Internship report - ENSEIRB-MATMECA - Computer Science

Alipi : A Web Accessibility Framework

8 novembre 2012

Servelots - Bangalore, India - From June to September

Professors : Julien Allali, TB Dinesh.

Florian Boudinet.

Abstract

Servelots is a small company that develops several web open source softwares. It wants to improve the accessibility of the Internet to print-impaired people. This report presents and describes the leads explored and developed in this goal during this three-months internship.

Acknowledgments

I would like to thank all of the employees of Servelots for their warm welcome to their team and for helping me in my work and life in India. Furthermore, I especially would like to thank all of the people I met in India, who introduced me to the indian culture, and the human aspect of my work. I especially would like to thank M. TB Dinesh for this incredible experience in India, and all of his help during my whole internship (regarding the work as well as the private life). It was an amazing experience. I also would like to thank M. Sebastien Fourestier for his help to find this internship and his useful advices, and obviously, all of the teachers of the ENSEIRB-MATMECA for the quality of the education and the knowledge I acquired from them.

Table des matières

Chapitre 1

Introduction

There are several assistive technologies used for web browsing such as screen readers, speech recognition, screen magnification and keyboard overlays. Web-page authoring guide-lines developed by the Web Accessibility Initiative (WAI) of the World Wide Web Consortium (W3C) help in understanding and implementing web accessible content.

These guidelines however, do not have a prescription neither for the print-impaired nor for meeting the needs of an oral web. The issue of Web-accessibility for the print-impaired can be considered as the issue of the next-phase of Internet users - the next billion new users who may not be as literate as the earlier Internet users.

There is a large class of people who cannot read the content today on (most) web-pages. It will include billions of people who are non-literate, but they might soon be able to access the Web through their mobile phones (smartphones, tablets, ...).

This is also an issue for inter-cultural inter-language inter-contextual communication that the current web is trying to cope with. The main goal of Servelots is to improve the access of the web to as many people as possible. Especially, for Alipi project, the print-impaired person. Print-impairment is a little known sort of disabilities that designates people who can see but not read. The main reason is because of the language, or just the understanding (sentence too complex, location, etc...).

In a country like India, the population tolls millions of persons so it is a big issue in the country and for the people in Servelots.

The aim is to create a new way of describing, re-narrating and sharing the web to each others. My initial internship topic was about using semantic technology into Alipi project.

Finally, I worked on a different aspect of the project.

The internship can be divided in three unequal parts.
The first days were dedicated to the handover of the software developed at Servelots, integrating with the team, and getting acquainted. The aim was to get familiar with the environment and the indian way of life.
During this first step, I worked on semantic technology, their understanding, how to setup an ontology, a rdf store, and mainly the importance of using semantic technology for accessibility, and a full integrated web site.

The second part was to set a test for Alipi framework, using Selenium and Python. The third part was to create a Firefox plug-in for Alipi. The aim being to provide a greater immersion for the user, and to add new feature like friend list, automatic selection of the best renarration available, etc... I will first describe the context of the internship : the company and their software. Then explain each part of my internship.

Chapitre 2

Servelots : The Company

2.1 Geographical context

Servelots is based in Bangalore, the capital of the indian state of Karnataka. Located on the Deccan Plateau in the south-eastern part of the state. Bangalore is known as the Silicon Valley of India because of its position as the nation's leading IT exporter, and thanks to his famous university.
A demographically diverse city, Bangalore is a major economic and cultural hub and the second fastest growing major metropolis in India.

2.2 Company name

Servelots is a web service provider for Small to Medium Enterprises.

It was founded in 1999 by a group of Computer Scientists who wanted to provide a highly cost effective but user friendly software for SME's with a special focus on the organizations working in the social development sectors. They wanted to help small companies and indian web users to access the internet easily, even without knowledge.
They develop several open source softwares.

Servelots has a social responsibility arm named Janastu. Janastu is a NPGO, which provides IT services for the voluntary and civil society organizations. It focuses on community empowerment projects by helping organizations to customize the Pantoto software for their needs. A lot of people and volunteers are involved in Servelots and Janatsu. Not only the people they work with, but artists, craftsmen, or others volunteers as well. This aspect is really important because it brings to Servelots a great variety of knowledge on different topics.

2.3 Realisations

2.3.1 Pantoto

Pantoto is a Free and Open Source toolset that enables the organizations to manage their information and communication without software development dependency. Over the last few years, Pantoto has been successfully deployed in various organizations through training a small group on the concepts of Information and Community Management.

