the best way to do that is to provide customizable areas in the portal (e.g. Facebook). These customizable areas can contain widgets; it means pretty anything, and is of course based on the W3C work (W3C 2007b) (Functional concept n°8)(Technical concept n°9). Like we said before vertical portals have to create widgets for people using portal services on others devices (e.g. iphone) or others portals (e.g. netvibes). Widgets allow to broadcast outside the portal edges but also to let users customize them inside. It means that our vertical portal is using his own widgets which they are draggable, customizable or even removable. Maybe it's too obscure for now; we will try to clarify it by an example. Our wind surfing portal is composed of a network layer, different kinds of production tools for text, video and photos and maybe an online newspaper written by the community.

Different people are using the portal for multiple reasons and by multiple ways, someone will be attracted by only reading and sometime commenting so he will move up all widgets containing pure content: "Last articles of the e-newspaper", "Best videos", "Best photos", "Best user articles" and he will move down or remove the following widgets "Your wind surf community", "Most popular surfers", "Beach weather forecast", etc ... This user could also add external widgets from various sites (e.g. emails, agenda, RSS, etc...) and so create an highly customized portal around his favourite activity: "wind surfing". Of course each webmaster would be free to decide how far he would allow customization of his interface; we can guess ads won't be removable.

2.2.10. Accessibility

Because I've realised that a lot of people do not know what accessibility is, the following definition may be useful: *'Accessibility is a general term used to describe the degree to which a system is usable by as many people as possible. In other words, it is the degree of ease with which it is possible to reach a certain location from other locations. It is not to be confused with usability which is used to describe how easily an entity (e.g., device, service, environment) can be used by any type of user... focuses on people with disabilities and their right of access to entities, often through use of assistive devices such as screen-reading web browsers or wheelchairs'* (Wikipedia *Accessibility*). Any web developer should remember that 20% of people in Europe/America have disabilities. Population world-wide is aging, 20% over 60 now, 30% by 2020 and 70% of people over 60 have a disability. It is true, most people over 60 don't even know internet, but because they have not lived with it, it is not the same case for youngest generations. Moreover accessibility isn't only a problem for disabled but also for motorists, mobile workers, people in noisy environments and all people who needs hands free. Remember that 20% of Europeans have disabilities (90M) and only 40% are employed, telework on web apps could be a huge benefit for these people. We have already said that a vertical portal represents a specific community composed of several smaller communities, disabled are one of them (technical concept n°10). We can guess that in our wind surfing portal more than 10% are permanent or temporary disabled, we can't simply forget them. Every component of the PortalLib project will be specifically analysed in order to evaluate his accessibility level. We will follow the WCAG 2.0 (Web

Content Accessibility Guidelines) and we will use the conformance level defined by the W3C for each component.

2.2.11. Trust

Trust is involved at different levels. The simple question "is this information reliable?" involves multiple subconscious questions like "what are my past experiences on this site?", "do I know the author?", "who recommend me this information?". Vertical portals are mostly considered as niches so you are supposed to find a good specialized content. This is true when you are coming from an upper layer. Indeed the web can be divided by different layers of niches where each niche belong to an upper niche, the long tail phenomenon also traduce this concept explaining that each tail is founded by multiple smaller tails (Anderson 2006:139). Let's make an example around trust... Imagine you are looking for a new wind surfing board. So you make a research on your favourite search engine and you finally land on a personal blog. You read an article comparing the two last boards of a 14 years old kid and you notice that other messages on the blog are dealing with mangas. We can assume that in a vertical portal dealing with wind surfing, more trusted information concerning boards quality and prices should be found.

So you can imagine that trust is a very good lever for buying online and that's why niches are very interesting for advertisers, it is a matter of trust. Online ads services are displaying random ads on your site based on a poor semantic analyse of the content. Creating a vertical portal means you know well your community and his needs, the community should be able to find what kind of products or services are relevant to suggest. You don't need to create a new e-business site because you can customize your own shop (e.g. Zlio.com). What do you think about an online shop specialized in wind surf items, with thousand of comments and people talking about boards all the day?

2.2.12. Business model

In the last IAB report, Randall Rothenberg, President and CEO, said:

> Interactive advertising revenues continue to show solid growth as advertisers and agencies recognize that it is a medium that can uniquely impact consumer behaviour from product awareness, to purchase intent, to actual purchase and then brand loyalty. We have every confidence that this growth trend will continue as marketers allocate more of their total marketing dollars to interactive and the industry delivers effective and innovative platforms for connecting with consumers.

(IAB 2007)

We will keep two thoughts. The first one is that the web "can uniquely impact consumer behaviour from product awareness". Indeed Ads revenues are increasing by 33% each year and it would represent more than twenty billions dollars in 2007 (IAB 2007). The second interesting point is that advertisers are looking for "innovative platforms for connecting with consumers" and these innovative platforms are vertical portals. The Long Tail also reaches advertisers world, you can find ads for anything. So how can vertical portals attract advertisers? The definition we've actually given of vertical portals provides different tools for advertising. We have included more meaning thanks to RDF and OWL, so the machine is able to understand the content of the page and so include ads highly relevant. We have also advice to create vertical shops for sharing experience and comments around vertical products. With this 33% growth rate, we can expect that more and more small niches will be soon attractive for entrepreneurs. Especially when advertisers realize they are targeting the wrong community in a too wide horizontal portal (BBC News 2007). The advertising around wind surfing is mature? I'm not sure, but if it isn't, it won't be the case for a long time.

2.2.13. Web 2.0 technologies

Here we will talk about the famous web 2.0 technologies. How can they be used in our vertical portal concept?

2.2.13.1. Ajax

Ajax the well known, like we mentioned earlier, isn't such new, technologically speaking. It's a very interesting mix of different technologies and has allowed the creation of very famous and useful tools like googlemap, netvibes, etc... But Ajax must not be used everywhere, you should implement it carefully or it will destroy your site's accessibility. Ajax mustn't be critical in the user path; any task should be achievable without Ajax.

