Multilingual Web Site Construction and Maintenance

Giorgos Lepouras & Costas Vassilakis

Department of Informatics

University of Athens

157 71 TYPA Buildings

Athens, Greece

e-mail: {glepoura, costas}@di.uoa.gr

Abstract

The construction of multilingual web sites is probably the best answer to addressing the problem of the diverse cultural background of the Internet community. However, developing multiple instances of the same site in different languages induces increased overhead for both the implementation and the maintenance phase. The paper reviews current techniques and describes an alternative to constructing multilingual web sites, which eases the development and maintenance phases, without possessing any of the drawbacks of existing tools. The paper concludes proposing possible future enhancements.

Keywords: multilingual, internationalisation, web site development, web site maintenance

Introduction

The past few years have witnessed a significant rise in the number of people that use the Internet as a means of communicating and gaining access to information. The web site has become the premium means of disseminating information to Internet users. Although the Internet community initially consisted mainly by English speaking users, this is not anymore the case. To this end, web sites are being developed that contain the majority of their information in a language other than English. This trend results in web sites that are usable only to persons who can understand the web site's language and exclude all other users.

Localisation of information [1, 2] may be employed to adapt the web site to other languages, although in some cases the problem may be alleviated by providing support for key words [3]. For web-based information systems localisation implies that the information will be stored in more than one language, creating therefore a multilingual web site.

A multilingual web site as already stated contains information in more than one language. There is usually one main language and translations for some or all of the web pages in other languages. The user selects the preferred language of interaction, and in most cases, at any point of the navigation, he/she may switch between the same page in different languages, assuming one exists (as shown in figure 1).

Although such a web site meets better the needs of the multilingual Internet user community, the effort required for its design, implementation and even more for its maintenance is often prohibiting. It is only recently that tools have appeared that take into consideration the requirement for multilingual support both form the developer's and the user's point of view. Even though these tools offer some alleviation of the problem, they suffer from other disadvantages which render their use impractical at least for certain cases.

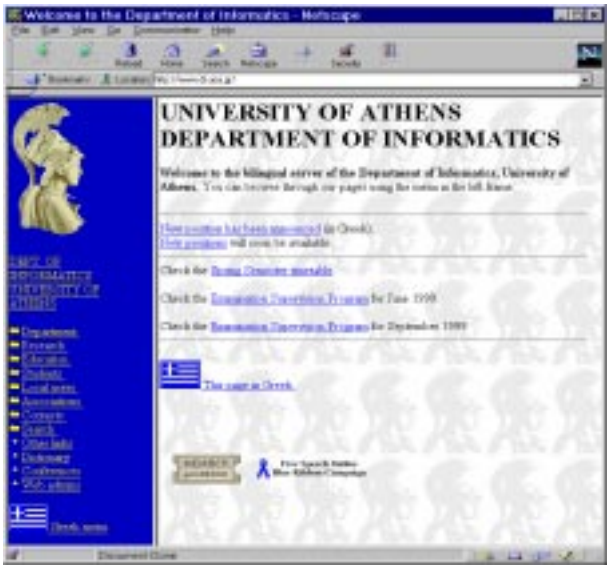

Figure 1: Department of Informatics bilingual site

The rest of the paper continues as follows: section two describes the general issues concerning the design and implementation of a multilingual web site, section three presents related work, section four describes a new approach for a multilingual web site construction tool and section five outlines its implementation.

Design and implementation of a multilingual web site

The design of a multilingual web site has to deal with some extra issues due to the coexistence of the same information in many different languages. To this end the web site has to be designed to be extensible to more than one language. Its design should allow the easy location and modification of the language versions of a file. Among other issues to be examined, a scheme has to be devised to allow the categorisation of files according to language. To easily identify the various language versions of the same file two alternatives can be used. The first relies on a naming convention and the second on the directory structure. According to the first all file names should have an extension to signify the corresponding language version. That is *in***de***x.html* should be named *in***de***x*-**e***ng.html* for the English version and *in***de***x*-**gr***k.html* for the Greek version. This includes all types of files and not only html files. For example, an image file containing a picture of flag should also have this extension. Files that are being shared between different languages do not have an extension. Since this scheme stores all versions of the same file in the same directory, it will work well for a small number of languages, but it may have problems when the number of language is large. In such a case the number of files in each directory may hinder the maintenance process. Alternatively the web site may be designed to utilise a different directory for each language version. This solution is depicted in the next diagram.