2.3.2 Alipi

Alipi is a web solution for accessibility. Because there are a lot of print impaired or uneducated person in India, Servelots decided to provide a web solution to help people to translate, annotate, customize, and edit web pages. This is the project on which I worked during my internship.

2.3.3 Follow the Sheep

Helping people to walk with the shepherds for a few days. People interested in walking with the sheep to collect track information using GPS and other information related to the land, the sheep and the shepherds such as the CPRs, the water bodies, the pastures, the farms and roads, the forests and the green, and the lives of the communities. The goal is to get as much information as possible to help shepherds do their job and just be recognized by the country. This is a new project which was launched this summer.

2.4 Organisation and Turnover

The developing team is composed of 5-8 indians engineers who also receive the support of several external developers interested in Sevelots's project. The team is led by Mr. T.B Dinesh.

The whole team works in the offices of Servelots, where everyone bring his own computer. Almost every two weeks, we had meetings to discuss projects and technologies to share knowledge discovery, ask questions and define the roadmap. Therefore, the design phase and software engineering are quite important for the company. The main funds are obtained from others organizations and companies interested in the project.

All of the softwares developed are open source and free, and everything is designed with this kept in mind. More than the software, it's the idea and the solution made that are important.

Chapitre 3

Semantic Web

The first month was dedicated to the handover of the developed software, integrating with the team, getting acquainted, and starting to handle semantic technologies. The aim was to get familiar with the environment and the indian way of life. During this first step, I worked on semantic technology. The main goal of semantic technologies is to separate data or content file from application code.

According to Wikipedia :

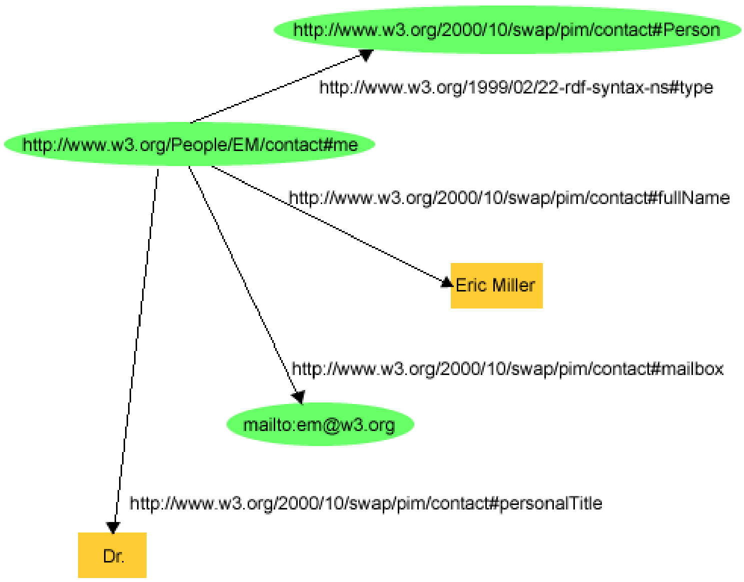
Semantic Technology "enables machines as well as people to understand, share and reason with them at execution time. With semantic technologies, adding, changing and implementing new relationships or interconnecting programs in a different way can be just as simple as changing the external model that these programs share."

3.1 What is the Semantic Web ?

The semantic web is a collaborative movement led by the international standards body, the World Wide Web Consortium (W3C). The semantic aims at convert unstructured and semi-structured data from the web, into a web full of data.
The Semantic Web stack builds on the W3C's Resource Description Framework (RDF).Those RDF file can be stored and accessed by an RDF store.

Thanks to the semantic web, datas can be fully understood by machine, and we can create a better search, etc...

For example, when somebody speaks about a friend in Blogger, the computer can't know who this user is, his name, address, and even more, he can't know if he is a real person. By using semantic technologies, the blogger, can "link" his friend, and define him as an user, and refer to his personal web page. The computer is now able to fully understand the entity.


http://www.w3.org/2000/10/swap/pim/contact#Person
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.w3.org/People/EM/contact#me
http://www.w3.org/2000/10/swap/pim/contact#fullName
Eric Miller
http://www.w3.org/2000/10/swap/pim/contact#mailbox
mailto:em@w3.org
http://www.w3.org/2000/10/swap/pim/contact#personalTitle
Dr.

