

Project P923-PF

Multilingual WEB sites: Best practice, guidelines and architectures

Deliverable 1

Guidelines for building multilingual Web Sites

Volume 5 of 5: Annex D

Define possible architectures for the pre-selected services

Suggested readers:

This document is primarily aimed at anyone who is involved in the process of designing, building or managing WEB sites. It is of immediate relevance to those involved with multilingual WEB sites, but it nevertheless, provides information which will allow monolingual WEB site designers to design sites that are economically upgraded to multilingual sites.

EDIN 0011-0923

Project P923

For full publication

September 2000

EURESCOM PARTICIPANTS in Project P923-PF are:

- Koninklijke KPN N.V.
- France Télécom
- British Telecommunications plc
- Telecom Italia S.p.A.
- Portugal Telecom S.A.

This document contains material which is the copyright of certain EURESCOM PARTICIPANTS, and may not be reproduced or copied without permission.

All PARTICIPANTS have agreed to full publication of this document.

The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the PARTICIPANTS nor EURESCOM warrant that the information contained in the report is capable of use, or that use of the information is free from risk, and accept no liability for loss or damage suffered by any person using this information.

This document has been approved by EURESCOM Board of Governors for distribution to all EURESCOM Shareholders.

.

Executive Summary

The objective of this PIR is to define a global architecture that includes components for building, managing and exploring multilingual web sites. Our investigations for existing or possible architectures for multilingual web sites allowed us to conclude that no generic or abstract architecture exists and no standard model has been defined. To define such architecture and to organise this task, we have broken down a web service into "functional elements" :

- Information provision: text-based, discourse and sentence-based.
- Information retrieval: mono & cross language retrieval.
- E-commerce: currency, legal issues, distribution (pragmatic distribution).
- User interface: getting information from/about the user.
- Multimedia information.

The content of this document is organised into three sections. The introduction section presents our approach to this task and introduces the functional elements that have been identified. Section 2, the largest section of this document, contains a detailed description of each functional element in a multilingual perspective. Moreover architectural requirements for each functional elements are identified and presented. Based on these individual architectural requirements, section 3 summarises global architectural requirements for multilingual web site management and exploration. The architectural components comprise tools, procedures, static and dynamic components that are necessary to be added or adapted to monolingual web site architecture to make it multilingual. Some features are also considered in building and managing multilingual web sites.

List of Authors

Almeida Luis (PT)

Appleby Stephen (BT)

Beires Nuno (PT)

Boualem Malek (FT)

Boves Louis (KPN)

Branco Glória (PT)

Codogno Maurizio (IT)

Den Os Els (KPN)

Vinasse Jérôme (FT)

Contents

| | |
|---|----|
| Executive Summary | 1 |
| List of Authors | 2 |
| Abbreviations | 4 |
| 1 Introduction | 5 |
| 2 Description of functional elements and architectural requirements | 7 |
| 2.1. Description of the information provision functional element | 7 |
| 2.1.1. General description of the information provision functional element | 7 |
| 2.1.2 Authoring Systems | 15 |
| 2.2 Description of the information retrieval functional element | 17 |
| 2.2.1 Information retrieval techniques | 17 |
| 2.2.2 Mono-lingual and cross-language information retrieval | 17 |
| 2.2.3 Information extraction..... | 21 |
| 2.3 Description of the E-commerce functional element..... | 22 |
| 2.3.1 General description of the E-commerce functional element | 22 |
| 2.3.2 Architectural implications | 23 |
| 2.4 Description of the user interface functional element..... | 25 |
| 2.4.1 General description of the user interface functional element | 25 |
| 2.4.2 Motivation for multilinguality in user input..... | 27 |
| 2.4.3 Architectural components required for user interface | 28 |
| 2.5 Description of the multimedia functional element | 32 |
| 2.5.1 General description of the multimedia functional element | 32 |
| 2.5.2 Motivation for multilinguality in multimedia functional element | 32 |
| 2.5.3 Architectural components required for multimedia information..... | 32 |
| 2.5.4 Relation of multimedia with other functional elements | 33 |
| 3 Global architectural requirements for multilingual web sites based on functional elements..... | 34 |
| 4 References | 37 |