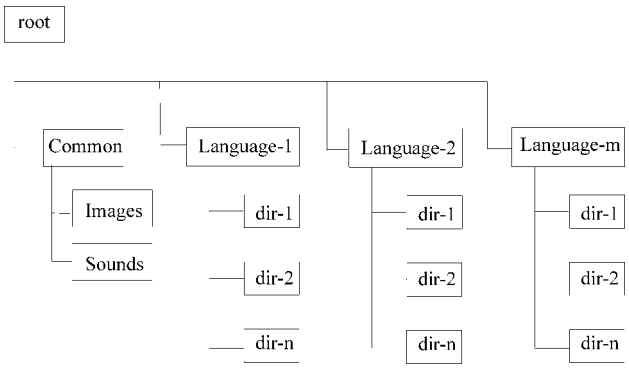

Figure 2: Mirror directory structure approach

In the above diagram *Languag***e**-*i* directory contains the same sub-directories as *Languag***e**-*1*. There is no need for a file to exist in all language versions, but if it does it should have the same name for every version. To distinguish the language version of a file, one has to know the directory path of this file. Files that are common between language versions are stored separately. These are usually data files and images. It should be noted here that in above diagram, *Languag***e**-*i* directories may also contain language-dependant images or sounds. Only files that are shared between languages are stored under the *C***o***mm***o***n* directory.

Once the design issues have been dealt with, the implementation phase can start. During this phase a number of tedious tasks have to be carried out. These include:

- the implementation of the site in the main language,
- the replication of the site for each one of the supported languages and
- the creation of links between sibling pages of different languages.

Except the first one of these tasks that takes place for every web site constructed, the rest of the tasks are particular only to multilingual web sites. The replication of the site for each one of the supported languages involves the translation of each web page along with the creation of the corresponding links between pages. Of course, this task requires the presence of one or more experts who will undertake the effort of accurate translation to the supported language set. Accuracy in translation is necessary for pages to convey the same information regardless of the language, otherwise users able to understand more than one of the supported languages may be confused. Once the replication is completed, links have to be created between versions of the same page to enable the navigation through the language set.

When the development phase is completed, the maintenance phase starts. The information of the web site has to be kept up to date, new information has to be inserted, while other information has to be removed. This should happen in a consistent, uniform manner to ensure that the information provided by a web page remains the same in all supported languages. The update process can prove to be a time and effort consuming task. Again the need for an expert translator may be of the essence. Even in cases where information has to be removed, the web constructor/ administrator should be able to identify where the corresponding piece of information lies in the different language versions of the file.

During the first days of the web, web developers used to write their pages employing a simple text processor. Nowadays, a variety of tools has and is still being developed to assist both the needs for web page authoring and maintenance. These tools along with their features that address multilingual web site issues are outlined in the next section.

Web site development tools

Tools for web page authoring and maintenance can be classified into two broad categories. The first consists of tools used to compose web pages and the second tools that ease the maintenance of web pages. Of course, the distinction between the two categories is not clear and a tool from one category may comprise features of the other.

A*uth***or***ing t***oo***l*s

Web site authoring tools are usually enhanced versions of editors that enable the user to compose a document and save it in HTML format. Although nowadays most web authoring tools offer a WYSIWYG interface, older versions can be regarded as text editors that can validate HTML. Some authoring tools provide extra features such as the ability to create directories, to search for unlinked documents or images, to find "broken" links or to publish directly web documents to multiple web servers. Web authoring tools ease the task of creating and publishing documents to

the web, but they do not provide much support for the phase of maintenance, especially for multilingual web sites. Since they do not keep any kind of translation memory, the search and replace functionality they offer is restricted to one language at a time. On the other hand the web developer may employ a translation tool that supports HTML formatting (such as SYSTRAN professional [4]). This type of tool does not provide the range of facilities for web page development an authoring tool does, but enables the easy translation for the supported language pairs.