This is a basic schema for the RDF technology. The user defines all of the information about an user, using known entities class (like User, PersonalTitle, FullName, etc..)

3.2 Setup

To add semantic information to data, we need to add attributes, and define these attributes. This definition can be defined by the developer in the same web domain, or use existed files, like FOAF, DublinCore, etc... This definition can use RDF format, or extension, like OWL (Web Ontology Language). When the definition (schema) file is done, we can add RDF information into our data (alipi information). There are different ways to add these information. We can add RDF attributes directly inside the XHTML.

Example :

```
<span property="dc:creator">Florian Boudinet</span>
```

Here, we specify that Florian Boudinet is a Creator, for DublinCore language.

It's the easiest way to add information inside an existing web page. The other way, the one used for alipi, is to dynamically generate a RDF file -like XML- using a web service.

Example :

```
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:dc="http://purl.org/dc/elements/1.1/">
      <rdf:Description rdf:about="http://en.wikipedia.org/wiki/Tony_Benn">
              <dc:title>Tony Benn</dc:title>
              <dc:publisher>Wikipedia</dc:publisher>
      </rdf:Description>
</rdf:RDF>
```

I haven't worked on this part. I was asked to search information on RDF and to learn.

3.3 TripleStore

But the power of the semantic web is in TripleStore. They are like specific databases for RDF metadatas.

Almost like a relational database, one stores information in a triple store and retrieves it via a query language. Unlike a relational database, a triple store is optimized for the storage and retrieval of triples. In addition to queries, triples can usually be imported/exported using RDF and other formats. Some triple stores can store billions of triples. Finally, we can access the data through SPARQL, a RDF query language.

With all these tools at our disposal, we can add a new aspect to our framework, stronger links with others webs services, but also make available to whoever is interested, a total access to metadata in a standard and powerful standard. A good example is a semantic search Web engine that seeks to improve search accuracy by understanding searcher intent and the contextual meaning of terms as they appear in the searchable data space, whether they are on the Web or within a closed system, to generate more relevant results.

Chapitre 4

Alipi Framework

Alipi is a web solution about accessibility provided by Servelots. Because there are a lot of print impaired or uneducated people in India, Servelots decided to provid a web solution to help people translating, annotating, customizing, and editing web page.

A person using alipi service can modify any web page to change his content. For example, if somebody notices a mistake on a web page, for example a wrong phone number or anything, the user can edit this element and publish the renarration to his personal blog or to Alipi's blog. Text, video, music and image can be edited/removed/added. The main idea is to fit the web to the user, and not the opposite. The idea is to develop a Web structure, which is localized to every human in terms of its language and description. Such a web structure will enable idea-sharing and similar thoughts to any other social media (like Facebook, Twitter), but in a more distributive and easy way.

Ryan (webpage-http ://www.psych.rochester.edu/faculty/ryan/) has done tremendous studies in human psychology to find out the factors, which motivate humans to contribute in social systems. Among the different categories of motivation factors, we believe for Alipi a possible factor can be the reputation in social media, which abides people with

certain standards, rules and protocols. A person in social media is exposed by his public profile, which may be Twitter, Facebook or any other. This public profile will simultaneously help increasing credibility of annotations. Emotional awards can also persuade users, making them feel they have done something for someone. Alipi does not deal with any sort of entertainment or fun, but it is only a social interaction. Therefore, the challenge is to persuade people for such a contribution.

The idea is now to provide a full user-friendly experience for Alipi. We need to show available narrations for each web site while the user is browsing the web. He can be able to see the best renarration easily and quickly. So we decided to set up a Firefox plug-in. This plug-in choses the best renarration for each web page and modify the page, based on the user preferences. The user can chose his favorite languages, retrieve his or her friends from Facebook, Twitter or Google, follow some re narrator, or specify his disabilities. Furthermore, an user can be able to see easily all the renarrations available, by languages and by authors.

4.1 Presentation

4.1.1 Vocabulary

- **XUL :** XML User Interface Language, developed by the Mozilla Project. XUL is a user interface markup language implemented as an XML dialect ; it allows for graphical user interfaces to be written in a similar manner to Web pages.
- **XPI :** XPInstall (Cross-Platform Install) is a technology used by the Mozilla Application Suite, SeaMonkey, Mozilla Firefox, Mozilla Thunderbird and other XUL-based applications for installing Mozilla extensions that add functionality to the main application.
- **Alipi.us :** Server provided by Servelots. WebService + Database
- **User Preferences :** A set of variables define by plugin user. Such as favorite language, user name, list of friends, disabilities etc...
- **Chrome :** The visible graphical interface features of an application, in fact, it's all the things defined by a XUL file. (No, it's NOT Google Chrome ! ! !)

