ENSEIRB-MATMECA
Filière Informatique
33402 Talence

Servelots Infotech
3354 KR Road
Bangalore, 560070
Karnataka, India

Improvement of the accessibility of a web platform

Internship report

Johanna ROYER
2009-2010

Internship tutor (Servelots) : TB Dinesh

Internship tutor (Enseirb) : M. Aymeric Vincent

Abstract

Servelots is a small enterprise that develops an open source software for generating web sites.
It wants to improve the accessibility to 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 the employees of Servelots for their warm welcome in their team and their help. I would also like to thank all the different persons I met in the framework of this internship who showed me that work was more than the technical aspect of it.

I especially thank M. TB Dinesh for giving me the opportunity of this job and for his availability.

I also thank M. Aymeric Vincent for his useful comments over the course of my internship and all the teachers of the ENSEIRB-MATMECA for the quality of the education and the knowledge I acquired from them.

Table of Contents

Illustration Index

Introduction

Out of a permanent concern for accessibility one of Servelots' main goal is to improve the access of the web to as many people as possible, in particular to print-impaired. Print-impairment is a little known sort of disabilities that designates people who can see but not read. In a country like India, this population tolls millions of persons so it is a big issue in the country and for the people in Servelots.

This issue is very poorly addressed in the existing web-guidelines for accessibility. So one of the company's aim is to discover new ways of improving the access to the web to print-impaired. This will be the subject of this internship.

The internship can be divided in two unequal parts. The first days were dedicated to the handover of the software developed at Servelots. The aim was to get familiar with the environment I would be working on later. However, after a month or so, a completely new version of it was released so the learning process was repeated. On this occasion, a few bugs were fixed that will not be mentioned in this report as they have nothing to do with the actual project. This work was really important to get the hand of the code and helped me understand the working process of the team.

The second, and most important part of the internship was dedicated to the accessibility project. This project is mainly a research project as not much informations are available on the subject of print-impaired. The ideas on which we oriented the project will be detailed in this report.

I will first describe the context of the internship : the company and their software. Then more insight on what is accessibility what already exists to improve will be given. Finally, I will describe the work accomplished during the time of this internship and give some open leads that hopefully will serve some future work on the subject.

1. Presentation of the company

1.1. Geographical context

Bangalore is the capital of the Indian state of Karnataka. Located on the Deccan Plateau in the south-eastern part of Karnataka, Bangalore is India's third-most populous city and fifth-most populous urban agglomeration. As a large city and growing metropolis, Bangalore is home to many of the most well-recognized colleges and research institutions in India. It is known as the *Silicon Valley* of India because of its position as the nation's leading IT exporter.

1.2. Presentation of the company

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.

It develops and distributes its software named Pantoto. Pantoto is a Free and Open Source tool set 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.

The company is headed by Asha Prasannakumar and has the full-time services of Dr. T.B. Dinesh in the capacity of Technical Director. The developing team is composed of 3 engineers who also receive the support of several external developers interested in Pantoto's future.

Servelots has a social responsibility arm named Janastu. Janastu provides IT services for the voluntary and civil society organizations. It focuses on community empowerment projects by helping organization customize the Pantoto software for their needs.

Diverse people are also affiliated with Servelots and Janastu. Not only the persons they work with, but artists, craftsmen, volunteers as well. We will draw some conclusion about their influence on the work of the company later in this report.

1.3. Presentation of Pantoto

1.3.1. Functional description

Pantoto is a highly flexible and extensible information structuring system. It is meant to be a community software - configurable, free and Open Source - that can help capture, maintain, and analyze data in a simple and sustainable manner, without the need for continuous technical support.

Pantoto model is especially useful in sectors such as social development, schools, and small and medium size organizations. These organizations work directly with beneficiaries, capture, create and work intensively with data, information and knowledge. The dependency on technical professionals to build information management systems causes delays. The cost of hiring such professionals and related resources usually makes the effort unsustainable. So it is important that they can create, maintain and deal with information management systems by themselves.
Based on these factors, if IT is to be used, these organizations have to be empowered to:

- Take the design, implementation and maintenance of an information management system for granted and not necessarily have to acquire IT skills or hire IT professionals

- Redesign or evolve the system based on changing information needs

- Build solutions that are easy-to-use, affordable and quick to deploy

- Use the same process across projects and in different areas

- Manage the participatory creation of locally relevant and locally created content.

1.3.2. Technical description

Pantoto uses Web-technology -- software runs on a server and people can access it through a Web-browser. Pantoto is installed on a server in an organization, or on an Internet host, and people use only a Web-browser to architect the needs of a community of interest. Typical community needs include provisioning for delegation of roles, collaboration, and creating the space and forms for disseminating and gathering information.
The latest version of Pantoto was reimplemented with Web 2.0 tools and infrastructure. The current approach is to use Django, a high-level Python web-framework, for its strengths. It allows to create templates, controller and forms. Then MongoDB for a flexible object model and jQuery and jqGrid for the User Interface (UI) elements.

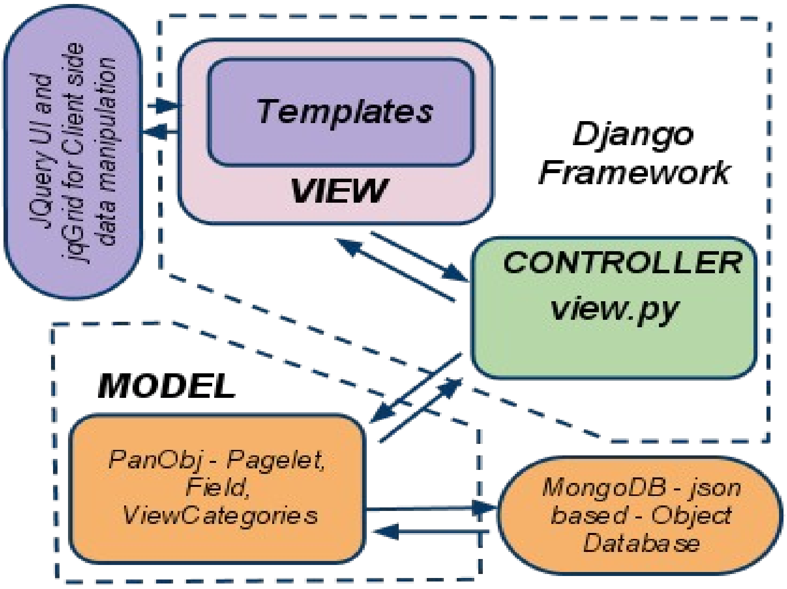


Illustration 1: Overview of Pantoto architecture

1.3.3. Functional architecture

At the core of Pantoto is an entity called a pagelet. It can be viewed as an extensible form which can be authored by multiple authors to create a continuum of content. It's likely that the pagelet is part of a work flow where authors develop the content at various phases as the relevant content itself evolves.

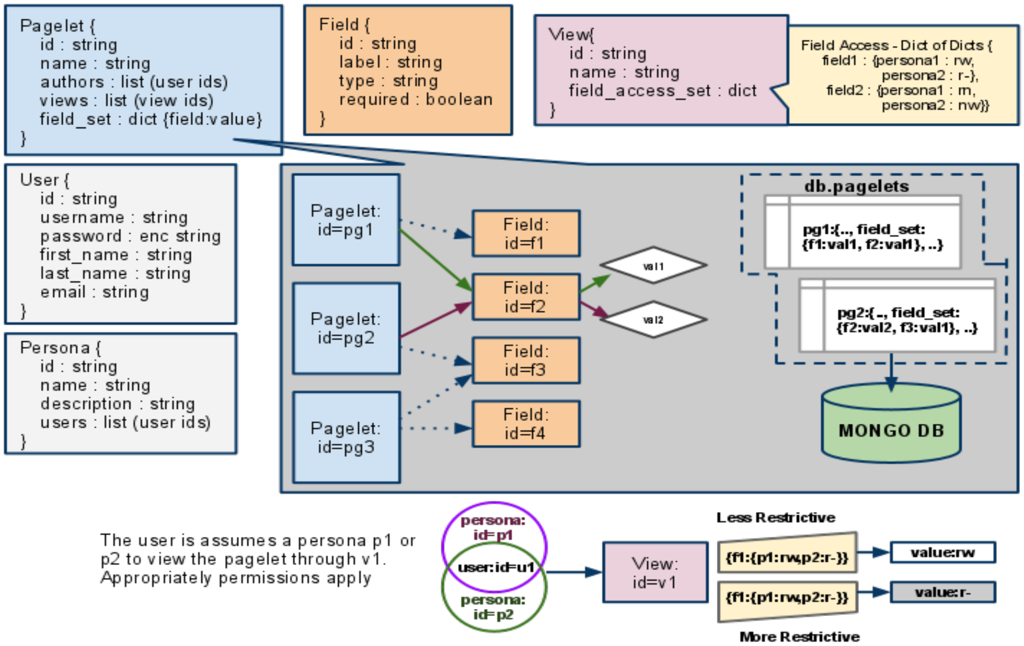
We see this as very useful in many scenarios, to mention a few:

- A school admission form (an empty pagelet created by a school admin), can then be authored by a parent and subsequently extended and authored by the school official to add details such as admission ID

- A complaint form for the electricity board (an empty pagelet available on the board's website) can then be authored by any consumer, subsequently containing follow-up information added on by the electric company officials

- A women's entrepreneurship network creates an FAQ for sharing legal advice (an empty pagelet created by any member of the group) where many team members just extend the form by answering different questions and add as comments their experiences during pertinent legal procedures.