2.2.13.2. Semantic Web

Actually the Semantic Web is more know by his folksonomies (tags) than the real W3C Semantic Web (RDF+OWL). We believe strongly that the semantic web will be the next big step in the web history and vertical portals have their role to play. Indeed who knows better about a subject than its community itself? On this purpose Nova Spivack says*:*

> Where might we see this content initially arising? In my opinion it will most likely be within vertical communities of interest, communities of practice, and communities of purpose. Within such communities there is a need to create a common body of knowledge and to make that knowledge more accessible, connected and useful.
>
> (Spivack 2006:18)

So, Semantic Web should be implemented as far as possible in vertical portals, creating vertical ontologies and spreading his content through the web. Shifting from the old web to the semantic web won't be easy and fast. Anyway, some tools might help this transition. Promising technologies like RDFa or GRDDL should be widely used in the next years. But what is the real benefit for vertical portals? In fact a simple index of subject areas may not provide the community with sufficient ability to search for the content that its members require. To allow more intelligent syndication, vertical portals can define specific ontologies for the community.

2.2.13.3. RSS

RSS is a good transitional technology for sharing content. It's not obvious that RDF Site Summary will be replaced by an arising semantic web. I think people will always need summaries. Nevertheless RSS will continue to be transformed, containing more and more semantic.

2.2.13.4. Widget

As written above, widgets have also a promising future. It's a good way to embed portal's functionality in another device or another portal.

2.2.13.5. Web services

Web services should also find his way in a web more and more opened. Web services can provide a higher level of security and they are still useful to resolve complicated requests.

2.2.13.6. RIA

I do not see RIAs as the future of user interfaces but they could be used for games or for highly interactive applications (I think about 3D visiting). However RIAs still have a serious problem concerning accessibility and they shouldn't be used for diffusing information.

2.2.14. Other information relative to vertical portals

2.2.14.1. Security note

Working on security is of course a critical issue for any vertical portal but PortalLib won't support the whole security of your site because it's a library which is already based on another layer: Prado. Webmasters have to check as often as possible last versions of Prado and PortalLib in order to ensure the higher level of security possible. A system is never 100% secure but PortalLib will do his best.

2.2.14.2. Environment

Because the health of the planet is critically linked to our technological run, we advise developers to have an ecological policy (Spivack 2006:24-26). Vertical portals help building communities, and these communities will grow as long as the web grows. So your servers will need extensible capacities. If you don't want to waste energy you should take a look to utility computing services (e.g. Amazone EC2, Flexiscale).

2.3. Vertical portals: analysis and comparison

Portals on the web are too numerous to be listed and analysed. Nevertheless we have selected the most popular and the most interesting ones.

2.3.1. Horizontal portals

A horizontal portal is a web site providing a service for any type of community, everyone's welcome.

2.3.1.2. MySpace

According to Alexa Internet, MySpace is currently the world's sixth most popular English-language website and the sixth most popular website in any language, and the third most popular website in the US, though it has topped the chart on various weeks. The service has gradually gained more popularity than similar websites to achieve nearly 80% of visits to online social networking websites. It has become an increasingly influential part of contemporary popular culture, especially in English speaking countries (Wikipedia *MySpace*). The big problem is that MySpace pages are designed by individuals with little HTML experience so a very large proportion of pages do not satisfy the accessibility criterias laid down by the W3C (Wikipedia *MySpace*). Because MySpace is the biggest SNS (Social Network Service), and also the most horizontal one, we have just noticed that a lot of people are shifting from MySpace to more vertical sites (e.g. college student moving on Facebook) (Boyd 2007). MySpace is the biggest SNS because a lot of people basically thought that the best is the biggest. But finally users have realized that they want to meet people with common interests so they are moving to more vertical SNSs. This phenomenon should increase when portals will be able to import remote profiles.

2.3.1.3. Facebook

We can't consider Facebook as a vertical web site even if we have seen people shift from MySpace. When we take a deeper look into Facebook we realise that Facebook doesn't provide much more functionalities than MySpace and MySpace is even better in audio and video sharing. Facebook is currently trying to become the biggest SNS, so it has left his first goal: being a social networking site for college students. In my opinion Facebook should keep focused on his first and biggest community: college students.

2.3.1.4. Friendster

Friendster was considered the top online social network service until around April 2004 when it was overtaken by MySpace in terms of page views, mainly because MySpace was much more customizable, handling video and audio. Today Friendster is trying to fill the gap but it's too late.

2.3.1.6. Tribe

Nice web site highly focused on sub communities so this site does not own a specific community but a lot of little communities disconnected. The only good thing is that you can import/export a FOAF file.

2.3.1.7. Cyworld

90 percent of South Koreans in their 20s and 25 percent of the total population of South Korea are registered users of Cyworld, and as of September 2005, daily unique visitors are about 20 million (Wikipedia *Cyworld*). The site is highly focused on customization with a miniroom concept which is really interesting. But a business model around customization is really a good idea? Nevertheless, I don't know the Korean culture very well.

2.3.1.8. Orkut

The Orkut case is quite interesting because Orkut is a horizontal portal owned by Google and it occurs that Brazilian Orkut visitors count for 72.5% of the total users (Wikipedia *Orkut*). With this dominant Brazilian community, we can't consider Orkut as a horizontal portal anymore.

2.3.1.9. Bebo

Bebo is basically a horizontal portal but it announced on March 2007 that it was the most popular website in Ireland (Wikipedia *Bebo*).

2.3.1.10. Other horizontal portals studied

The summary table contains a review of Multiply, Yahoo 360, TagWorld and Imeem.

2.3.2. Vertical portals and niches

A vertical portal is built for a specific community.

2.3.2.1. Dodgeball

Interesting service focusing on geolocating users, the service is mainly used through mobiles. Unfortunately, that's the only interesting functionality.

2.3.2.3. Community Connect (AsianAvenue, BlackPlanet, MiGente, Glee)

Community Connect was basically focused on creating portal for US ethnic groups. They currently publish the three largest niche-targeted communities: AsianAvenue.com (Asian), BlackPlanet.com (African American) and MiGente.com (Latino). They have also created Glee.com (for Gays and lesbians) and Faithbase.com (for Christians). All sites are using the same template and the same functionalities so they are grouped in the summary table with the name "Community Connect".