4.1.2 Paquet XPI

This is how a classic Firefox Plug-in is built. Especially Alipi-Plugin. It's important to have a full understanding of the architecture, folder organisation and design.

```
extension.xpi/
    install.rdf
    chrome/
        content/
            files.xul
```

```
                files.js
                ...
        components/
            extension.dll
            extension.js
        defaults/
            extension.something
            preferences/
                extension.js
```

```
Main files are in ./chrome/content/.
```

A XUL file is XML-like. XUL is a language created by Mozilla to describe and define differents elements and actions in the browser.

For example, you can define a toolbar, and create button inside.

```
<toolbox>
   <toolbar id="findfiles-toolbar">
    <toolbarbutton id="opensearch" label="Open"/>
    <toolbarbutton id="savesearch" label="Save"/>
   </toolbar>
</toolbox>
```

4.2 Architecture and Design

This section will explain how the software was designed and the work provides. Some aspect are a little bit technical, but are needed to a full comprehension and a complete overview of the work done. Tree

Based on my knowledge and lessons learnt in school, I design some simple Javascript module.

The plug-in is divided in severals files by meaning and class. XUL files define the GUI and JS files are used to describe functions and can be used to filed XUL files (with AJAX query for example).

- **The main** file/class is alipi.xul. It describes what to do when a page is load, store and retrieve data from user preference (using mozilla-service), define a GUI for the user (new toolbar).
- **Settings** is used to define and field all the content in the edit-menu provide to the user.
- **edit-preference** define the window pop-up used to show and edit the user-preference file

- **add-language** define the window pop-up used to manage user languages
- **setup-preference** define a form to edit the user-preference.
- **module_id** is a used to communicate with web service like get facebook or twitter friend from a server (Ex : alipi.us)

```
alipi             (xul/js)
settings         (xul/js)
edit-preferences     (xul)
add-language          (xul)
setup-preference     (xul)
module_id         (js)
```

4.2.1 Identification System

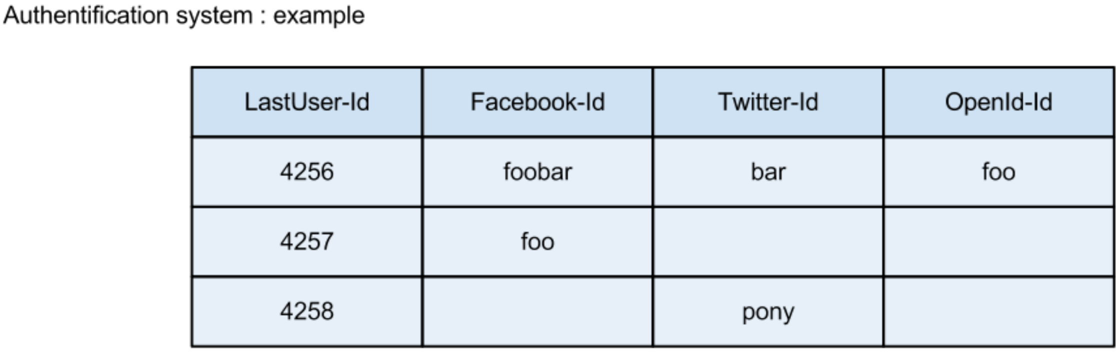
In order to compare user friends and renarration author, we need an uniformisation of the name in the whole process, to provide an identification system to disambiguate global ID provide by Facebook, Twitter or Google. Even more, perhaps, sometime some renarator will used a different name in alipi. That's why we need an identification system for the whole alipi experience.

First, we need something which allow the plugin and alipi to match a Facebook, twitter and G+ id with an universal id system.

The second step is to force the narrator to connect to the authentication system provided by alipi.us. Then, we can track and recognize the user.

This authentication system is called LastUser, and was created by HasGeek. It allows user to login with an Facebook/twitter/Google/gitHub/OpenID account, or create a new LastUser account. The user can provide on alipi.us others kind of account, and, like this, match each profile with the same id.

So now, an user can retrieve all his facebook/twitter friends, and recognize those id with alipi's author.