D*at*a b*a*se b*a*sed *t*oo*l*s

Database-based tools (such as Oracle's WebDB [5]) employ a database to store web documents. The general approach of this solution is outlined in the next figure.

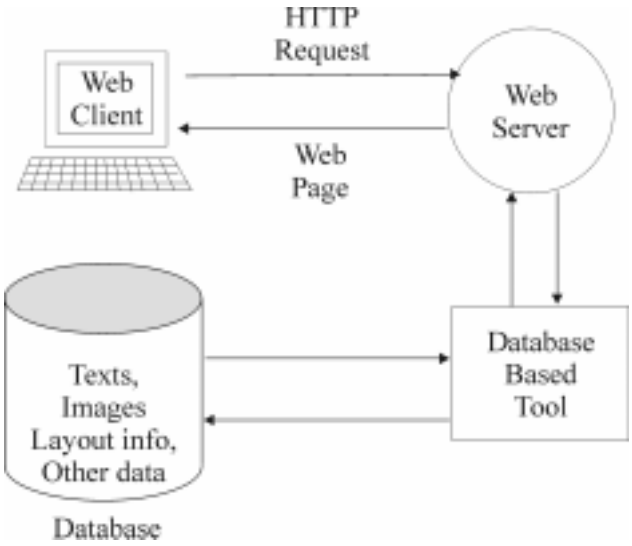

Figure 3: General architecture for a database based tool

The database may contain already created (static) web pages, but it usually holds data such as texts and images, along with the necessary layout information to enable the dynamic creation of web pages. Thus, when a web client sends an HTTP request to the Web server, the request is forwarded to the database based tool. The tool will read the necessary information from the database, format it according to the layout information and send it back through the web server to the client. Especially in the case of Oracle's WebDB, the tool automatically selects correct display language based on the browser settings. If for example, the user has selected Greek encoding in his/her browser, and the requested page exists in that language, the tool will send back the Greek page, otherwise it will send the default page.

By separating the layout information from the actual data the task of maintaining the same page in several languages can be facilitated. A web administrator can easily force a uniform appearance on the web documents and achieve a consistent user interface. On the other hand the solution of a tool that employs a database to store data suffers from some drawbacks. The first is the requirement for the presence of a database. This requirement increases the cost for the development of the web site, and although it facilitates the maintenance of the site it imposes a need for a person capable to administer the database. These extra necessities may render inappropriate the use of such a tool for a small company or an individual. Secondly, since the web pages are being stored in some proprietary format, the web developer does not have direct access to the HTML pages. As a consequence he/she cannot make direct modifications to the web pages (by means of a text editor). This also means that the output of the tool is not portable/exportable. If at some point the web developer decides to switch to another tool he/she has to develop the web site from the beginning. Finally the use of a database and the dynamic creation of web pages puts an extra load on processing power and I/O capacity requirements.

Development and maintenance tool

From the issues discussed in the previous section it can be noted that even though the idea of creating a multilingual web site to better serve the needs of the Internet community sounds attractive, the development of such a web site can prove to be a difficult and cumbersome task. The creation of a bilingual web site will demand more than double the effort of creating an equivalent monolingual web site, especially due to the effort needed for maintenance. In a multilingual web site, the number of web pages is multiplied by a factor equal to the number of supported languages. This could result in an enormous number of pages that have to be maintained and kept up to date. Existing tools either do not support the maintenance phase or set extra requirements, locking web developers with a specific software vendor/solution.

The functional requirements of a tool addressing the development and maintenance of a web site should cover all the phases of the site's life cycle, without experiencing the problems of the existing tools. To this end it is necessary to detect the tasks that can be automated as well as the tasks for which the web developer can be supported.

The first category comprises mainly of tasks that deal with the creation of web pages. To this end the tool should:

Automatically replicate the structure of the web site to each one of the supported languages. Depending on whether the web constructor wants to distinguish between language versions of the same file using a name convention or a mirror directory structure, the tool should be able to implement the preferred solution. This is probably the first and most basic phase in the implementation of a multilingual web site. It can easily be automated and rid the developer of a routine task.