2.3.2.4. Ning

With Ning your own vertical portal can be created, restricting registration and selecting your own functionalities.

2.3.2.5. PeopleAggregator

Broadband Mechanics' PeopleAggregator is an experiment in building social networks around open standards. Five years from now, we may look back on PeopleAggregator and consider it a pioneering product. Because PeopleAggregator is an open source project, we can't really say that PeopleAggregator is a Social Network Service, it's more like a framework based on three main layers: Authentication Layer, Import/Export Layer and a Vocabulary Layer. Nevertheless a full implementation of PeopleAggregator already exists and you can easily create your own customizable portal based on PeopleAggregator architecture. We have studied the main portal: peopleaggregator.net.

2.3.2.6. Other vertical portals studied

The summary table contains a review of Hi5 (Students), Ryze (Business), Studi (European Students), Zaadz (Thinkers, Changing the world), Piczo (kids & pre-teens, mainly UK & Canada), Skyblog (Teens, mainly French)

2.3.3. Analyse

2.3.3.1. When everything becoming vertical

When we take a deep look into horizontal portals we can realize they are not as horizontal as we basically thought, look at Orkut (75% Brazilian), Bebo (Mainly Irish), Cyworld (Korean) and Facebook (College student and high American classes). This phenomenon is quite new, people are moving from a SNS to another, trying to find the best place for building his network. Biggest horizontal portals are still growing because a lot of people are not yet involved in their "internet life" and most people are still on MySpace and Facebook. But when you take a look to the vertical niches, you might find some very interesting social community growing up. The case of Community Connect is very interesting, their portal for African American has 16.5 million members and the Latino's portal 2.8 million (as of June 2007), remember what Danah Loyd said '*When the masses adopted social media, they replicated the same social structures present in the offline world*' (Boyd 2006).

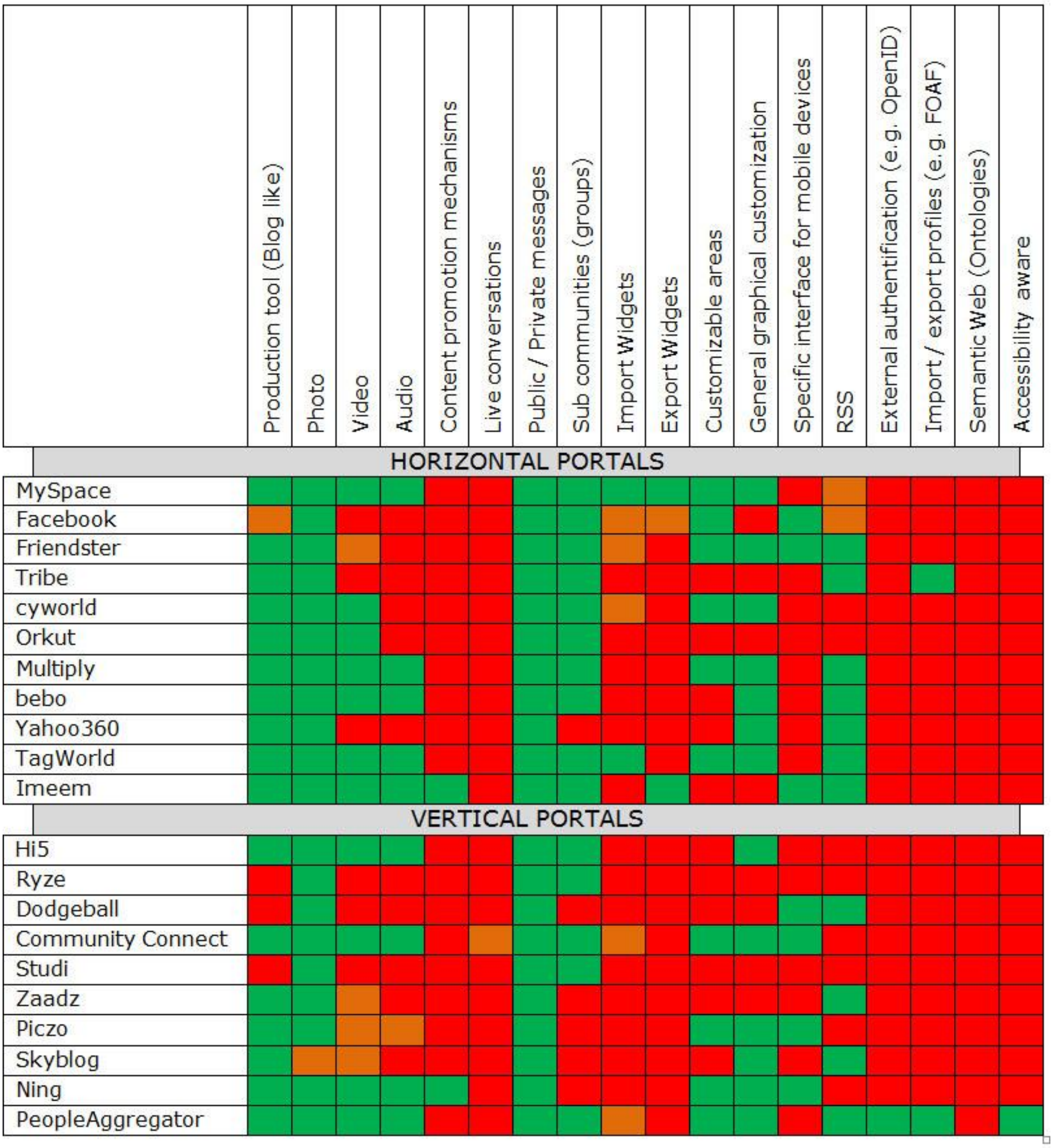
2.3.3.2. Migration

It is very difficult to see people moving from one service to another. Danah Boyd has worked on that question during several months, specifically relationships between MySpace and Facebook users, and she has discovered that higher American classes are moving on Facebook (Boyd 2007). His article has been discussed a lot on internet and I'm not sure that it is really higher classes; nevertheless there are a lot of people belonging to the same community offline who are gathering on the same portal. I strongly believe that this phenomenon will increase in the next years, in fact each time a new vertical portal is created. The future of vertical portal is of course to become open social network systems and ease these shifts. We can assume that MySpace and Facebook will stay "close" as long as possible, they won't let their users running away to smaller niches.