Authentification system : example

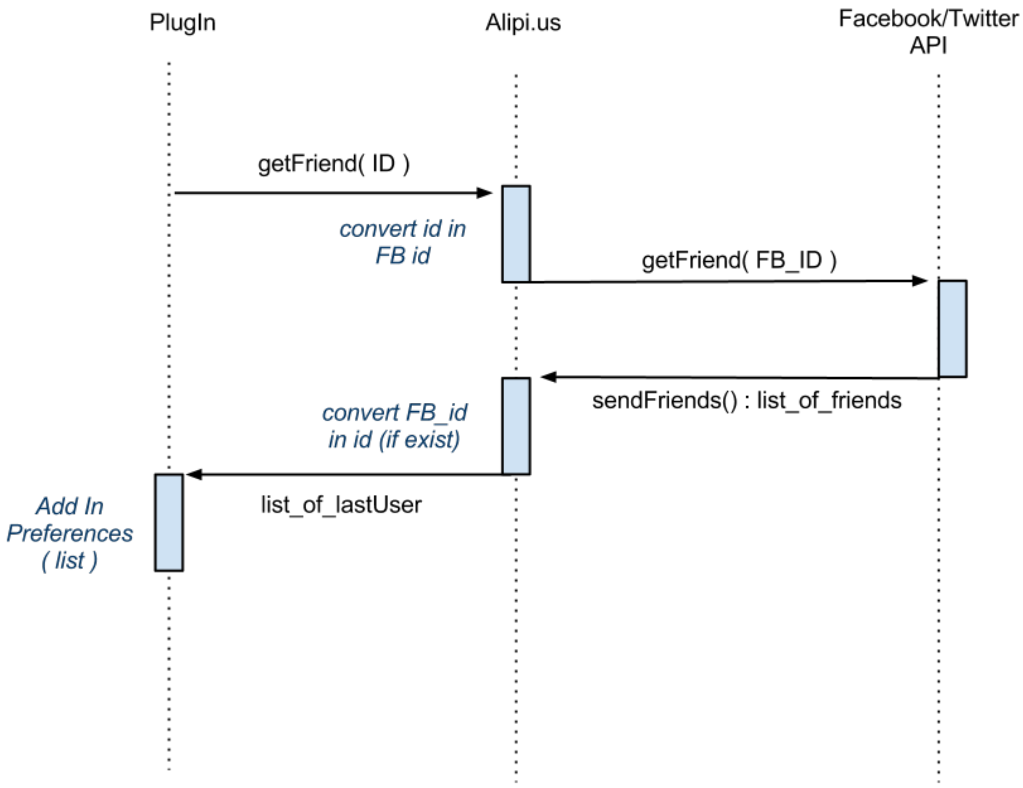
LastUser-Id	Facebook-Id	Twitter-Id	OpenId-Id
4256	foobar	bar	foo
4257	foo		
4258		pony	

4.2.2 Filter

The main plug-in function is to match the best renarration for the user, and display it on the screen, without human intervention. You have not modified the web, but the web has been modified for you.

The user, through the plug-in can edit his preferences. Preferences are stored in Mozilla Service Component (storage system used by the browser) as a JSON file.

```
{
    "name" : "Foo",
    "gender" : "male",
    "languages" :
     [
        { "name" : "marathi", "priority" : 1, "category" : "read-and-write" },
        { "name" : "english", "priority" : 2, "category" : "read-and-write" },
        { "name" : "hindi", "priority" : 3, "category" : "read-and-write" },
        { "name" : "telugu", "priority" : 4, "category" : "listen-only" }
    ],
    "friends" :
    [ ‘‘Bar’’, ‘‘foobar’’, ‘‘flyingSpagMonster’’ ],
    "location" :
    [
        { "current" : "parbhani" },
        { "hometown" : "parbhani"},
        { "visited" : "new delhi")
    ]
}
```



Preferences can be edited by several way. The user can use the languages editing tools (cf. screenshot) or a form to field differents information like Name or location. For the most geeky, the JSON preferences can be edited directly.

Each time a new web page, link or tab is open, the plug-in ask all the renarration available for the current URL, and while match the best renarration. For each renarration, a 'distance' is attributed. A distance notion is something without measuring unit, it's just just some point attribute by the plug-in, based on different rules who can change. For example, a simple equality, or something more complicated, like the similarity between two languages (French and Spanish are closer than Hindi and English for example).