Pantoto is a fairly simple architecture with these basic building blocks:
Users can assume a persona and manipulate (add, edit, delete) a part of or a whole pagelet from their limited or whole view of the fields in the pagelet based on the permissions that the view allows the user on the pagelet fields. A more formal definition of the non-trivial entities is below:

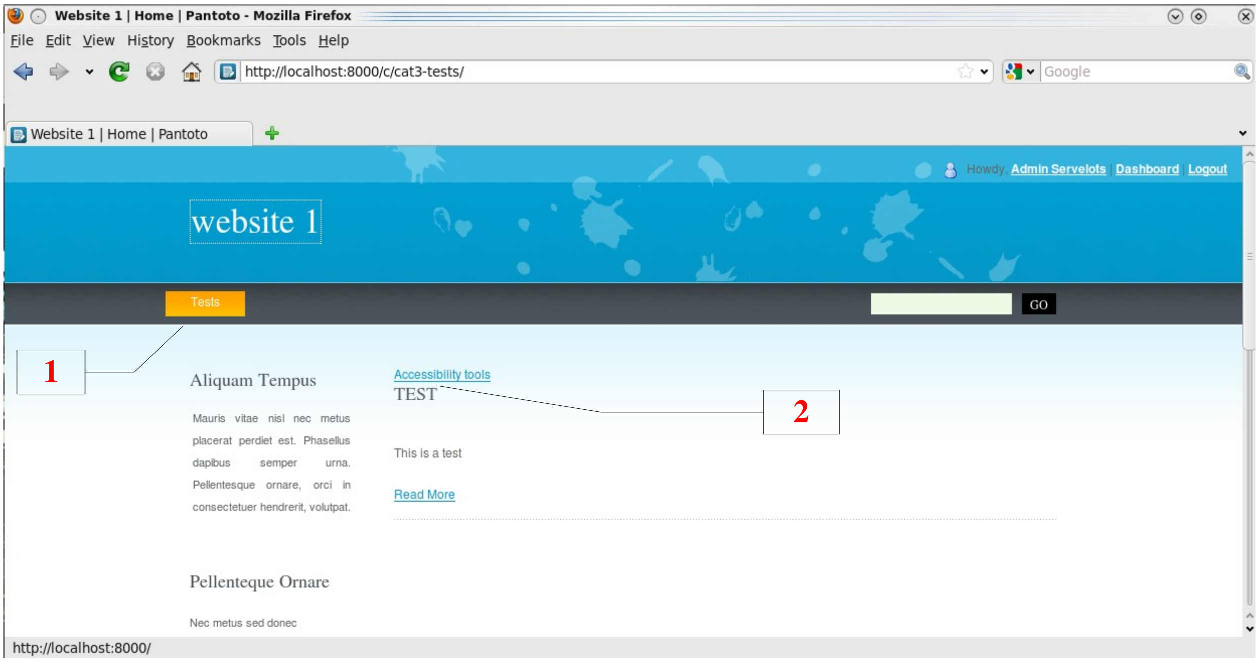


```
Pagelet {
    id : string
    name : string
    authors : list (user ids)
    views : list (view ids)
    field_set : dict {field:value}
}

Field {
    id : string
    label : string
    type : string
    required : boolean
}

View{
    id : string
    name : string
    field_access_set : dict
}

Field Access - Dict of Dicts {
    field1 : {persona1 : rw,
              persona2 : r-},
    field2 : {persona1 : rn,
              persona2 : nw}}

User {
    id : string
    username : string
    password : enc string
    first_name : string
    last_name : string
    email : string
}

Persona {
    id : string
    name : string
    description : string
    users : list (user ids)
}
```

The user is assumes a persona p1 or p2 to view the pagelet through v1. Appropriately permissions apply

Illustration 2: Overview of Pantoto entities

From the pagelets, one can generate a customized website. The basic structure will be the same for every site since it is based on HTML templates. The categories created will become the different tabs of the horizontal menu. In each category, the pagelets belonging to them will appear as articles.


Website 1 | Home | Pantoto - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8000/c/cat3-tests/
Google
Website 1 | Home | Pantoto
Howdy Admin Servelots Dashboard Logout
website 1
Tests
GO
1
2
Accessibility tools
Read More
http://localhost:8000/

Aliquam Tempus

Mauris vitae nisl nec metus placerat perdiet est. Phasellus dapibus semper urna. Pellentesque ornare, orci in consectetuer hendrerit, volutpat.

TEST

This is a test

Pellenteque Ornare

Nec metus sed donec

Illustration 3: Website generated with Pantoto

1 : Category

2 : Pagelet

2. Accessibility

2.1. Definition of accessibility

The term accessibility describes the extent to which a product or device is accessible by as many people as possible. Usually, it is used to described the "accessibility" of a product to people with disabilities. Some general examples would be audio signals at crossings for blind people, wheelchair ramps, etc. In this report we will focus on a special type of accessibility : web-accessibility, and special types of disabilities : visual and cognitive disabilities.

Web-accessibility means that people with disabilities can use the Web. Not only can they use the web, but ideally they can enjoy as much features as people without disabilities. They can navigate, interact with the Web, and also contribute to it. Millions of people have disabilities that affect their use of the Web. Disabilities can be visual, auditory, physical, speech, cognitive or neurological ones. All of them affect people in different ways and to different degrees.

In this part we will list the different disabilities we will be focusing on during the project, that is visual and cognitive impairments, and how they affect the use of Internet.

2.1.1. Visual disabilities

Blindness

Blindness involves a loss of vision of both eyes.
To be able to access the web, many people who cannot see use screen readers, software that reads text on the screen and output the information with a speech synthesizer or with a Braille display. Alternatively, they use a text-based browser such as Lynx or a voice browser.
Even with these tools, blind persons can encounter barriers on the Web. For example, images that do not have alternative text, tables that do not make sense when read serially or forms that are poorly labeled.

Low vision

People with low-vision, or partially-sighted, need extra-large monitor and increase the size of the font and the images to be able to use the Web. They can also use screen magnifiers and screen enhancement software that can be adapted to their vision requirements.
However partially sighted persons can have some trouble to read a web page which font size do not change easily or that have poor contrasts.

As we will see in the next section, a lot of web-accessibility guidelines already exist for partially and non-sighted people. So we will keep in mind those guidelines all along the development of the project, but these users will not be our main targets. Instead we will focus on print-impaired users, who represent a very significant part of the population in India.

2.1.2. Print impairment

Print-impaired refers to people who can see clearly but who are not comfortable with text. It can designate illiteracy, dyslexia, or any other kind of cognitive disability that prevent someone to understand a written text. They can use the same tools as a blind person such as screen readers and voice browsers.

However, here we need to take in account the social context. In India, several millions of people are print-impaired. The main reason being the unequal access to education poor and rural persons receive.

This incidentally means that print-impaired users do not necessarily own a personal computer. So they might not be able to install any software they need to read correctly a website. The only thing they can probably do is change the settings of the browser. In most of the cases it is enough to magnify the letters, but not for a read-aloud feature.

Another thing, maybe less obvious, is that a screen-reader can be very frustrating for a person who can see. Indeed, the way to go through a page with a screen-reader is by using the keyboard. For blind user it is of course the only possible way, but for a print-impaired it can be really annoying to navigate. Such users can usually use a mouse and would prefer to click on a specific section in order to hear the content.

During this project we will focus mainly on this category of disabilities and on how to make a website accessible for them.

2.2. Existing web-accessibility guidelines

In this section we will describe briefly the web-accessibility guidelines that already exist. Most of them are standards of the W3C (World Wide Web Consortium) used by many websites. All of these guidelines are for non to partially sighted users. They are not our target users but we will have to keep these guidelines in mind all along the project as we still want websites accessible by the more people possible.

Here is a non-exhaustive list of the main standards in use on the Web.

Text-equivalents

Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language. Non-text content include audio and video but also informations conveyed by color changes for example.

Color contrast

The objective of this technique is to make sure that users, and especially color blind people, can read text that is presented over a background.

Structure and content

Create content that can be presented in different ways without losing structure and information.

Language

Identify the language used in the web page so that speech and Braille outputs are consistent with the content.

Navigation

Provide clear navigation mechanisms.

3. Accessibility project

3.1. Definition of the project

As we said in the previous section, one of the main concerns of the company is accessibility. After applying all the necessary guidelines to improve the accessibility of their websites to non-sighted and partially sighted users, their new target users are print-impaired.

Of course they first want their generated websites to be accessible for these users, but they are also aiming at providing new standards applicable to any site on the web. This will be the subject of this internship. As it is a relatively new topic, no real guidelines exist at the moment and the three months of internship were mainly dedicated to researches. We analyzed the different web-technologies in use to see how they could serve our purpose and tried to find some new leads. The goal really was to explore new paths and not necessarily to produce a lot of code. In this sense, we did reach our objectives and have some beginnings of solutions. We will describe in the following sections all the tests achieved and solutions foreseen.

The underlying theme is that print-impaired users are not blind and therefore do not necessarily find screen-readers very adapted to their need. Like we said before, the use of a screen reader with the keyboard might be frustrating. They might instead want to click on a section and hear everything that is related to it. The section could for instance be an image, which would "speak" to a print-impaired person and that would have some text related to it. Let us take the example of a user's profile page on a website. It is usually composed of a picture of the user followed by some description of this person. But the page will also contain some informations on the website that are not relevant for the profile description. A print-impaired could, in our logic, be able to click on the picture and hear only the related informations.

Logically, the first focus will be on the semantic associations within a page. We need to find a way of describing the relationships between the different elements of a web page.
Once this will be done, we will try to integrate a Text-To-Speech tool to read the sections related to a specific element. The methods to mark the semantic associations and the integration of the Text-To-Speech in Pantoto will be described in the next sections.
Then we will suggest some ideas to generalize the accessibility to print-impaired to other websites.
Finally we will discuss how the new multimedia-web technologies could be used to improve the accessibility of any website.

3.2. Semantic associations

One way to improve the accessibility of a site would be to relate the different elements of a page between them. So we need to introduce some semantic associations in the HTML code.
To this end, we looked at the different technologies existing to add such informations to HTML.
The first evident solution that came up was RDFa (Resource Description Framework -in- attributes) which is a W3C recommendation that allows to add some attributes to XHTML to embed rich meta data to Web documents.
Another solution we found was to use the class attributes of HTML to some other extent than what they are supposedly for.

In the following sections we will describe these two methods and then compare them in regard of their use in Pantoto.