Create corresponding to the first language files for each one of the supported languages. If the developer has chosen to follow a naming convention based on the extension, then for the Greek language and for a file named *Welcome-eng.html* the tool should create an identical file named *Welcome-grk.html*. If the developer has chosen to create mirror directories then for the Greek language and for a file named *index.html* that resides under */English/Main* the tool should create an identical file named *index.html* under */Greek/Main*. The tool should change all HTML tags to reflect the current character set. For example, for the corresponding file in Greek the tool should change the charset enconding to ISO-8859-7. Furthermore, in the case of the <Font Face> tag the tool should be able to set the corresponding font depending on the language and according to a user preset font substitution list and in the case of mirror directory structures the BASE HREF tag can be adjusted accordingly. It should also be able to easily incorporate features such as the one considered for the new version of HTML [6], where the web developer will be able to add link information for corresponding pages in other languages in the HEAD of the document.

Consequently the tool should create the corresponding links between the new files to match the structure of the existing main language hierarchy. The implementation status of the multilingual web site at this stage is depicted in the next diagram, where the tool creates a mirror web page hierarchy for Language-2 based on Language-1. Note here that under common reside all shared among supported languages files such as images and sounds. If an image (such as the image of a flag depicting the current language selection) or sound is language dependent it should be stored under Language-n. The user should also be able to designate whether specific pages should be created only for a subset of site's language. If a page is chosen not to appear in a language, a default page may be specified.

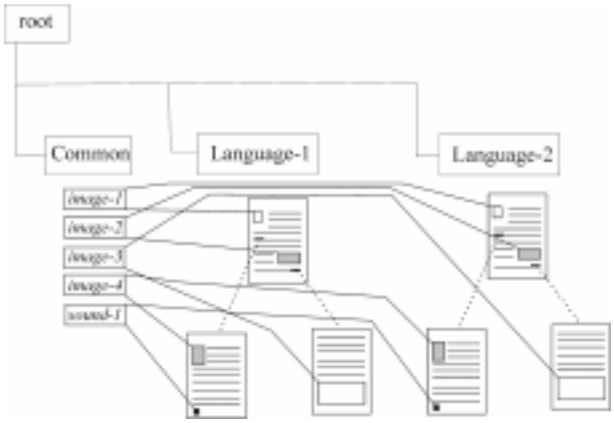

Figure 4: Replication of links for supported languages

Finally, the tool should create links between language versions of the same page. This will enable users to navigate through the supported language set. This stage is illustrated in the next diagram.

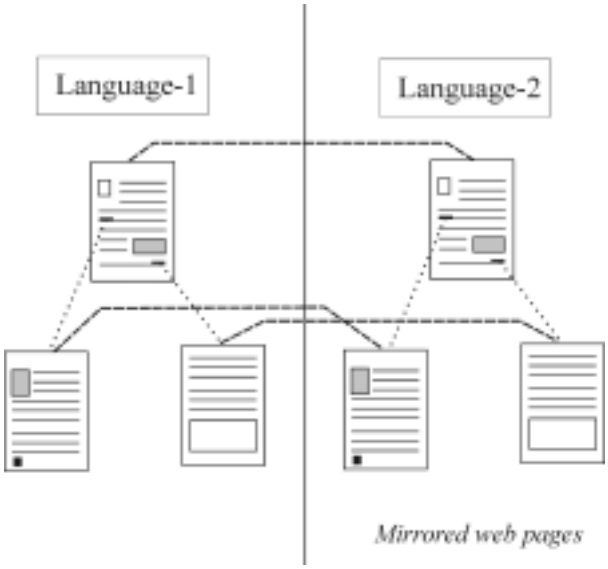

Figure 5: Creation of in-between languages links

The tool should automatically create site map for each language set supported in the web site. Although site maps can assist users to navigate easily and quickly in a web site, the implementation and maintenance of a site map needs a lot of effort if it has to carried out manually. The tool should also create and maintain indexes in each one of the supported languages to further aid users' navigation.