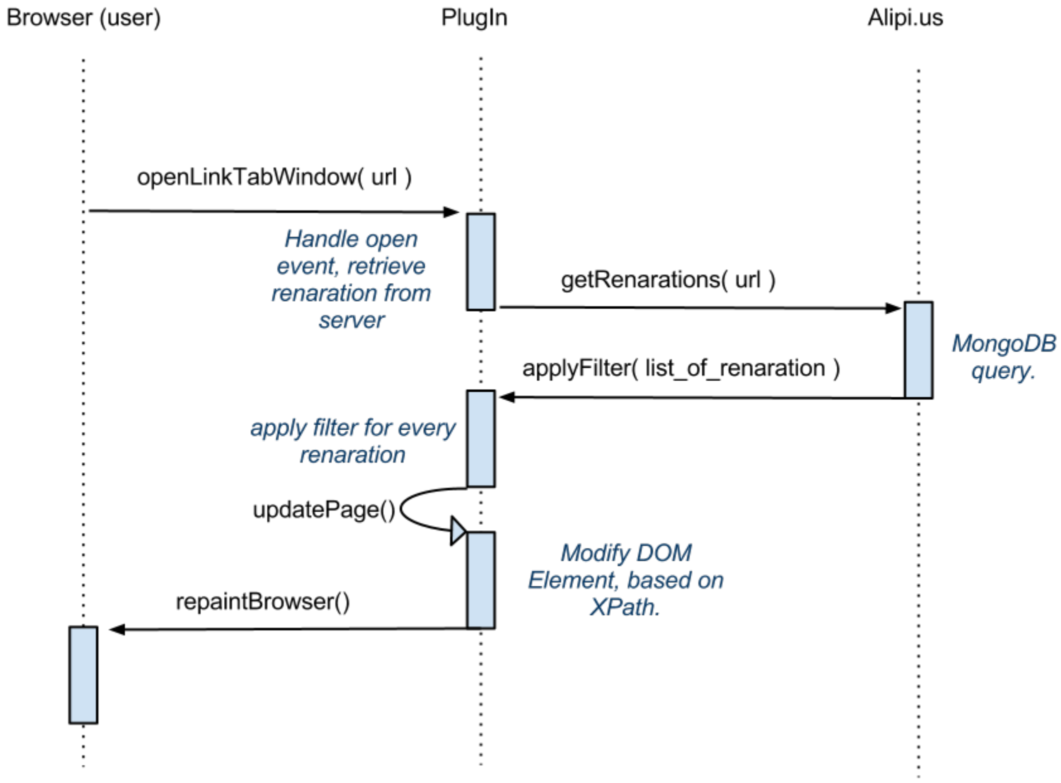
Different rules are applied (languages, friends, location, . . .) and, at the end, the renarration with the smallest distance from the user preferences is chosen. First, we calculate the distance of the renarration languages from the favorite user languages. For example (number means nothing, it is just for the record), if the current renarration is in French- Canadian, and my favorite language is French, the distance will be 20. The distance between French and Spanish will be 50 for example, and 200 for Hindu. The same thing is done for Location. Beside, for friends and family, we only check the equality. If the current author is in our friendlist, the distance stay unchanged, otherwise, distance is increased by 50 for example.

Thanks to that, we can check all the renarration, for all the parameters, and select the ‘best’. Furthermore, the user can edit the “priority’ of each filter, by edited a coefficient.

```
function applyFilter( E: renarrations, list of renarration):best_renarration
begin
    ren : renarration
    prefs <- user.preferences
    best_ren <-null;

    For ren in  renarrations Do
    begin
      ren.distance <- languagesDistance( ren, prefs.languages )
      ren.distance <- ren.distance + fromMeDistance( ren.author, prefs. name )
      ren.distance <- ren.distance + familyDistance( ren.author, prefs. family )
      ren.distance <- ren.distance + friendsDistance( ren.author, prefs. friends
      ren.distance <- ren.distance + locationDistance(ren.location, prefs.locati
      ...
      ...
      if ( ren.distance < best_ren.distance )
        best_ren <- ren
    end
    return best_ren
end
```

Below, a classic use case scenario to retrieve and display the best renaration.