2.3.4. Summary table

This table summarizes portals reviewed. Criterias chosen correspond to the concepts pointed out during the research part (see part3).

Social Network Systems Review



	Production tool (Blog like)	Photo	Video	Audio	Content promotion mechanisms	Live conversations	Public / Private messages	Sub communities (groups)	Import Widgets	Export Widgets	Customizable areas	General graphical customization	Specific interface for mobile devices	RSS	External authentification (e.g. OpenID)	Import / export profiles (e.g. FOAF)	Semantic Web (Ontologies)	Accessibility aware
HORIZONTAL PORTALS																		
MySpace	🟩	🟩	🟩	🟩	🟥	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟥	🟧	🟥	🟥	🟥	🟥
Facebook	🟧	🟩	🟥	🟥	🟥	🟥	🟩	🟩	🟧	🟧	🟩	🟥	🟩	🟧	🟥	🟥	🟥	🟥
Friendster	🟩	🟩	🟧	🟥	🟥	🟥	🟩	🟩	🟧	🟥	🟩	🟩	🟩	🟩	🟥	🟥	🟥	🟥
Tribe	🟩	🟩	🟥	🟥	🟥	🟥	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟩	🟥	🟥
cyworld	🟩	🟩	🟩	🟥	🟥	🟥	🟩	🟩	🟧	🟥	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥
Orkut	🟩	🟩	🟩	🟥	🟥	🟥	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
Multiply	🟩	🟩	🟩	🟩	🟥	🟥	🟩	🟩	🟥	🟥	🟩	🟩	🟥	🟩	🟥	🟥	🟥	🟥
bebo	🟩	🟩	🟩	🟩	🟥	🟥	🟩	🟩	🟥	🟥	🟥	🟩	🟥	🟩	🟥	🟥	🟥	🟥
Yahoo360	🟩	🟩	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟩	🟥	🟩	🟥	🟥	🟥	🟥
TagWorld	🟩	🟩	🟩	🟩	🟥	🟥	🟩	🟩	🟩	🟥	🟩	🟩	🟥	🟩	🟥	🟥	🟥	🟥
Imeem	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟩	🟥	🟩	🟥	🟥	🟩	🟩	🟥	🟥	🟥	🟥
VERTICAL PORTALS																		
Hi5	🟩	🟩	🟩	🟩	🟥	🟥	🟩	🟩	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥
Ryze	🟥	🟩	🟥	🟥	🟥	🟥	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
Dodgeball	🟥	🟩	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟥	🟥	🟥	🟥
Community Connect	🟩	🟩	🟩	🟩	🟥	🟧	🟩	🟩	🟧	🟥	🟩	🟩	🟩	🟥	🟥	🟥	🟥	🟥
Studi	🟥	🟩	🟥	🟥	🟥	🟥	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
Zaadz	🟩	🟩	🟧	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥
Piczo	🟩	🟩	🟧	🟧	🟥	🟥	🟩	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟥	🟥	🟥	🟥
Skyblog	🟩	🟧	🟧	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟩	🟥	🟩	🟥	🟥	🟥	🟥
Ning	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟥	🟥	🟥	🟥
PeopleAggregator	🟩	🟩	🟩	🟩	🟥	🟥	🟩	🟩	🟧	🟥	🟩	🟩	🟥	🟩	🟩	🟩	🟥	🟩

By Nicolas Cynober. http://cyno.pbwiki.com. 29/09/2007

NOTE CONCERNING COLOR MARKS

GREEN:

- Compliant to the criteria

RED:

- This functionality is not provided

ORANGE:

- Video: the web site uses an external video web site.
- RSS: the RSS is too basic or hidden in the site
- The production tool (blog): it uses a plugin or a widget.
- Import widget: proprietary widgets
- Exported widget: doesn’t respect W3C standards or difficult to implement.

3. Standardisation of vertical portal concepts

3.1. Functional

3.1.1. Any member of a community must have producing tools in order to deliver information with the rest of his community.

E.g. Community's blog is a unique producing tool where all the information produced by someone is referenced. (See 2.2.1.1)

3.1.2. Promotion mechanisms must be added to each information.

Vote systems everywhere for evaluate accuracy and interest of each piece of information. All information should be able to move from the end of the tail to the head. (See 2.2.1.1)

3.1.3. Let users communicate through public and private messages

These messages can become live conversations. (See 2.2.5.1)

3.1.4. Let the community manages its sub-communities

Users can create and register to groups/sub-communities. (See 2.2.5.3)

3.1.5. Broadcast different kind of media, text, audio and video

(See 2.2.6)

3.1.6. Exporting widgets is critical

Widgets are now everywhere, on client side (e.g. Windows Vista), on aggregators (e.g. Netvibes), on mobile devices (e.g. IPhone) and even on other portals (e.g. MySpace) (See 2.2.8)

3.1.7. Graphical customization

It's your place and everyone doesn't have the same preferred colour. (See 2.2.9)

3.1.8. Provide highly customizable areas for importing widgets

(See 2.2.9)

3.2. Technical

3.2.1. When it's possible, semantic should be added to each vertical information.

This technical recommendation has been pointed out during a reflexion on blogs as main producing tool for portal's users. In that case templates could help the integration of RDFa with vertical ontologies. When there aren't ontologies for a specific information folksonomies (tag) are still a great way to categorize information. (See 2.2.1.1)

3.2.2. Messages and comments may become live conversations thanks to Ajax.

Ajax should be implemented anywhere a conversation can potentially start e.g. tangler.com. (See 2.2.1.2)

3.2.3. Implement external authentication and configuration files

External authentication (e.g. OpenID) and configuration files (e.g. import/export FOAF) (See 2.2.4)

3.2.4. Summarize services into a user-friendly way designed specifically for mobile devices

Focusing on how we can ease information access. (See 2.2.7)