The second category of requirements includes tasks that deal both with the deployment and the maintenance of the web site.

The tool should propose translations of phrases whenever this is possible. This can be achieved if the tool maintains a translation memory on a phrase to phrase basis. The translation memory will be updated and enrinched each time the user translates a web page from one language to another. In the case a phrase does not already exists in translation memory the tool should be able to parse the phrase, filter out any formatting tags and store the pair (original – target language) of phrases. The translation memory can support both the development and the maintenance phase. It can be used when translating from one language to another to propose a translation if the same phrase exists in memory. It can also be used to locate one phrase in the different language versions of the same file. By doing so it could reduce the amount of time and effort to maintain a large number of pages in a variety of languages.

To test the possible effectiveness of translation memory for a multilingual web site maintenance a survey carried out. In this survey, a portion of the www.w3.org web pages (1100 pages) were parsed. It is worth noting that almost 25% of the phrases were encountered more than once.

The function of the translation memory may be further enhanced be means of dictionaries, especially terminology ones. The tool can help the web developer to maintain a consistent user interface for all or for subsets of the web pages through the use of web page templates. The web developer should be able to create different templates for each one of the languages or for different sections of the web site

Current status

Currently, a tool is being developed according to the specifications given above. In its present state the tool contains enough functionality to enable a web developer to use it for web site construction. The tool employs a simple database developed in MS Access 97 to store the translation memory. Currently work has focussed on the improvement of the parsing algorithms for the translation memory database as well as on the implementation of frame and javascript support. The next figure illustrates the function of automatic translation for the current version of the tool.

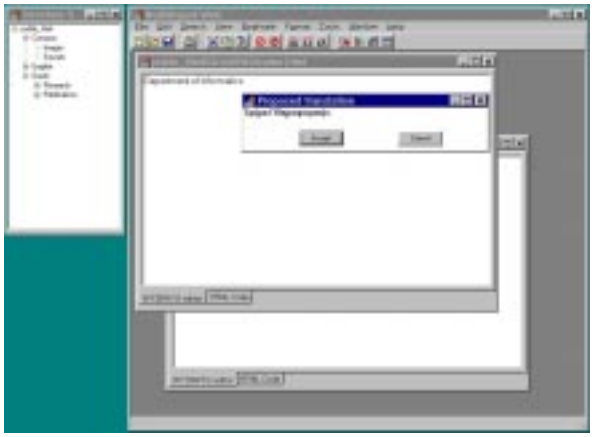

Figure 6: Multilingual Web tool

As depicted in the figure, the user is able to select a phrase in the editor then run a search for a proposed translation. The user is able to select a folder from the floating window representing the directory structure, ask for replication or check the files he/she does not wish to replicate to another language. In the next version of the tool, the editor will parse the document in the original language and propose a translation whenever one exists.

Conclusions

This paper illustrated the issues concerning the design and maintenance of multilingual web sites. It enumerated the tools aiming at this target along with their advantages and disadvantages. It then continued to describe the specifications for a tool that exploits well known methodologies and techniques to overcome other tools' disadvantages. To this end the proposed tool, can aid the average web developer who does not want to commit to a specific tool or invest to a proprietary technology, but needs to create and maintain a multilingual web site.

We believe that once complete, the envisaged tool will provide an efficient solution to the creation of multilingual web sites.

References

[1] E. Uren, H. Robert and T. Perinotti, Software Internationalization and localization. An introduction. Van Nostrand Reinhold, New York, 1993.

[2] J. Karat and C. M. Karat, Perspectives on Design and Internationalization, SIGCHI Bulletin, vol. 28, No. 1 (Jan) 1996, p. 39.

[3] G.R.S. Weir & G. Lepouras, Dynamic Second Language Support for Web-based Information Systems, Proceedings of SCI' 98 Conference.

[4] Systran translation software, professional edition, home page: http://www.systransoft.com/rofessional.htm

[5] Oracle's WebDB home page http://www.oracle.com/ products/tools/webdb/

[6] World Wide Web Consortium available at http://www.w3.org/International/O-help-lang.html