Browser (user)
PlugIn
Alipi.us
openLinkTabWindow(url)
Handle open event, retrieve renaration from server
getRenarations(url)
MongoDB query.
applyFilter(list_of_renaration)
apply filter for every renaration
updatePage()
Modify DOM Element, based on XPath.
repaintBrowser()

Example of user preferences :

```
{ "name": "SkapiN",
  "languages": [
      { "name": "French",  "priority": 1 "category": "read-and-write" },
      { "name": "English","priority": 2,"category": "read-and-write"},
      { "name": "Spanish","priority": 3,"category": "read-and-write"}
  ],
  "family": [{ "name": "francois.boudinet", "relation": "father"}],
  "friends": [
      { "name": "tbdinesh", " trust": "" }
  ], "location": "Bangalore"
```

4.2.3 Utilisation, Installation and Deployment

In this part, i will explain how to setup, configure and use the plug-in, and especially why it's so important for both, users and developer.

For user Alipi is design for print-impaired or disable people, so it's important to provide a good user experience. We need a simple and well documented way to install and use Alipi plug-in.

First, the new user need to download the Alipi-plug-in package (*.xpi) from somewhere, like alipi.us, or by asking Servelots. When it's done, the user has to add this new package to Firefox :

Firefox > Add-On > Tools for Add-on (top-right icon) > Install from file > Select your alipi *.xpi file

Then, restart the browser and enjoy the full Alipi experience!

Simple and easy.

For Dev Now user side is done, it's important to do something for the developer. Create some documentation and manual. Create a new XPI paquet each time we want to modify and test is really painful. Open the browser, flush cache mechanism, reload web page etc...Hopefully, XPI is not the only way (but the easiest) to add a plugin.

First, we don't want to use the default firefox profile for severals obvious reasons, like protect your usual user experience, or if you plug in is bugged. Furthermore, we need to configure firefox to flush all caches each time. The last reason, we want to keep our documentation, Google, mail, etc.. open all the time.

I will explain how to setup and configure a good work environment. Create a new firefox account (linux)

1. You need to tell Firefox to add a new extension. This is can

```
firefox -P test -no-remote
```

-no-remote : do NOT use current firefox profile (default by default), so you can have two different firefox session open in the same time

test : new profil name

Note : If you want to open Firefox with you default settings, you can still use firefox -P default -no-remote

Setup your new firefox account (linux)

We need to disable firefox Cache and XUL cache, useful for dev and test without any problems or troubles :

Open prefs.js file :

```
      cd ~/.mozilla/firefox/*.test/prefs.js
```

Add new attributes :

```
user_pref("nglayout.debug.disable_xul_cache", true);
                        // disable XUL Cache
user_pref("nglayout.debug.disable_xul_fastload", true);

user_pref("javascript.options.strict", true);
                        // enable debug output
user_pref("javascript.options.showInConsole", true);
                        // enable debug in console
user_pref("browser.dom.window.dump.enabled", true);
                        //  enable console dump
```

be done by adding a specific file into firefox setting folder. This file should be named by the extension id (check install.rdf and chrome.manifest), and contain one string which is the absolute path to your extension folder. Be careful about the folder permissions (crossed or target) Example :

```
      cp alipi@skapinthefourb.fr ~/.mozilla/firefox/*.test/extensions/
```

Your extension folder Your firefox profile folder

2. Now, we can work in your extension folder and start a firefox test session each time we want to see how works the plug-in.

Furthermore, I created a full documentation report for users and developer.

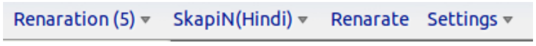
4.2.4 Evolution

When I leaved the project, we decided to not use LastUser, because it doesn't really match our goals. And instead, create our own user manager system. The purpose is the same but by creating our own system, we can do something more precise, and we keep the control of the whole Alipi framework.

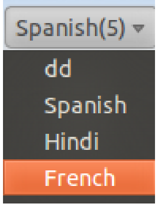
The facebook and twitter part of the plug-in are done, but the web Service using the manager system is still under construction. So the main part of the plug-in is done (alipi functions), but the social aspect is not finished, especially the web service part. The main reason is because someone else was in charge of this.

4.3 Screenshots

A small overview of the plug-in, how it looks like.
The Alipi-tool bar, compound of buttons and menu.
Renaration(5) shows how many languages are available.
Skapin(Hindi) shows the current author(language) used by the pug-in.
Renarate allows the user to re narrate the current web page.