3.2.5. Consider accessibility for video implementation as a priority

Be ready to implement SMIL as soon as a browser can render it. (See 2.2.6)

3.2.6. Implement RSS everywhere an information is likely to be aggregated

(See 2.2.8)

3.2.7. Common semantic tools means common database through the semantic web

The first rule when you create a database is to avoid duplicated information. (See 2.2.8)

3.2.8. Use XHTML + CSS would ease the graphical customization process

(See 2.2.9)

3.2.9. Widgets are following W3C recommendations

(See 2.2.9)

3.2.10. Accessible for everyone

Vertical portals should strictly follow WAI recommendations and the WCAG 2. (See 2.2.10)

4. PortalLib: Open Component Library for a Vertical Portal

4.1. Introduction

PortalLib is a Prado library responding to specific concepts defined in this dissertation. Using PortalLib reduces costs for building a vertical portal. Creating a portal for a niche is now profitable. This library has been designed over a component-based PHP framework named Prado. The main issue here is to find the best technologies and the best compromises for each component created. The PortalLib project has been created as an independent and open source project (under GPL licence). You can find fresher info here: www.portallib.net.

4.2. Open library

Because we want PortalLib to live after this dissertation we have decided to create this project under an open architecture. This project has been created under the GNU General Public Licence (GPL). We use a Sourceforge repository for managing different tasks (sourceforge.net/projects/portallib/). We have created a wiki for building the documentation, so people are free to write examples or comments on each component (portallib.wiki.sourceforge.net). Sourceforge also provides a forum for discussing with the PortalLib community and a tracker tool for resolving bugs and responding to new feature requests. The PortalLib project has his own blog at portallib.blogspot.com, check it out for the latest news.

4.3. Prado

Prado is a component-based and event-driven programming framework for developing Web applications in PHP 5 (www.pradosoft.com). We have chosen Prado as a technological base because it's an open source project, it uses a full internet programming language and the community is very active. Because Prado is built around components with inheriting principles it won't be difficult to create nice components that you can easily modify and reuse.

4.4. Methodology

Because we have defined a lot of concepts in this dissertation, we won't implement everything in this first version. It's very difficult to make a choice but I think we should focus firstly on non-graphical components and avoid for a moment interface customization and the design for mobile devices. Even if these two concepts are deeply important, they will need a huge work, especially around accessibility issues.

Each component is one by one created. For each one we produce tests around accessibility and scalability. Finally the component documentation is written on the dedicated wiki.

4.5. Components

4.5.1. PLAuth

This component handles the authentication layer through OpenID. It responds to the first part of the technical concept n°3: Implement "external authentication" and configuration files.

Demo page: http://www.portallib.net/demo/PLAuth/
Source: PLAuth (see also 7.1.)
Documentation: http://portallib.wiki.sourceforge.net/PLAuth
Version: 1.0
Dependencies: openid_simple (But included within the library)
Accessibility: Code based on a simple text input.
Description:
This component eases the implementation of an OpenID authentication. This component is based on the Prado TTextBox component. Checking authentication validity is a very simple two steps process:
1) $PLAuth->login();
2) $PLAuth->isValid();
The error message can be easily retrieved with $PLAuth->getError().
PortalLib doesn't provide an OpenID server system, just the client part (checking account validity). If you are interested in create an OpenID provider you should check out: phpMyID (Single user, easy to install), OpenID Enabled (Multi-user provider)

TODO: Implement an OpenID provider.

4.5.2. PLProfile

This component handles profile import/export through FOAF. This component respond to the second part of the technical concept n°3: Implement external authentication and "configuration files".

Demo page: http://www.portallib.net/demo/PLProfile/
Source: PLProfileImport, PLProfileExport (see also 7.2.)
Documentation: http://portallib.wiki.sourceforge.net/PLProfile
Version: 1.0
Dependencies: Pear XML_FOAF (based on RAP v0.7)
Accessibility: Code based on a simple text input.
Description:

The PLProfile component is in fact two components. PLProfileImport parses an external FOAF file. This component is based on the Prado TTextBox component. It's very easy to retrieve the FOAF attributes thanks to specific getters. PLProfileExport generate a XML Tree from a given php object. The PLProfileExport component is currently inheriting a TButton, it might be changed in order to increase flexibility. Both components use a FOAF object with simple setters and getters.

e.g.: PLProfileImport->getFoafObject()->getFamilyName()

Because this component is based on a Pear package, functionalities are limited by the package functions (the last version of XML_FOAF has been released in 2004). Fortunately XML_FOAF v0.2 already implements main FOAF attributes.

TODO: Only basic FOAF attributes are currently implemented, all the FOAF attributes should be available in future versions.

4.6. Tests

4.6.1. Benchmarking

Even if the first two components are pretty light we have performed performance benchmarks. The test machine has the following settings:

- Processor: VIA C7 2 GHz
- RAM: 1024 DDR2
- Hard drive: 152Go SATA2 Seagate
- System: Fedora Core 5

PHP configuration is available here: http://www.portallib.net/demo/serverinfo.php. Tests have been performed with the AB apache tool. AB has been configured to send 1000 requests to the server with a concurrency of 10.

4.6.1.1. Empty page

First of all we have done a simple benchmark with an empty Prado application:

```
<html><body></body></html>
```

Average time per request is: 139ms.

4.6.1.2. TTextBox

Then we had a text input to the page. This test is quite interesting because both PLAuth and PLProfile are based on a TTextBox component.

```
<html>
<body>
<com:TForm>
<com:TTextBox />
</com:TForm>
</body>
</html>
```

Average time per request is: 153ms.

So the TTextBox component adds only 14ms.

4.6.1.3. PLAuth

```
<html>
<body>
<com:TForm>
<com:PLAuth />
</com:TForm>
</body>
</html>
```

Average time per request is: 156ms.

So the PLauth component adds only 3ms compared to the Prado TTextBox component.

4.6.1.3. PLProfileImport

```
<html>
<body>
<com:TForm>
<com:PLProfileImport />
</com:TForm>
</body>
</html>
```

Average time per request is: 183ms.