3.2.1. RDFa

RDFa which stands for Resource Description Framework – in – attributes, is a W3C recommendation that allows to add some attributes to XHTML to embed rich meta data to Web documents. The RDF data model mapping enables its use for embedding RDF triples within XHTML documents, it also enables the extraction of RDF model triples.
RDF is the way information is expressed semantically on the web. It was created so that computer programs can start to understand the data's structure of a web page.
It is constituted by triples which are subject-predicate-objects statements. This lets machines understand human knowledge statements. When many triples are aggregated they are stored in what's called a "triple store." By querying a triple store, we can learn information that might otherwise be hard to gather by just browsing the web with our eyes.
RDFa is a way of expressing RDF triples inside XHTML pages using span tags.

```
<html xmlns="http://www.w3.org/1999/xhtml"
     xmlns:foaf="http://xmlns.com/foaf/0.1/"
     xmlns:dc="http://purl.org/dc/elements/1.1/"
     version="XHTML+RDFa 1.0" xml:lang="en">
<head>
     <title>John's Home Page</title>
</head>
<body about="http://example.org/john-d/#me">
     <h1>John's Home Page</h1>
     <p>My name is <span property="foaf:nick">John D</span>
          and I like <a href="http://www.neubauten.org/"
          rel="foaf:interest"
          xml:lang="de">Einstürzende Neubauten</a>.
     </p>
     <p>My <span rel="foaf:interest"
          resource="urn:ISBN:0752820907">favorite book is
          the  inspiring <span
          about="urn:ISBN:0752820907"><cite
          property="dc:title">Weaving the Web</cite> by
          <span property="dc:creator">Tim Berners-
               Lee</span></span>
          </span>
     </p>
</body>
</html>
```

Example found at http://en.wikipedia.org/wiki/RDFa

This example uses two different vocabularies : dc (Dublin Core) and FOAF (Friend Of A Friend). Specific vocabularies allow to describe some specific semantic relationships between the elements. The FOAF vocabulary describes what can be related to a person, like his name, interests, friends, etc. The Dublin Core vocabulary allow to describe resources such as books, videos or images. With this we can relate a book to its author, its title, etc. In this example, we semantically tag the name of the person the page is about and his interests. One of his interests is a book; we semantically tag the title, the creator and the ISBN reference of this specific book.

3.2.2. HTML tagging

The HTML tagging method consists of using simple HTML attributes to add some sort of meta-information to the content of a web page. When looking for a simple solution to relate objects in a page, we thought of the CSS classes. A same style can be applied to various elements, possibly completely unrelated, by adding the same class attribute to these elements. And this is very similar to what we wanted to achieve.

The only idea missing is to relate some elements to another element like an image or a title. To this end, we can use the id attribute also natively present in HTML. Following this method, a group of elements with the same class name can be related in a semantic way to a unique element in the page.

To illustrate this method, let us take the same profile page example.

```
     <html>
     <head>
          <title>John's Home Page</title>
     </head>
     <body>
          <h1 id='john'>John's Home Page</h1>
          <p class='john'>My name is John D and I like
               <a href="http://www.neubauten.org/">
               Einstürzende Neubauten</a>.
          </p>
          <p class='john'>
               My favorite book is the inspiring <cite
               id='fav_book'>Weaving the Web</cite>
               by <span class='fav_book'>Tim Berners-
               Lee</span>
          </p>
     </body>
     </html>
```

Same example with HTML tagging

With this example, we can notice that we loose a lot of informations. For example the program that will parse this code will not know that Tim Berners is the author of "Weaving the Web", whereas with the first example it is known. But we still have the information that Tim Berners is related to Weaving the Web.
In a sense, it is less restrictive. We can relate the title of the page to a full paragraph (or more) like we do here.
Also in this method, the elements do not need to be necessarily regrouped in a <div>. We do not have to care about the DOM structure at all to find all the elements with the same class. Thus, they can be spread out in the page without adding any complication for the developer.
Moreover, HTML allows multiple classes for one element. So it is very easy to relate one element to several elements in the web page. This can be very useful for nested elements for example. However, one must be careful when using several classes as it can lead to confusion in the design. If different styles are applied to different classes held by the same element, it can induce conflicts.

3.2.3. Comparison

Comparing these two methods is hard. They are at the same time, quite similar and very different.

They both use HTML attributes to embed meta-information in a web page, but the original aim of both of them is different. The first one was created so computer programs can have a better understanding of the Internet content and the second is more for the developer to improve the quality of his website.

If we focus on simplicity, the "HTML tagging" method has the clear advantage.
First of all it does not require any new knowledge, any web designer is capable of adding class and id attributes to a web document. The implementation of the RDFa solution is not much harder, but the developer must find the appropriate vocabularies to describe all the relationships. And if none matches what he needs, then he will have to create a new vocabulary, if not several.
It also allows for more flexibility. Indeed, with the "HTML tagging", elements can be placed anywhere in the page. More importantly, they do not need to be related in the DOM structure to be related to each others. Whereas with RDFa, they need to be grouped in some way so that a parser can understand the relationship.

The main advantage of RDFa resides in the standardization of the vocabularies. It allows a computer program, and possibly directly the browser, to know what kind of relationships the elements have between them. With the first method, a program is just able to understand that some elements are related. With RDFa the program has access to the different vocabularies used and understands the nature of the relationship.

We do not really need this feature in order to improve accessibility for print-impaired. Given that the text will be eventually spoken by a program, people who can see will have the same information as literate users. They do not need more details about the semantic relationships between elements.
However, this would be interesting for non-sighted users. A screen-reader could analyze the RDFa annotations of a web page and really improve its readability for blind users.
This is not the purpose of this project, so we chose to use the HTML tagging method. Besides, it is possible to use both methods at the same time. If we choose to use the class and id attributes, we can later add RDFa annotations to be understood by a screen-reader for example.

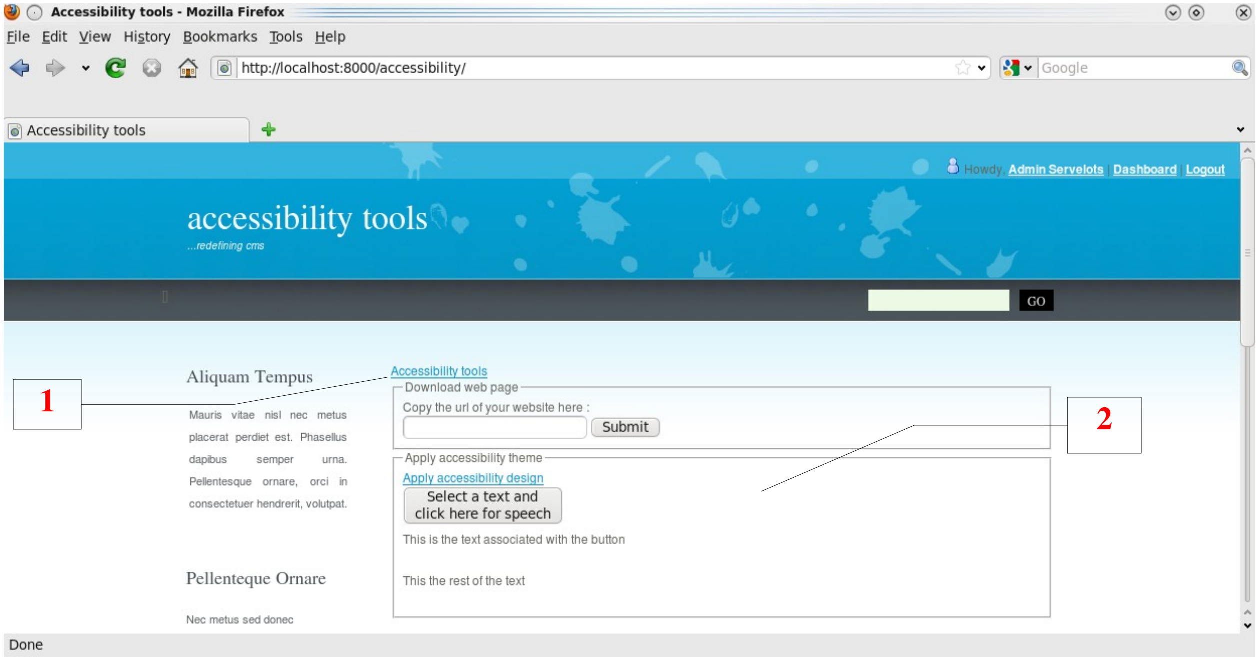
3.3. Towards a generalization of accessibility

As we said, the goal of this project is to find some new approaches in making websites accessible for print-impaired. Semantic associations were the starting point to work on a few leads. We started to think of some ideas to adapt to a particular site, namely Pantoto. Then we started to think more broadly and found some leads to generalize these ideas.

3.3.1. Work on Pantoto

Creation of an accessibility environment

To centralize all the tests we made during this project and the ones that will possibly be made in the future, an accessibility environment has been created. It is accessible from the website interface through a link. For the moment it is mainly for testing, but on the long term it will become a page that visitors can access to enable these tools on Pantoto as well as for any website.


Accessibility tools - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8000/accessibility/
Google
Accessibility tools
Howdy, Admin Servelots | Dashboard | Logout
accessibility tools
...redefining cms
GO
Aliquam Tempus
Accessibility tools
Download web page
Submit
Apply accessibility theme
Apply accessibility design
Select a text and click here for speech
Pellenteque Ornare
Done
1
2

Mauris vitae nisl nec metus placerat perdiet est. Phasellus dapibus semper urna. Pellentesque ornare, orci in consectetuer hendrerit, volutpat.

Copy the url of your website here :

This is the text associated with the button

This the rest of the text

Nec metus sed donec

Illustration 4: Accessibility environment

The link calls an HTML template that was added in the site branch of the code. It is located in the index template so it is visible in the content block of any web page of Pantoto. Eventually it can be moved to the sidebar when the latest will be implemented.

1 : Link to the accessibility environment
2 : Accessibility tools

The following code is a part of the template created to obtain this accessibility environment.

```
{% extends 'site/index.html' %}
{% block title %} Accessibility tools {% endblock %}
{% block extrahead %}
...
{% endblock %}

{% block content %}
<fieldset><legend> Download web page</legend>
<form method="get" action="/accessibility/url">
...
</form>
</fieldset>

<fieldset><legend>Apply accessibility theme</legend>
...
</fieldset>
{% endblock %}
```

accessibility.html

The first line indicates that the template inherits from the index template so we do not need to include all of the elements of the page like the sidebar, menus, etc.

The code written inside {% %} is Django code. For example, `{% block extrahead %}` contains the calls to the scripts and stylesheets ; `{% block content %}` contains the content of the page. The rest of the code is normal HTML code and describes the different elements of the page. The details of the tools will be given later on.

Semantic associations

As we discussed in the previous section, we worked on semantic associations. We said then that we would use the "HTML tagging" method. The goal was to implement it on the whole Pantoto site part, but at the end of the internship this part of Pantoto was still under construction so it is not completely implemented. We did however introduce it in the accessibility environment. This will allow the team to understand how it works and spread it to the rest of Pantoto when it will be ready for it.