Alipi Plug-in toolbar

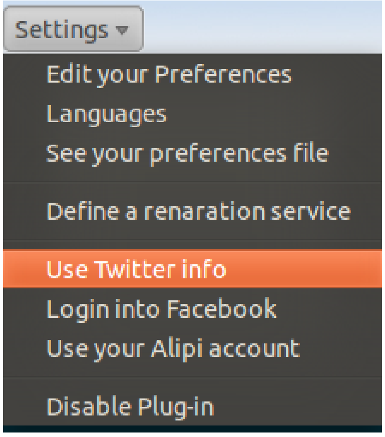


Languages available

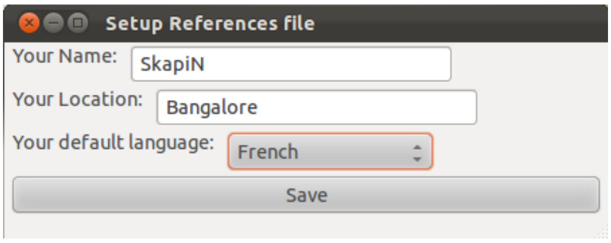
A settings overview. The user can edit several information, disable/enable the plug-in or use his twitter or facebook account to retrieve his friends and family into his preferences file (still not available) :

A form pop-up shown when the user want to edit his preferences :

Language manager. The user can add easily new language, setup the priority or delete an old language.



Settings Menu

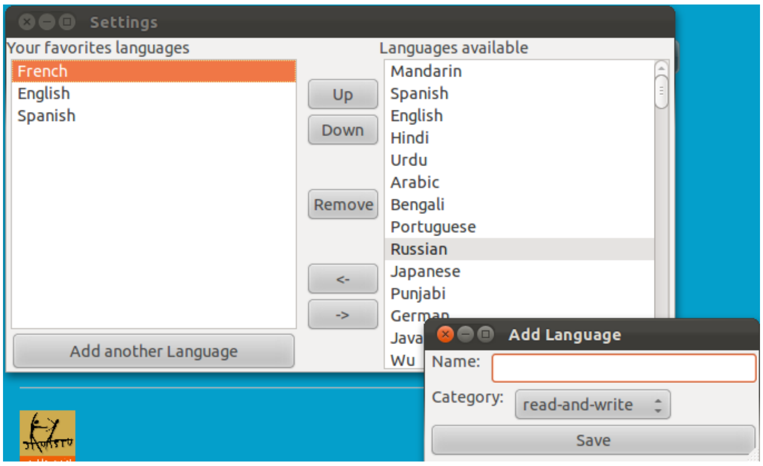


Your Name: SkapiN

Your Location: Bangalore

Your default language: French

A way to edit user preference



Name:

Category: read-and-write

Favorite languages manager

Chapitre 5

Conclusion

During this three-month internship, I worked on severals innovative project to improve the web accessibility. I discover a new side of the software engineering, the importance of matching the software to targeted people.

I chose this internship because of severals reasons, the location (India), the subject and use new technology (RDF,XUL,..) and because the company is opensource and Non-profit. This experience will surely help me go about the end of my studies and my career with a different point of view. In a personal aspect, it was very enriching to witness the way the company worked.The fact that it was based in India played a role in how they tackle the print-impairment issue and how they are working. Secondly, the composition of the persons involved in the project was interesting. The developing team is of course composed of computer scientists, but a lot of diverse people are affiliated with the project in some ways. Sociologists, craftsmen, students, teacher, photograph or even artists, they all see the accessibility issue with a different perspective and their influence on the work of the company is very beneficial.

In a technically way, it learnt new technologies and way of think. So now I master technologies like JQuery, Python, RDFa, XUL. I was in the heart of the design process, I learned a lot about project management and teamwork.

Now I know better my career and I will be able to find a job that meets my expectations and needs.

Chapitre 6

Annexes

6.1 Liens

- *http ://xulfr.org/*
- *https ://developer.mozilla.org/en-US/docs/XUL*$_R$*eferencehttp ://en.wikipedia.org/*
- *http ://www.w3schools.com/js/default.as*
- *http ://www.siteduzero.com/*
- *http ://diveintohtml5.info/*
- *http ://web-semantique.developpez.com/tutoriels/jena/arq/introduction-sparql/*
- *http ://notes.3kbo.com/rdfa-tutorials*
- *https ://www.mozilla.org/fr/plugincheck/*
- *Previous Report and paper about Alipi.*