It takes 30ms for the RDF-API to be loaded.

4.6.2. Online test

Various tests have been released on each component but it was essential to build a system using all of them. That's why I've created the website: http://www.guidedurider.fr. This site should be in a near future a very nice vertical portal for all French riders. Building a portal for this niche is a good opportunity for implementing next PortalLib components. Nowadays this is a simple site showing some sports location on a map. Both PLAuth and PLProfile components have been implemented and the result is quite interesting. Indeed the website doesn't store any password in its database; the authentication is fully handled by OpenID. During registration the user can use the PLProfile textbox or fill in a classical form. Finally if the user uses both components he could register and login in only 3 clicks and less than 10 sec (if he has already an OpenID and a FOAF profile).

5. Conclusion

This dissertation was really exciting in many ways. First of all, my researches on web history have pointed out one future for the web, a future following one man's vision. This vision has driven my researches: How communities could live on the web? How Tim Berners Lee's vision could help them? These deep researches about a web more semantic, open, trustable and accessible were also linked to nowadays tendencies as blogging, mobile web, etc... The weird thing is that during the last days of my dissertation I have discovered that some parts of the Tim's vision are now becoming tendencies:

- September, 20: Six Apart is opening the social graph using FOAF (Recordon 2007)
- September, 25: Orange becomes an OpenID provider and consumer (Nixey 2007)

Even it sounds good, Web Semantic and more generally W3C values have still a long road to reach mass markets and niches.

This dissertation was also a good opportunity to create my first open project: PortalLib. Even if the library is still quite empty (only 2 components), I have noticed that there are already very nice existing tools for handling SW technologies (RDF/OWL) with PHP. Each day, these technologies seem more and more mature. Anyway, weaving the semantic web shouldn't take long. Some great semantic web companies carrying a true vision are building the first semantic web killer-applications (dbpedia.org , freebase.com and radarnetworks.com are good examples).

Another very interesting point was the studies of current web communities. Horizontal networks have serious problems concerning security, child protection issues and specialized advertising. People are also bored by too big networks without any specific customization (horizontal networks are not build for a specific community) so new restricted networks are created: asmallworld.net or ning.com for instance. At the same time some horizontal portals are becoming vertical by the nature of their communities (Orkut can't be anymore considered as horizontal with a 72% Brazilian population). People should remember Chris Anderson's sentence: *'The mass market is turning into a mass of niches'*.

6. References

Wikipedia (2007). *Blog* [online], 2.1.1. From<http://en.wikipedia.org/wiki/Blog> [2 August 2007]

Berners Lee,T (1999). *Weaving the web*, 2.1.1.

O'Reilly,T (2005). *What Is Web 2.0* [online], 2.1.1 - 2.2.8. From <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html> [24 August 2007]

Wikipedia (2007). *RSS* [online], 2.1.2. From <http://en.wikipedia.org/wiki/Rss> [2 August 2007]

W3C (2001). *Semantic Web* [online], 2.1.3 <http://www.w3.org/2001/sw/> [3 August 2007]

Wikipedia (2007). *Infobox* [online], 2.2.1.1. From <http://en.wikipedia.org/wiki/Infobox> [4 August 2007]

Sifry,D (2004 to 2007). *State of the live web* [online], 2.2.1.1 - 2.3.3. From <http://www.sifry.com/stateoftheliveweb/> [4 August 2007]

Boyd,D (2006). *On being vertual* [online],2.2.5.3. From <http://many.corante.com/archives/2006/12/15/on_being_virtual.php> [8 August 2007]

Fadner,R (2005). *Proving To Be Worst Newspaper Year Since Recession* [online], 2.2.6. From <http://publications.mediapost.com/index.cfm?fuseaction=Articles.showArticleHomePage&art_aid=35690> [9 August 2007]

Malone,M (2007). *Flash Sucks* [online], 2.2.6. From <http://immike.net/blog/2007/07/31/flash-sucks> [9 August 2007]

W3C (2007a). *SMIL* [online], . From <http://www.w3.org/AudioVideo/> [9 August 2007].

Axistive (2007). *SMIL standards and microsoft internet explorer* [online], 2.2.6. From <http://www.axistive.com/smil-standards-and-microsoft-internet-explorer-6-7-and-8.html> [9 August 2007]

Anderson,C (2006). *The long tail: Why the Future of Business is Selling Less of More*, 2.2.1.1 - 2.2.9 - 2.2.11

Berners Lee,T & Hall,W & Hendler,J & O'Hara,K & Shadbolt,N & Weitzner,D (2006). *A framework for web science*, 2.2.8 - 2.2.9

W3C (2007b). *Widgets working draft* [online], 2.2.9. From <http://www.w3.org/TR/widgets-reqs/> [12 August 2007]

Wikipedia (2007). *Accessibility* [online], 2.2.10. From <http://en.wikipedia.org/wiki/Accessibility[> [12 August 2007]

W3C (2007c). *Accessibility* [online], 2.2.10. From <http://www.w3.org/TR/WCAG20/> [12 August 2007]

Spivack,N (2006). *The Meaning and Future of the Semantic Web* [online], 2.2.1.1 - 2.2.13.2. From <http://novaspivack.typepad.com/nova_spivacks_weblog/2006/11/minding_the_pla.html> [14 August 2007]

IAB (2007). *2006 Full Year Results* [online], 2.2.12. From <http://www.iab.net/resources/adrevenue/pdf/IAB_PwC_2006_Final.pdf> [14 August 2007]

Wikipedia (2007). *Virtual Community* [online], 2.2.1.1. From <http://en.wikipedia.org/wiki/Virtual_community> [14 August 2007]

BBC News (2007). *Firms withdraw BNP Facebook ads* [online], 2.2.12. From <http://news.bbc.co.uk/1/hi/uk_politics/6929161.stm> [14 August 2007]

Wikipedia (2007). *MySpace* [online], 2.3.1.2. From <http://en.wikipedia.org/wiki/MySpace> [17 August 2007]

Boyd,D (2007). *American class divisions through Facebook and MySpace* [online], 2.3.1.2 - 2.3.3.2. http://www.danah.org/papers/essays/ClassDivisions.html. [17 August 2007]