Technically, a simple jQuery script has been added.

```
1. $(document).ready(function(){
2. [...]
3. var highlight = function() {
4. var currentId = $(this).attr("id");
5. if(currentId != "") {
6.   currentId = "." + currentId;
7.   $(currentId).addClass("highlight");
8. }
9. };
10. $("h2").click(highlight);
11.  }
```

tts.js

The first line indicates that the following jQuery code will be executed when the web document is ready.
The function `highlight` is executed when a user clicks on a `h2` element. This is just a temporary choice, it can work with any HTML element as long as it is specified here. The `highlight` function fetches the `id` attribute of the element clicked on line 4 and save it in a variable. We add a "." before the variable because it represents a class attribute in jQuery. In the line 7, the function looks up for every element with a class attribute equal to the variable and add a CSS class highlight to them.

```
.highlight {
background-color: yellow;
  }
```

accessibility.css

The CSS class just adds a yellow background to these elements so they are clearly visible to the user. Of course this is just for testing purpose and can be changed later if needed.

The script is called when a certain link is clicked in the accessibility environment. This link `Apply accessibility design` can also work as a bookmarklet. That is, we can drag the link in the bookmark toolbar and call it from there. It corresponds to the lines 19 to 25 of the accessibility.html file. Basically, it adds an element `script` which calls the script tts.js and an element link which calls the accessibility.css stylesheet we need.

Text-To-Speech

The highlighting of the text related is only the first step to a more accessible web page. The goal was to be able to output a speech of this text. To develop a Text-To-Speech (TTS) software is not an easy thing and it would have been a complete project on its own. So at first, we looked for an existing TTS software that we could adapt for our needs. The main constraints were that it needed to be open-source and in a language easily integrable to Pantoto.

After several weeks of research, we found Vozme. Vozme is a web service that transforms some text in mp3. Its main feature is that one can integrate it to their website. It will allow their users to select a text, then click on a button and hear the text read aloud by the TTS software.

The original code to integrate looks like the following :

```
  <script type="text/javascript"
src="http://vozme.com/get_text.js"></script>
<button
onclick="get_selection('en','mlfm');">
  Select a text and<br/>click here for speech</button>
```

Code to integrate Vozme

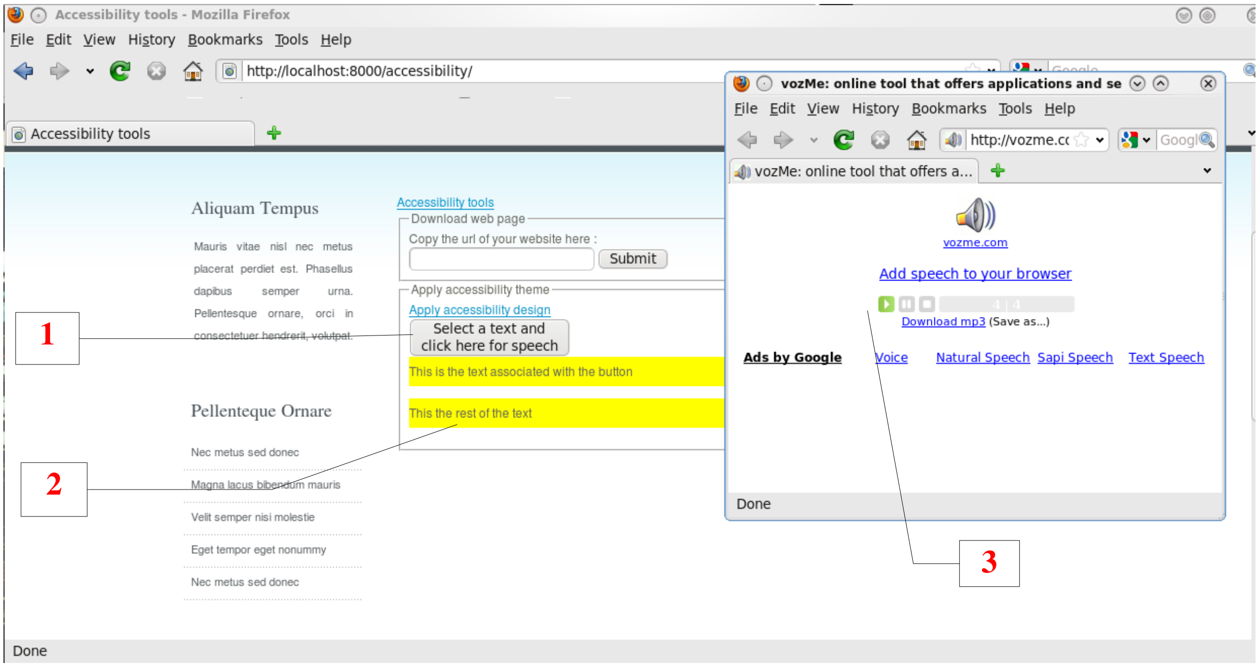
It just invoke a script of a Vozme domain and calls a function with certain parameters when the user selects a text and click on the button.

However, we wanted something more automatic than this. Our users should be able to click on an element and hear the sections related to it by the class attributes. So we got the get_text.js script and modified it a little bit to read what we wanted.

```
1. $(document).ready(function(){
2. function create_form(d,txt,lang,gn,tgt){
3. [...]
4.   }
5.
5. function get_selection(){
     ...
21.   }

22. var tts = function() {
23.   var currentClass = $(this).attr("class");
24.   if (currentClass == "tts") {
25.        var lang = arguments[0] || '';
26.        var gn = arguments[1] || '';
27.        var tgt='voice_'+parseInt(Math.random()*100000);
28.        var d=window.document;
29.        var txt='';
30.        var currentId = $(this).attr("id");
31.        if(currentId != "") {
32.             currentId = "." + currentId;
33.             txt = txt + ' ' + $(currentId).text();
34.        }
35.        $(currentId).addClass("highlight");
36.        create_form(d,txt,lang,gn,tgt);
37.        return false;
38.   }
39.});

40.   $("button").click(tts);
```

tts.js

The function `create_form` creates the window that will contain the player. It takes in arguments the text to be read and some options like the language it is supposed to be speaking in. The original function was `get_selection`. It was fetching the text selected by the user to give it to `create_form`. We adapted this function so it would take the text of the elements related to the element clicked instead. The block from line 25 to 29 initialize the arguments. Then we do the same than in the function `highlight` except that here we get the text of the related elements and feed it to the function `create_form`. The function is called when a user clicks on a button element. But like before, this is only for testing purpose, it can be changed to something more specific.


Accessibility tools - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8000/accessibility/
Accessibility tools
Aliquam Tempus
Accessibility tools
Download web page
Submit
Apply accessibility theme
Apply accessibility design
Select a text and click here for speech
Pellenteque Ornare
Done
vozMe: online tool that offers applications and se
File Edit View History Bookmarks Tools Help
http://vozme.cc
vozMe: online tool that offers a...
vozme.com
Add speech to your browser
Download mp3 (Save as...)
Ads by Google
Voice
Natural Speech
Sapi Speech
Text Speech
Done
1
2
3

Mauris vitae nisl nec metus placerat perdiet est. Phasellus dapibus semper urna. Pellentesque ornare, orci in consectetuer hendrerit, volutpat.

Copy the url of your website here :

This is the text associated with the button

This the rest of the text

Nec metus sed donec

Magna lacus bibendum mauris

Velit semper nisi molestie

Eget tempor eget nonummy

Nec metus sed donec

Illustration 5: TTS tool

1 : Button to trigger the Text-To-Speech
2 : Text related to the button, highlighted when the button is clicked
3 : TTS player opened when the button is clicked

To enable this feature, first the user has to click on the link `Apply accessibility design`. Then in this testing environment, he clicks on the button. It will highlight the related sections as well as call the TTS tool. This last action will open a new window containing the player that will speak out the text highlighted.

For the same reasons previously mentioned, this TTS tool is only implemented in this accessibility environment. However it is ready to be used on the whole site part of Pantoto as soon as it will be ready.
Even if it is working as it is, a few improvements could be done on this part.

For example, Vozme "understands" several languages like English, Spanish, Italian and Hindi. It would be interesting, not to say important, to allow our users to choose between these languages. We could even imagine an automatic solution that will look for the HTML `lang` attribute and call the proper language based on this information.
Another improvement would be to integrate the player in the site instead of calling a new window every time. Indeed this can seem confusing to some users compared to player that they will see as soon as they enter the web page.

3.3.2. How to apply an accessibility theme to any website

After these conclusive tests on Pantoto side, we tried to see how we could use these tools and technologies to apply an accessible design to any website.

The idea considered is relatively simple. Since it is not possible to modify the source code of any page of the web if do not own it, the idea is to get the source code and store it on a dedicated server. Then the user could modify it, that is to say apply the accessibility theme, and then re-access it anytime through Pantoto.

There is however one problem : it is not legally possible with any webpage. The website in question should either be open-source or owned by the user who wants to modify it. This being a know issue, we will only have to be careful during our tests and be very clear about it if we actually introduce it.

By lack of time this feature has not been fully implemented nor tested during the internship. However, the steps to achieve are quite clear. In this section we will describe the phases that have been started and give some details about what is expected for the next ones.

Fetching the source code

This first part has been partially implemented. It consists in fetching the source code of a given URL and save it on our server side.
In the accessibility environment, there is a text box for this purpose.

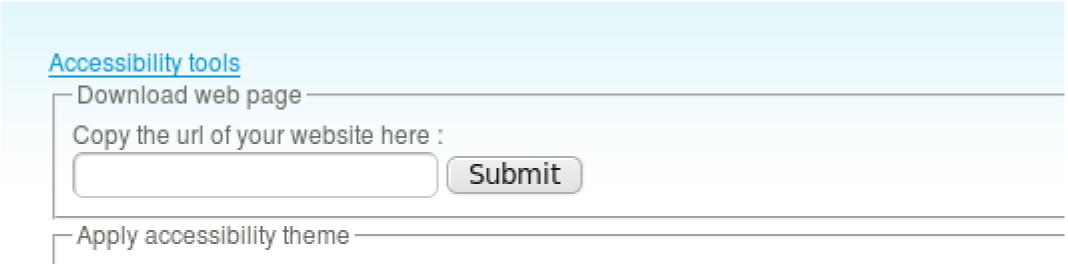

Accessibility tools
Download web page
Copy the url of your website here :
Submit
Apply accessibility theme

Illustration 6: Fetching a source code

The piece of code that calls this action is a part of the `accessibility.html` file introduced in the previous section.

```
<form method="get" action="/accessibility/url">
Copy the url of your website here : <br/>
<input type="text" id="url_client" name="url_client" />
<input type="submit" name="submit_url" value="Submit" />
  </form>
```

accessibility.html

Once the user has submitted his request, a URL is called in the action attribute of the `form`.
This URL is interpreted by Django and calls a certain function `do_get_source` :

```
('^accessibility/url/$', 'do_get_source')
```

This python function gets the content of the text box and execute a system call line 4 to get the source code of the page requested.

```
1. @login_required
2. def do_get_source(request):
3.   source = request.GET["url_client"]
4.   os.system("wget " + source)
5.   return render_to_response('site/accessibility.html',
{},context_instance =
6.   RequestContext(request))
```

views/site.py

This whole action stores the web page in a particular directory of Pantoto code. A possible improvement would be to have a directory by user. This way, each user would only see the web pages they are modifying, it would allow more privacy. Another option would be to have a collaborative environment so users can share some of the websites they are modifying in addition to their private environment.

The next step of this part is to display the fetched code in a rich text editor on a part of Pantoto so the user can modify it.

Modify a rich text editor

This step will consist in modifying a rich text editor in which a source code will be displayed in order to add some semantic tagging to this code. This part has not been started yet, but we will explain in the following what should be done in our idea.

A rich text editor is already used in Pantoto in the admin interface to write the Pagelets. A couple of options should be added to it.

First, an option `id`. This option would allow a user to mark an element to which some other elements will be related to. Second, an option `class` that would allow the user to mark the related elements.

The use case could be the following :

When a user clicks on it, he first chooses a unique name for it. The rich text editor should be able to check if the id already exists in the page, and if so, change it. Then the user clicks on the element of his choice. Maybe the elements able to receive an id attribute should be limited to a few HTML elements according to the script previously written (tts.js). Then this id is saved until the end of the operation.
The user can then click on as many (and any) elements that he wants to relate to the first element. When he is done with the tagging, he clicks again on the `id` option that will have become `end`.
This operation could be repeated as many times as the user wants to. When he is definitely over, the modified code would be saved in the same directory as the original one. That is, in a personal or shared directory according to the decision that will have been made for the first step.

This scenario brings about a few issues that we did not have to time to think about yet. For example, how to cancel such an action without detract the original code ? Or how to relate some elements to another element which already have an id ? Indeed, an HTML element cannot have several id attributes so this could be an issue.

Apply accessibility theme

The last step would be to open the web page modified with the rich text editor and apply the accessibility design on it. This part is actually already implemented and was described in the section 3.3.1.
The end of the use case would be :

The user goes to his profile on Pantoto. In the accessibility environment, all the web pages he modified are available along with the pages other users have shared. He could choose one and apply the accessibility design on it with the bookmarklet. This will enable the highlighting and Text-To-Speech tool integrated.

3.4. Multimedia websites

Another part of the work on accessibility is to think of how websites can benefit from the new webs standards and multimedia technologies available in terms of accessibility.

New web standards including HTML5 and CSS3 are now widely used to add visual features and interactivity to websites. The question is how can we use them to make a website accessible to the most people possible. In these sections we will focus mainly on overcoming the text barriers even if other handicaps can obviously benefit from these technologies.

3.4.1. CSS3

The new version of CSS offers a lot of exciting possibilities. A lot more features are now feasible without using JavaScript for example. Web designs can really benefit from these visual effects. But hey are not the only target.

One of the new CSS3 module that seemed really promising for accessibility was the speech module. This module allows the designer to manage the voice output of a screen reader. The author of a web page can vary the different characteristics of synthetic speech such as voice type, frequency, etc.

It is interesting in the sense that the structure of the document can be designed to be read out.

Unfortunately, this module is not very largely used and its implementation has stopped since 2004. If however it was to be "relaunched" it would be very useful for our accessibility issues. We could imagine users being able to modify the CSS of a page on the spot, and therefore handling the voice properties themselves and adding speech output where they are interested for instance.

3.4.2. Videos

The more obvious alternatives to text are audio and video outputs. The relatively new version of HTML, HTML5 makes it easier to embed such files into websites. Indeed instead of the old tag <embed>, two more tags are available : <video> and <audio>.

One of the advantage of these tags is that the user is not supposed to install any new plugin for his browser, they should use the codecs natively handled in the user's browser.

The video tag also brings a unified API. So whatever codec the user is using, the same methods are available. However it is possible to define our own controls instead of using the browser's ones.

```
<script>
var video = document.getElementById("video");
</script>
<p><button type="button"
onclick="video.play();">Play</button>
<button type="button" onclick="video.pause();">Pause</button>
<button type="button"
onclick="video.stop();">Stop</button></p>
```

Padma

As part of this analysis of the new web-technologies, we participated to a workshop organized by an association that handles an online archive of text-annotated video material.
They are exploring news ways to exploit video materials that could be very interesting for Pantoto and for accessibility in general. Their idea is to add layers of annotations to a video. That is different users can add their own interpretations of a video or just a part of the video. The text can be linked to some time lines, which is useful for subtitles for example.

Not only this can be used for subtitles, but it can also be very useful for print-impaired. A piece of text could be related to a special portion of a video. Even if the text is not exactly a subtitle of what is said in the film, at least they would have an idea of what it is about.

While we were at this workshop we worked on the idea of translations. We will talk furthermore about this work in the next section.

3.4.3. Translation

One accessibility issue that concerns more than print-impaired people is the language barrier. It happens to everyone not to be able to read a website because it was written in a different language. This problem is especially true in India since every state has its own language. In this section we will discuss some solutions that already exist for translation issues and we will try to find a more interactive solution to them.

Multi-languages websites

One solution, obvious and in use in a lot of official websites, is a multi-language website. It allows users to be certain of the translated content, since it comes from the creator of the website. However, it is a very static solution. Whenever the content of the main site changes, the web master needs to change the translated pages corresponding.

This solution is good for commercial websites for example, where the content will not change everyday. But for community websites where several people can add content or where visitors can interact, this is not enough.

Translation on the spot with an embedded tool

To get around this, translation tools were created. One of the most famous is the Google Translator Toolkit. First, it allows a web master to include the toolkit directly on the website. This way, users can choose their language the same way they would on a multi-language website. Only this time it is translated dynamically. There is no need to change the translated versions of the site every time something is added.

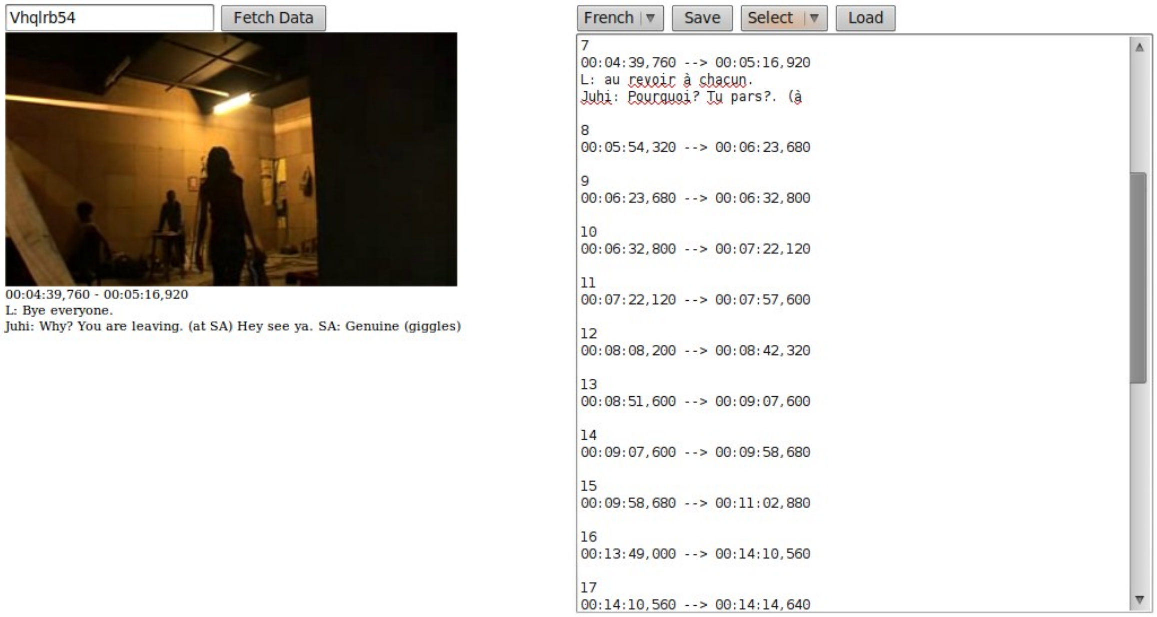
This toolkit also offers another feature, for the visitors to use on any website. People can add a bookmarklet in their browser to be able to translate automatically any website they are visiting.

However the quality of automatic translations is usually not very good. It is especially true when it comes to non-widely spoken languages. It is often good enough to understand the global meaning of the text but it will in most of the cases miss the details.

Adding a translation of a given section

Even if the translation tools become more accurate in the future, it can happen that language is not the only barrier. For instance, people might need cultural references to understand.

Let us come back to the feature we worked on for Padma that could be very interesting for Pantoto and in general for any web site. We implemented a script that allows a user to translate a video on the fly. Within the given time, what we developed offers some interesting possibilities.



00:04:39,760 - 00:05:16,920
L: Bye everyone.
Juhi: Why? You are leaving. (at SA) Hey see ya. SA: Genuine (giggles)

```
7
00:04:39,760 --> 00:05:16,920
L: au revoir à chacun.
Juhi: Pourquoi? Tu pars?. (à

8
00:05:54,320 --> 00:06:23,680

9
00:06:23,680 --> 00:06:32,800

10
00:06:32,800 --> 00:07:22,120

11
00:07:22,120 --> 00:07:57,600

12
00:08:08,200 --> 00:08:42,320

13
00:08:51,600 --> 00:09:07,600

14
00:09:07,600 --> 00:09:58,680

15
00:09:58,680 --> 00:11:02,880

16
00:13:49,000 --> 00:14:10,560

17
00:14:10,560 --> 00:14:14,640
```

There is a video on the left part with some transcription already integrated. These transcriptions have time lines. On the left, a text box is loaded with the same time lines. It allows a user to choose the language he wants to translate the video in and write in the appropriate time lines. We chose a particular pattern to be able to save it directly as an srt file, which is a standard translation format.
This tool also allows the user to load a particular srt file previously written.