Wikipedia (2007). *Bebo* [online], 2.3.1.9. From <http://en.wikipedia.org/wiki/Bebo> [17 August 2007]

Wikipedia (2007). *Orkut* [online], 2.3.1.8. From <http://en.wikipedia.org/wiki/Orkut> [17 August 2007]

Wikipedia (2007). *Cyworld* [online], 2.3.1.7. From <http://en.wikipedia.org/wiki/Cyworld> [17 August 2007]

Recordon,D (2007). *We Are Opening the Social Graph* [online], 5. From <http://www.sixapart.com/about/news/2007/09/were_opening_th.html> [20 September 2007]

Nixey,P (2007). *Orange becomes an OpenID provider and consumer* [online], 5. From <www.webkitchen.co.uk/2007/09/orange-becomes-openid-provider-and.html> [25 September 2007]

7. Appendices

7.1. PLAuth code

```
/**
 * PLAuth class file.
 *
 * @author Nicolas Cynober <nicolas.cynober@gmail.com>
 * @link http://www.portallib.net/
 * @license http://www.portallib.net/license/
 * @version 1.0 6/09/2007
 * @package PortalLib
 */

/**
 * PLAuth
 *
 * PLAuth handles authentication through OpenID
 *
 * @author Nicolas Cynober <nicolas.cynober@gmail.com>
 * @version 1.0 6/09/2007
 * @package PortalLib
 */
class PLAuth extends TTextBox {

    /**
    * Array:
    * $error['code']
    * $error['description']
    */
    protected $error;

    /**
    * Initiate the component with a default textbox style
    */
    public function onInit($param){
        parent::OnInit($param);

        /* Default Style */
        $this->setStyle("background-image:url('../../../core/comps/PLAuth/openid.png');
                         background-repeat:no-repeat;
                         padding-left:20px;");
    }

    /**
    * Request OpenID provider for authentication
    */
    public function login(){
        $openid = new SimpleOpenID;
        $uri = $this->getText();
        $openid->SetIdentity($uri);
        $openid->SetTrustRoot('http://' . $_SERVER["HTTP_HOST"]);
        $openid->SetRequiredFields(array('email','fullname'));
        $openid-
>SetOptionalFields(array('dob','gender','postcode','country','language','timezone'));
        if ($openid->GetOpenIDServer()){
            $openid-
>SetApprovedURL('http://'.$_SERVER["SERVER_NAME"].$_SERVER["REQUEST_URI"]);
            $openid->Redirect();
        }else{
            $this->error = $openid->GetError();
            return $this->error;
        }
    }
    /**
    * Check if the authentication has been validated by the provider
    */
    public function isValid(){
        if($_GET['openid_mode'] == 'id_res'){
            $openid = new SimpleOpenID;
            $openid->SetIdentity($_GET['openid_identity']);
            $openid_validation_result = $openid->ValidateWithServer();
            if ($openid_validation_result == true){
                return true;
            }else if($openid->IsError() == true){
                $this->error = $openid->GetError();
                return false;
            }else{
                $this->error = array('code'=>'','description'=>'INVALID AUTHORIZATION');
                return false;
            }
        }else if ($_GET['openid_mode'] == 'cancel'){
            return false;
        }
    }

    /**
    * Return errors
    */
    public function getError(){
        return $this->error;
    }

    /**
    * Check for the provider response
    */
    public function isOpenIDResponse(){
        if($_GET['openid_mode']) return true;
        else return false;
    }
}
```