This is of course very interesting for pure translation. But as we said it be somehow extended to write some annotations that would facilitate the understanding of a particular scene for people who do not speak the original language.

Conclusion

During this three-month internship, I worked on an innovative accessibility project intended to print-impaired people. Indeed the company's initiators always had disabled people interests at heart while developing their software. After having applied all the web guidelines to improve accessibility of their web sites to non-sighted and partially-sighted users, they wanted to create new standards for this poorly addressed disability.

I partially chose this internship because of this concern for accessibility that I share. It was in direct line with a project I carried through during my second year of studies at ENSEIRB, which was a musical software designed for blind people. I think that these issues are becoming more important as technology grows faster and especially in the multimedia technologies which is what I want to specialize in. These experiences will surely help me go about the end of my studies and my career with a different point of view.

In a personal respect, it was also very enriching to witness the way the company worked. First, the fact that it was based in India played a role in how they tackle the print-impairment issue. Secondly, the composition of the persons involved in the project was fascinating. The developing team is of course composed of computer scientists, but a lot of diverse people are affiliated with the project in some ways. May them be sociologists, craftsmen or artists, they all see the accessibility issue with a different perspective and their influence on the work of the company is very interesting.

Technically speaking, it was an extremely interesting work. I had the chance to be involved in this project at its inception. This means that I had to do all the researches necessary to begin the work. During three months I was challenged to find new leads never or poorly explored by others.

These researches bore fruits since we did implement a number of features in Pantoto, like the semantic associations and the Text-To-Speech tool.

This internship also gave me the chance to work with the latest web-technologies such as jQuery or CSS3 that I did not master before then.

The great outcome of this project is that it leaves some open leads for the future. Hopefully all these ideas we explored during these three months will be explored again and will really serve as a basis for new recommendations and standards.

4. Bibliography

Accessibility standards

http://www.standards-schmandards.com/

http://en.wikipedia.org/wiki/Web_accessibility

http://www.w3.org/TR/2008/REC-WCAG20-20081211/

http://www.w3.org/WAI/

http://www.w3.org/TR/CSS2/aural.html

Speech synthesis

http://vozme.com/index.php?lang=en

http://simpe.wikispaces.com/

Semantic tagging

http://www.w3.org/TR/xhtml-rdfa-primer/

http://www.digital-web.com/articles/rdf_for_the_rest_of_us/

Miscellaneous

http://pad.ma/

http://www.cs.sunysb.edu/about/news/hearsay.html

6. Appendice - Summary in French

ENSEIRB-MATMECA
Filière Informatique
33402 Talence

Servelots Infotech
3354 KR Road
Bangalore, 560070
Karnataka, India

Amélioration de l'accessiblité d'une plateforme web

Rapport de stage

Johanna ROYER
2009-2010

Résumé

Servelots est une petite entreprise qui développe un produit open-source pour la génération de sites web.
Ils veulent améliorer l'accès à Internet aux personnes illétrées.
Ce rapport présente et décrit les pistes explorées et développées au cours de ce stage de trois mois.
Il est à noter que ce rapport ne représente qu'une version résumée du rapport originel rédigé en anglais.

Maître de stage : TB Dinesh Professeur responsable : M. Aymeric Vincent

Introduction

Un des principaux intérêts de Servelots concerne l'amélioration de l'accès à Internet au plus de personnes possibles. Son dernier objectif est d'aider les personnes illétrées à obtenir les mêmes informations que tout le monde sur les sites web qu'elles visitent. C'est un problème important pour les personnes de l'entreprise car dans un pays comme l'Inde, la population de personnes illéttrées atteint des millions.

De plus c'est un problème relativement peu reconnu et peu de lignes de conduite sont disponibles pour les développeurs de sites web. Une des aspiration de l'entreprise est de découvrir de nouvelles façons d'améliorer l'accès à Internet aux illétrés et cela constituera l'objet de ce stage.

Le stage peut être divisé en deux parties assez inégales. Les premiers jours ont été consacrés à la prise en main du logiciel développé à Servelots. Le but était de me familiariser avec l'environnement avec lequel je devrais travailler plus tard. Cependant, environ un mois après le début du stage, une toute nouvelle version du logiciel est sortie et une partie du processus d'apprentissage a du être renouvellé. A cette occasion, j'ai pu corriger quelques erreurs dans le nouveau logiciel. Ce travail de correction de bugs n'ayant pas grand chose à voir avec mon projet, il ne sera pas commenté dans ce rapport. Cependant cela a été très bénéfique du point de vue de la prise en main de l'environnement de travail.

La deuxième partie, et la plus importante du stage a été dédiée au project d'accessibilité. Ce project a été dans son immense majorité un travail de recherche puisque ce sujet de l'accès à Internet aux analphabètes a été peu traité jusqu'ici. Les idées vers lesquelles nous avons orienté ce projet seront détaillées dans ce rapport.

Je décrirais en premier lieu le contexte du stage : l'entreprise et le logiciel qu'ils y développent. Ensuite, le point sera fait sur ce qu'est l'accessiblité. Enfin je décrirais le travail accompli au cours de ces trois mois et ouvrirais le sujet sur quelques pistes qui, nous l'espérons, serviront de base pour de futurs développements.

1. Présentation of l'entreprise

1.1. Contexte géographique

Bangalore est la capitale de l'état indien de Karnataka. Situé sur le plateau du Deccan dans le sud-est du Karnataka, Bangalore est la troisième ville la plus peuplée d'Inde. En tant que métropole en pleine expansion, Bangalore accueille un nombre important d'universités et de centres de recherche parmi les plus réputés du pays. La ville est connue sous le nom de *Silicon Valley* de l'Inde à cause de sa position de leader dans l'exportation de technologies.

1.2. Présentation de l'entreprise

Servelots est une entreprise qui fournit des services web à des Petites et Moyennes Entreprises. Elle a été fondée en 1999 par un groupe d'informaticiens qui voulait fournir des logiciels specialisé dans le secteur du développement social pour des PME.

L'entreprise développe et distribue un logiciel nommé Pantoto. Pantoto est un outil gratuit et open-source qui permet aux organisations de gérer leurs informations et communications sans dépendre d'un quelconque soutien technique.

L'équipe est constituée de trois ingénieurs qui recoivent l'aide occasionnelle de stagiaires et de volontaires intéressés dans le développement de Pantoto. Des personnes d'origines différentes gravitent également autour du projet, des artisans et des artistes entre autres.

1.3. Présentation de Pantoto

1.3.1. Description fonctionnelle

Pantoto est un système de structuration d'information flexible et extensible. Il est pensé pour être un logiciel communautaire - configurable, gratuit et open-source – qui peut capturer, maintenir et analyser les données d'une manière simple, sans le besoin d'un soutien technique permanent.

Le modèle de Pantoto est utile pour des secteurs tels que le développement social, scolaire ainsi que pour des petites entreprises. Pour ces organisations, qui travaillent avec beaucoup de données, la dépendance technique implique souvent une importante perte de temps et de ressources; il est donc important qu'elles puissent créer, maintenir et gérer leur système de gestion d'information par elles-même.
C'est dans ce but que Pantoto a été pensé et réalisé.

1.3.2. Description technique

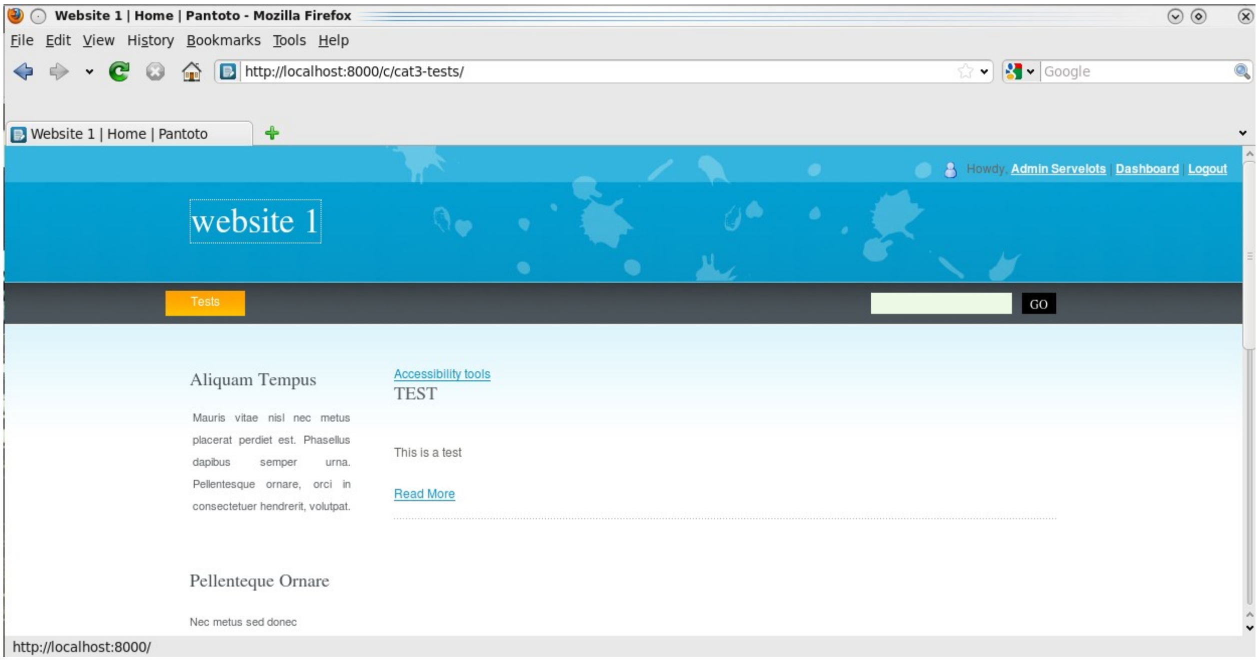
Pantoto est installé sur un serveur dans une organisation ou sur un hôte distant et les utilisisateurs ont seulemement besoin d'un navigateur pour organiser les besoins d'une communauté. Ceux-ci peuvent venir sous la forme de la creation de formulaires pour la dissémination et la récupération d'informations par exemple.

La dernière version de Pantoto a été réimplémentée avec des outils Web 2.0. Elle a été développée avec le framewrok Python Django qui permet de facilement créer des templates, controleurs et formulaires. Le modèle objet a été créé avec MongoDB et l'interface utilisateur est à base de jQuery et jqGrid.

1.3.3. Architecture fonctionnelle

Au coeur de Pantoto se trouve une entité nommée Pagelet. Cela peut être vu comme une sorte de formulaire extensible pouvant être manipulé par plusieurs auteurs pour créer un flux d'informations.
A titre d'exemple, un administrateur pourrait créer un formulaire d'admission pour une école. Ce formulaire pourrait ensuite être rempli par les parents de l'élève. Enfin l'administration de l'école pourrait ajouter les détails administratifs afin d'enregistrer l'admission.

A partir de ces pagelets, un utilisateur peut générer des sites web personnalisés. La structure de base sera la même pour tous les sites générés par Pantoto. On peut créer des catégories, qui deviendront le menu du site. Chaque Pagelet créée appartient à une certaine catégorie qui permettra d'y accéder dans le site. Ces Pagelets apparaîtront ensuite comme des articles dans la page web.


Website 1 | Home | Pantoto - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8000/c/cat3-tests/
Google
Website 1 | Home | Pantoto
Howdy Admin Servelots Dashboard Logout
Tests
GO
Accessibility tools
Read More
http://localhost:8000/

website 1

Aliquam Tempus

Mauris vitae nisl nec metus placerat perdiet est. Phasellus dapibus semper urna. Pellentesque ornare, orci in consectetuer hendrerit, volutpat.

TEST

This is a test

Pellenteque Ornare

Nec metus sed donec

Illustration 7: Site web généré avec Pantoto

2. Accessibilité

2.1. Définition de l'accessibilité

Le terme d'accessibilité décrit le fait qu'un produit, service ou outil est accessible par le plus de personnes possible. En général, il s'applique à l'accessibilité d'un produit à des personnes présentant un handicap. Au cours de ce stage et dans ce rapport nous nous concentrerons sur un type d'accessibilité : l'accessibilité au web; ainsi que sur certains types de handicaps : handicaps visuels et cognitifs.

Handicaps visuels

Les handicaps visuels incluent la perte totale ou partielle de la vision. Ces personnes utilisent généralement des lecteurs d'écran qui permet de retransmettre la plupart des informations visuelles présentes à l'écran. Afin de faciliter la navigation sur un site web aux personnes utilisant ces lecteurs d'écran, bon nombre de recommendations existent déjà. Nous garderons donc en tête ces recommendations tout au long du développement du stage. Cependant, nous nous intéresseront plutôt à une autre sorte de handicap, moins connue et donc moins abordée dansles standards d'accessibilité du web.

Handicaps cognitifs

En Inde, des millions de personnes sont touchées par le « print-impairement ». Il s'agit des personnes capables de voir mais n'étant pas à l'aise avec les textes. Il peut s'agir bien sûr d'analphabétisme, mais aussi d'autres types de handicaps cognitifs comme la dyslexie par exemple. Ceci peut paraître étrange, mais dans des pays comme l'Inde où l'accès à l'éducation est plus qu'inégal, le traitement de ce genre de handicap est bien plus laborieux qu'en France par exemple.

Bien sûr ces personnes peuvent utiliser les mêmes outils que les personnes aveugles. Mais nous avons besoin ici encore de resituer le contexte. La plupart des personnes « print-impaired » n'ont pas la possibilité d'installer les logiciels dont ils pourraient avoir besoin pour correctement utiliser un ordinateur ou naviguer sur Internet.
Par ailleurs, il peut être assez frustrant pour une personne voyante d'utiliser un lecteur d'écran. La plupart de ces logiciels se dirigent au clavier pour des raisons évidentes alors que des personnes capables de voir pourraient vouloir utiliser la souris et cliquer sur une section spécifique d'une page web pour en écouter le contenu.

Au cours de ce projet, nous nous intéresserons principalement à cette catégorie de handicap et à comment rendre un site accessible aux personnes atteintes.

3. Projet Accessibilité

3.1. Définition du projet

Comme nous l'avons dit précédemment, l'entreprise prend très à coeur les problèmes d'accessibilité. Après avoir appliquer du mieux possible les recommendations pour améliorer l'accès à leurs sites web aux aveugles et aux malvoyants, leur nouvel objectif et d'en améliorer l'accès aux « print-impaired ».

Leur idée va cependant plus loin puisque leur objectif final est de pouvoir proposer de nouvelles recommendations qui seraient applicables à n'importe quel site internet. Puisque de telles recommendations n'existent pas encore, une importante partie du stage a été dédié à de la recherche. Nous avons analysé les différentes technologies utilisées sur Internet et cherché de nouvelles pistes.
L'objectif était vraiment d'explorer le plus d'idées possibles sans pour autant les implémenter dans le logiciel. Nous décrirons plus loin les tests effectués et les solutions envisagées.

L'idée récurrente est que les utilisateurs « print-impaired » peuvent utiliser leur souris à leur guise au contraire des aveugles. En explorant cette piste, l'idée est venue de relier les éléments entre eux dans une page afin de pouvoir faire lire à lecteur d'écran uniquement ce que l'utilisateur veut.
Prenons l'exemple d'une page présentant le profil d'un visiteur sur un site internet. Ce type de page comporte généralement la photo du visiteur suivie d'une description de la personne. Mais cette page contiendra aussi des éléments « parasites ». C'est-à-dire des éléments reliés non pas au profil mais à la page web en elle-même. L'utilisateur pourrait être en mesure de cliquer sur la photo et d'entendre seulement les sections reliées à cette photo, soit sa description.

Logiquement, nous nous intéresseront en premier lieu aux associations sémantiques à l'intérieur d'une page. Il faut trouver un moyen de décrire les relations entre les différents éléments d'une page web.
Une fois cela fait, nous essayerons d'intégrer un outil de « Text-To-Speech » (TTS), soit un outil permettant de lire à voix haute un texte donné en entrée.
Nous décrirons dans les sections suivantes l'implémentation de ces fonctionnalités dans Pantoto et donnerons ensuite des idées pour généraliser leur utilisation à d'autres sites.
Enfin nous nous intéresserons également aux nouvelles technologies du web et à leur utilité quand à l'accessibilité que nous recherchons.

3.2. Associations sémantiques

La première partie de ce projet a donc consisté à définir par quels moyens nous pouvions relier les éléments d'une page, autrement comment introduire des informations sémantiques dans le code HTML.
Deux solutions ont été envisagées et analysées. Dans cette version simplifiée du rapport nous ne les décrirons pas en détail mais en viendront directement aux conclusions tirées de cette analyse.

La première possibilité envisagée est RDFa (Ressource Description Framework – attributes). C'est une recommendation du consortium W3C qui permet d'ajouter des attributs aux éléments XHTML en vue d'enrichir un document web avec des meta-informations.
Cette recommendation est née en vue d'améliorer la compréhension d'un site web par les programmes informatiques. Les attributs que le déveppeur peut ajouter à sa page HTML sont standardisés et donc compréhensibles par n'importe quel programme. Cependant il ne permet pas de relier un élémént spécifique avec n'importe quel autre, et pour pouvoir décrire une relation de son choix, le développeur aura certainement besoin d'ajouter de nouveaux vocabulaires aux standards existants.

La deuxième solution consiste aussi à ajouter des attributs au éléments HTML d'une page. Il s'agit ici de se servir des attributs `id` et `class` déjà présents dans HTML. En reprenant l'exemple de la page de profil cité précédemment, on pourrait attribuer un certain attribut `id` à l'élément photo. Ensuite on attribuerait aux éléments reliés (description de la personne), des attributs `class` du même nom. L'inconvénient de cette solution est que les relations ainsi décrites ne pourraient être comprises de n'importe quel programme. Cependant, elle a l'avantage de la simplicité puisque le développeur a simplement besoin de connaître le HTML. De plus tout type de relation peut être décrit ainsi.

Etant donné que pour notre problème d'accessibilité, le programme n'a pas besoin de connaître la nature de la relation entre les éléments, mais simplement besoin de savoir qu'ils sont reliés entre eux, la deuxième solution a été retenue par souci de simplicité. Bien sûr, l'utilisation des deux solutions peut être cumulées sur une même page si besoin.

3.3. Vers une généralisation de l'accessibilité

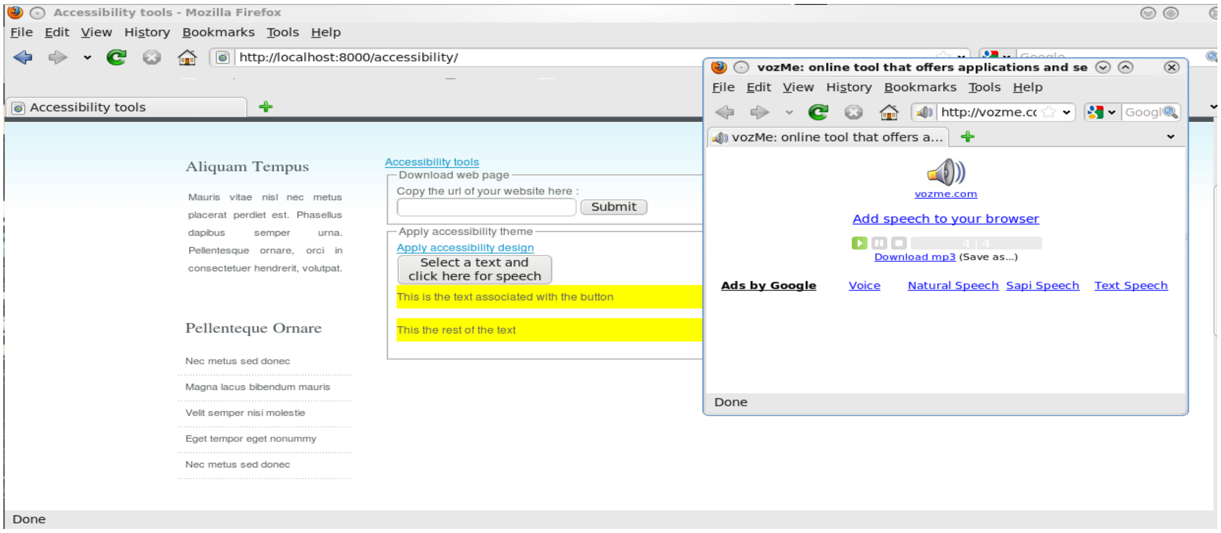
Nous présenterons tout d'abord les outils ayant été mis en place sur Pantoto, puis nous présenterons les idées que nous avons à plus grande échelle, pour rendre accessible n'importe quel site internet.