7.2. PLProfile code

7.2.1. PLProfileImport

```
define("RDFAPI_INCLUDE_DIR", ".");
require_once(RDFAPI_INCLUDE_DIR . "RdfAPI.php");
require_once("foaf.php");

/**
 * PLProfileImport class file.
 *
 * @author Nicolas Cynober <nicolas.cynober@gmail.com>
 * @link http://www.portallib.net/
 * @license http://www.portallib.net/license/
 * @version 1.0 6/09/2007
 * @package PortalLib
 */

/**
 * PLProfileImport
 *
 * PLProfileImport handles external FOAF profiles.
 *
 * @author Nicolas Cynober <nicolas.cynober@gmail.com>
 * @version 1.0 6/09/2007
 * @package PortalLib
 */
class PLProfileImport extends TTextBox {

    /**
    * Initiate the component with a default textbox style
    */
    public function onInit($param){
        parent::OnInit($param);
        /* Default Style */
        $this->setStyle("background-
image:url(./protected/controls/lib/portallib/comps/PLProfile/foaf.png);
                         background-repeat:no-repeat;
                         padding-left:40px;");
    }

    /**
    * Return an HTML table describing the entire RDF file
    */
    public function getHtmlTable(){
        $url = $this->getText();
        $model = ModelFactory::getDefaultModel();
        $model->load($url);
        ob_start();
        $model->writeAsHtmlTable();
        $result = ob_get_contents();
        ob_end_clean();
        return $result;
    }

    /**
    * Return a FOAF object containing file's attributes
    * NOTE: The XML_FOAF parser didn't work for me so I've decided to use directly RAP.
    * e.g.:
    * $parser = new XML_FOAF_Parser();
    * -> Fatal error: Cannot re-assign $this in /usr/share/pear/XML/FOAF/RAP/model/DbModel.php
on line 839
    */
    public function getFoafObject(){

        $foaf = new foaf();
        $url = $this->getText();
        $model = ModelFactory::getDefaultModel();
        $model->load($url);

        $rdfTypeResource = new Resource("http://www.w3.org/1999/02/22-rdf-syntax-ns#",
"type");
        $foafPersonResource = new Resource("http://xmlns.com/foaf/0.1/", "Person");
        $peopleResource = $model->findFirstMatchingStatement(NULL, $rdfTypeResource,
$foafPersonResource)->getSubject();

        $rdfTypeResource = new Resource("http://xmlns.com/foaf/0.1/", "name");
        $foaf->setName($model->findFirstMatchingStatement($peopleResource, $rdfTypeResource,
NULL)->getLabelObject());

        $rdfTypeResource = new Resource("http://xmlns.com/foaf/0.1/", "title");
        $foaf->setTitle($model->findFirstMatchingStatement($peopleResource, $rdfTypeResource,
NULL)->getLabelObject());

        $rdfTypeResource = new Resource("http://xmlns.com/foaf/0.1/", "givenname");
        $foaf->setGivenName($model->findFirstMatchingStatement($peopleResource,
$rdfTypeResource, NULL)->getLabelObject());

        $rdfTypeResource = new Resource("http://xmlns.com/foaf/0.1/", "family_name");
        $foaf->setFamilyName($model->findFirstMatchingStatement($peopleResource,
$rdfTypeResource, NULL)->getLabelObject());

        $rdfTypeResource = new Resource("http://xmlns.com/foaf/0.1/", "nick");
        $foaf->setNick($model->findFirstMatchingStatement($peopleResource, $rdfTypeResource,
NULL)->getLabelObject());

        $rdfTypeResource = new Resource("http://xmlns.com/foaf/0.1/", "mbox");
        if($object = $model->findFirstMatchingStatement($peopleResource, $rdfTypeResource,
NULL)){
            $foaf->setMbox($object->getLabelObject());
        }

        $rdfTypeResource = new Resource("http://xmlns.com/foaf/0.1/", "mbox_sha1sum");
        $foaf->setMbox_sha1sum($model->findFirstMatchingStatement($peopleResource,
$rdfTypeResource, NULL)->getLabelObject());

        $rdfTypeResource = new Resource("http://xmlns.com/foaf/0.1/", "homepage");
        $foaf->setHomepage($model->findFirstMatchingStatement($peopleResource,
$rdfTypeResource, NULL)->getLabelObject());

        $rdfTypeResource = new Resource("http://xmlns.com/foaf/0.1/", "depiction");
        $foaf->setDepiction($model->findFirstMatchingStatement($peopleResource,
$rdfTypeResource, NULL)->getLabelObject());

        $rdfTypeResource = new Resource("http://xmlns.com/foaf/0.1/", "workplaceHomepage");
        $foaf->setWorkplaceHomepage($model->findFirstMatchingStatement($peopleResource,
$rdfTypeResource, NULL)->getLabelObject());

        $rdfTypeResource = new Resource("http://xmlns.com/foaf/0.1/", "workInfoHomepage");
        $foaf->setWorkInfoHomepage($model->findFirstMatchingStatement($peopleResource,
$rdfTypeResource, NULL)->getLabelObject());

        $rdfTypeResource = new Resource("http://xmlns.com/foaf/0.1/", "schoolHomepage");
        $foaf->setSchoolHomepage($model->findFirstMatchingStatement($peopleResource,
$rdfTypeResource, NULL)->getLabelObject());

        $rdfTypeResource = new Resource("http://xmlns.com/foaf/0.1/", "phone");
        $foaf->setPhone($model->findFirstMatchingStatement($peopleResource, $rdfTypeResource,
NULL)->getLabelObject());

        $rdfTypeResource = new Resource("http://xmlns.com/foaf/0.1/", "name");
        $names = $model->find(NULL, $rdfTypeResource, NULL); //Not the best way to retreive
friends.
        $namesIterator = $names->getStatementIterator();
        $friends = array();
        while ($namesIterator->hasNext()) {
          $name = $namesIterator->next()->getLabelObject();
          if($name != $foaf->getName()){
              $friends[] = $name;
          }
        }
        $foaf->setFriends($friends);

        return $foaf;
    }

}
```

7.2.2. PLProfileExport

```
require_once("/usr/share/pear/XML/FOAF.php");
require_once("foaf.php");

/**
 * PLProfileExport class file.
 *
 * @author Nicolas Cynober <nicolas.cynober@gmail.com>
 * @link http://www.portallib.net/
 * @license http://www.portallib.net/license/
 * @version 1.0 9/09/2007
 * @package PortalLib
 */

/**
 * PLProfileImport
 *
 * PLProfileImport handles external FOAF profiles.
 *
 * @author Nicolas Cynober <nicolas.cynober@gmail.com>
 * @version 1.0 9/09/2007
 * @package PortalLib
 */
class PLProfileExport extends TButton {

    /**
    * Initiate the component with a default button style
    */
    public function onInit($param){
        parent::OnInit($param);
        /* Default Style */
        $this->setStyle("background-image:url('../../../core/comps/PLProfile/foaf.png');
                         background-repeat:no-repeat;
                         width:40px;
                         height:25px;");
    }

    public function getFoafXml($foafObject){

        $xmlfoaf = new XML_FOAF();
        $xmlfoaf->newAgent();
        $xmlfoaf->setName($foafObject->getName());
        if($foafObject->getTitle())$xmlfoaf->setTitle($foafObject->getTitle());
        if($foafObject->getGivenName())$xmlfoaf->setGivenName($foafObject->getGivenName());
        if($foafObject->getFamilyName())$xmlfoaf->setFamilyName($foafObject->getFamilyName());
        if($foafObject->getNick())$xmlfoaf->addNick($foafObject->getNick());
        if($foafObject->getMbox())$xmlfoaf->addMbox($foafObject->getMbox());
        if($foafObject->getMbox_sha1sum())$xmlfoaf->addMboxSha1Sum($foafObject-
>getMbox_sha1sum());
        if($foafObject->getHomepage())$xmlfoaf->addHomepage($foafObject->getHomepage());
        if($foafObject->getDepiction())$xmlfoaf->addDepiction($foafObject->getDepiction());
        if($foafObject->getWorkplaceHomepage())$xmlfoaf->addWorkplaceHomepage($foafObject-
>getWorkplaceHomepage());
        if($foafObject->getWorkInfoHomepage())$xmlfoaf->addWorkInfoHomepage($foafObject-
>getWorkInfoHomepage());
        if($foafObject->getSchoolHomepage())$xmlfoaf->addSchoolHomepage($foafObject-
>getSchoolHomepage());
        if($foafObject->getPhone())$xmlfoaf->addPhone($foafObject->getPhone());

        return $xmlfoaf->get();

    }

}
```