3.3.1. Travail sur Pantoto

Pour centraliser tous les tests que nous avons pu effectuer pendant ce projet, et ceux qui pourront être effectués par la suite, un environnement « accessibilité » a été créé. Il est accessible depuis les sites générés par Pantoto.

En ce qui concerne les associations sémantiques, nous avons donc choisi la méthode de « tagging HTML » qui consiste à ajouter des attributs `id` et `class` aux éléments que nous souhaitond relier. Nous avons testé cette méthode dans l'environnement de test et nous voulions l'étendre à tout le site de Pantoto. Cependant, à la fin du stage, le logiciel était encore en cours de développement, il était donc difficile de finir ce travail de liaison entre éléments puisqu'il impliquait de modifier beaucoup de fichiers.
Visuellement, lorsque l'on clique sur un élément « père », les éléments reliés sont mis en valeur dans la page. L'étape suivante a été de modifier un outil de TTS afin de lire ces éléments.

Le but n'était pas de développer un nouveau programme de TTS, cela aurait fait l'objet d'un stage complet. Nous avons donc cherché un outil que nous pourrions intégrer et modifier à notre guise. Il fallait donc que cet outil soit open-source, pour que nous puissions l'utiliser librement sur nos sites, et qu'il soit dans un langage facilement intégrable à Pantoto.
Nous avons trouvé Vozme, un logiciel permettant de lire du texte sélectionné dans un navigateur. Nous avons modifié le script fourni à intégrer dans notre code afin de l'alimenter avec le texte des éléments reliés.
Pour les mêmes raisons que précédemment, cet outil n'est pour l'instant intégré que la partie environnement de test, mais sera amené à être diffusé sur toute la partie « site internet » de Pantoto.


Accessibility tools - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8000/accessibility/
Accessibility tools
Aliquam Tempus
Pellenteque Ornare
Accessibility tools
Download web page
Copy the url of your website here :
Submit
Apply accessibility theme
Apply accessibility design
Select a text and click here for speech
Done
vozMe: online tool that offers applications and se
File Edit View History Bookmarks Tools Help
http://vozme.cc
vozMe: online tool that offers a...
vozme.com
Add speech to your browser
Download mp3 (Save as...)
Ads by Google
Voice
Natural Speech
Sapi Speech
Text Speech
Done

Mauris vitae nisl nec metus placerat perdiet est. Phasellus dapibus semper urna. Pellentesque ornare, orci in consectetuer hendrerit, volutpat.

Nec metus sed donec
Magna lacus bibendum mauris
Velit semper nisi molestie
Eget tempor eget nonummy
Nec metus sed donec

This is the text associated with the button

This the rest of the text

Illustration 8: outil TTS

3.3.2. Appliquer un thème « accessible » à d'autres site internet

Le but était d'appliquer ces outils que nous avons développé à d'autres sites internet. Puisqu'il n'est pas possible de modifier le code source en ligne d'une page qui ne nous appartient pas, l'idée est de récupérer le code source d'une page web et de la stocker sur un serveur dédié du côté de Pantoto. Une fois ainsi enregistrée, l'utilisateur pourrait la modifier et donc appliquer un thème « accessible » dessus. Il se pose un problème légal cependant : il faudrait que le code de la page soit open-source ou bien qu'elle appartienne à l'utilisateur voulant la modifier.
Les différentes étapes pour parvenir à implémenter cette fonctionnalité sont assez claires, mais par manque de temps n'ont pas été complètement implémentées.

La première de ces étapes est de récupérer le code source d'une page. Cette fonctionnalité est intégrée à l'environnement de test. Elle permet de copier l'URL de la page souhaitée et de l'enregistrer dans un dossier pré-défini.

L'étape suivante serait de modifier un éditeur de texte afin de pouvoir afficher le code de cette page et de relier les éléments souhaités grâce à la méthode de « tagging HTML ». On pourrait ajouter des options `id` et `class` à l'éditeur de texte déjà utilisé dans Pantoto,afin de marquer les éléments HTML.

Enfin, l'utilisateur pourrait enregistrer la page ainsi modifiée et y accéder à tout moment depuis son site web Pantoto personnalisé. Avec le thème « accessible » ajouté à la page modifée, il pourrait cliquer sur les éléments « pères » et entendre le contenu des sections reliées grâce à l'outil de TTS maintenant intégré à Pantoto.

3.4. Sites multimedia

Un autre aspect du travail sur l'accessibilité est de penser à comment les nouvelles technologies multimedia et les nouveaux standards peuvent profiter aux « print-impaired ». Les nouveaux standards comme HTML5 et CSS3 sont maintenant largement utilisés pour ajouter des effets visuels et de l'intéractivité aux sites web, mais comment peut-on utiliser ces fonctionnalités à notre avantage ?

3.4.1. CSS3

Une fonctionnalité intéressante de la nouvelle version de CSS est le module de « speech ». Ce module permet au développeur d'un site de gérer la structure d'un document en vue de le faire lire par un outil de TTS. Il peut ainsi gérer suivant les classes CSS, le type de voix, la fréquence, etc. Malheureusement, le développement de cette fonctionnalité s'est arrêté en 2004 et n'est pas très répandu. Il serait pourtant intéressant de l'explorer plus avant. On pourrait imaginer la possibilité de faire modifier le style d'une page par les utlisateurs afin de mieux cadrer leurs besoins de façon personnalisée par exemple.

3.4.2. Videos

Une alternative évidente au texte est la video. La dernière version d'HTML, HTML5 facilite d'intégration de videos dans une page web grâce à une nouvelle balise <video>. Un des avantages de cette balise est qu'elle n'oblige normalement pas l'utilisateur à installer de nouveaux plugins puisqu'elle utilise en principe les codecs nativement intégrés à la plupart des navigateurs. Un autre avantage est que cette balise offre une API unifiée, peu importe les codecs dont l'utilisateur se sert, les mêmes méthodes sont disponibles. Cela permet notamment de modifier l'apparence du lecteur video si on le souhaite.

3.4.3. Traduction

Un autre type de problème d'accessibilité est la barrière de la langue. Ce problème peut toucher tout le monde, et cela est particulièrement vrai en Inde où chaque état possède son propre langage.
Des solutions existent déjà pour palier à ces problèmes de traduction. Comme tout d'abord les ites proposant un langage alternatif pour tout ou une partie d'un site internet. Cette solution a l'avantage d'être fiable au niveau de la traduction, par contre si le contenu n'est pas statique cela impose une mise à jour régulière du contenu alternatif.

On peut également intégrer à son site un outil de traduction comme le Google Translator Toolkit. Cela permet de ne pas avoir à mettre à jour différentes versions du même site. Les visiteurs du site pourront choisir de sélectionner une partie du contenu de la page et utiliser cet outil pour la traduire dans la langue de leur choix. L'inconvénient avec cette méthode étant que les traductions ainsi générées ne sont en général pas de bonne qualité.

La qualité de traduction peut être influencée par des références culturelles. Dans ce sens et dans l'esprit d'interactivité qui est celui du Web 2.0, il pourrait être intéressant que les utlisateurs puissent ajouter eux-même du contenu au site pour en améliorer la traduction. Au cours d'un workshop nous avons travaillé sur cette idée avec une équipe travaillant sur un autre projet relié au multimédia.

Conclusion

Au cours de ces trois mois de stage, j'ai travaillé sur un projet d'accessibilité innovant destiné aux « print-impaired ». En effet, les créateurs de l'entreprise ont toujours eu les intérêts des personnes invalides à coeur. Tout ce qu'ils développent étant déjà le plus accessible possible pour les aveugles et mal-voyants, leur nouvel objectif est de créer de nouveaux standards pour aider les « print-impaired ».

J'ai en partie choisi ce stage pour cet intérêt pour l'accessibilité que je partage. Il présentait une continuité avec un projet efffectué au cours de ma 2e année à l'ENSEIRB, qui était un logiciel musical destiné aux aveugles. Ces problèmes d'accessibilité sont d'autant plus important dans le domaine des technologies multimedia dans lequelles je veux me spécialiser. Ces expériences m'aideront certainement à envisager la fin de mes études et ma carrière sous un angle différent.

D'un point de vue personnel également, il était très enrichissant d'être le témoin de l'organisation de cette entreprise. Premièrement, le fait qu'elle soit basée en Inde joue un rôle important dans leur manière d'aborder le problème des « print-impaired ». Ensuite, la nature des personnes impliquées dans le projet était assez fascinant. L'équipe gérant le développement du logiciel sont bien sûr composée d'informaticiens, mais un certain nombre de personnes d'origine différentes prennent part d'une manière ou d'une autre aux projets de l'entreprise. Qu'ils soient sociologues, artisans ou artistes, ils abordent tous ces questions d'accessibilité d'ue manière différente et leur influence sur le travail de l'équipe est très intéressant.

Techniquement parlant, le travail était également très intéressant. J'ai eu la chance d'être impliquée dans ce projet dès ses débuts, ce qui veut dire que j'ai eu un travail de recherche important avant de pouvoir commencer à travailler dessus. Pendant trois mois j'ai été au défi de trouver des pistes jamais ou peu explorées par d'autres.

Ces recherches ont porté leurs fruits puisqu'un certain nombre de fonctionnalités ont finalement été implémentées dans Pantoto, comme les associations sémantiques et l'outil de Text-To-Speech.

Ce stage m'a aussi permis de travailler avec les technologies du web les plus à la pointe comme jQuery et CSS3 que je ne maîtrisais pas auparavant.

Le résultat le plus intéressant de ce projet est qu'il laisse plusieurs portes ouvertes pour le future. Nous espérons que toutes ces idées explorées au cours de ces trois mois le seront à nouveau et serviront de base pour de nouveaux standards et recommendations.