Abbreviations

| | |
|------|-------------------------------|
| CGI | Common Gateway Interface |
| CSS | Cascading Style Sheets |
| DLL | Dynamically Linked Library |
| HTML | HyperText Markup Language |
| I18N | Internationalization |
| ISP | Internet Service Provider |
| LID | Language Identifier |
| MT | Machine Translation |
| TM | Translation Memory |
| URL | Universal Resource Locator |
| WAP | Wireless Application Protocol |
| WWW | World Wide Web |
| XML | eXtensible Markup Language |

1 Introduction

Our investigations to establish a state of the art of architectures for multilingual web sites allowed us to conclude that no generic or abstract architecture exists and no standard model has been defined. In fact it is not realistic to define or to draw a schema of a generic or abstract architecture for multilingual web sites. However, multilingual web site architecture can be seen as the association of a basic architecture for monolingual web sites together with specific complementary components related to specific languages. These components depend on the kind of application(s) targeted by the multilingual web site. The objective of our project is not to define basic architectural components for building, managing and exploring monolingual web sites. Also it is not aimed to go into technical details of web site architecture. The reader can refer to other literature (made for webmasters) to learn how to build and manage monolingual web sites. Our approach is to describe explicitly the complementary components to be added to a monolingual web site to make it multilingual. Some analysis allows to classify multilingual web sites into two main different categories:

- multilingual web site built as several parallel monolingual versions,
- multilingual web site built as one multilingual version including several languages.

It appears that this classification is not formal and it is not guided by any defined and precise rules. That means that a multilingual web site can be built according to one of these categories or it can also merge both of them. For example a multilingual web site may include pages grouping several European languages and other pages with only non-European languages put separately. The reason is that some languages have some common features that allow to group them and other languages have different features and need to be in separate parts. Finally some kind of confusion remains between building new multilingual web sites and localising monolingual web sites. It appears that the creation of a new multilingual web site can be seen as a localisation process of a monolingual web site based on one of the specified languages.

According to these considerations and to the ideas developed in previous steps in the project and to organise this task, we have broken down a web service into "**functional elements**":

1. **Information provision:** text-based, discourse and sentence-based.
2. **Information retrieval:** mono & cross language retrieval.
3. **E-commerce:** currency, legal issues, distribution (pragmatic distribution).
4. **User interface:** getting information from/about the user.
5. **Multimedia information.**

A web service will effectively consist of one or more functional elements. For example a web service dedicated to e-commerce might include information provision, e-commerce and user interface functional elements.

There will not be one global architecture for all multilingual web-based services. Instead there will be various architectural components which depend on the kinds of functions being provided by a given web site. That means that a set of architectural components will be defined for each functional element. Furthermore, these components are unlikely to be independent. Our approach to organising this task is to take each functional element in isolation and consider its demands on architectural

components in a multilingual web site. We are then in a position to consider the details of an aggregate architecture that might best serve various combinations of functional elements. It is expected that each functional element will require the presence of certain components in the architecture (and procedures for using those components). Conceptually, one could create a grid that relation the functional elements to the architectural components.

| | | Architectural components | | | |
|---------------------|-----------------------|--------------------------|-----|-----|-----|
| | | AC1 | AC2 | AC3 | ... |
| Functional elements | Information provision | ◆ | | ◆ | ◆ |
| | Information retrieval | ◆ | | | ◆ |
| | E-commerce | ◆ | ◆ | | |
| | User interface | ◆ | | ◆ | |
| | Multimedia | ◆ | | | |

[◆ indicates if architectural components are appropriate to functional elements].

Figure 1. Conceptual schema of functional elements and architectural components

2 Description of functional elements and architectural requirements

Each partner of the project has been assigned the responsibility of one of the functional elements (information provision, information retrieval, e-commerce, user interface and multimedia information). Descriptions of functional elements have been provided with relation to multilinguality. Moreover architectural requirements for each functional element have been identified.

2.1. Description of the information provision functional element

2.1.1. General description of the information provision functional element

In this section we discuss processes and procedures related to different types and sources of information that may be presented in multi-lingual web services. The discussion in this section is limited to information that is best presented in the form of written text. Thus, we will not deal with information that is presented in the form of speech or other audio, nor in the form of video or any comparable medium. Multimedia information is covered in detail in section 2.5. Also, we will not cover 'text' information that is embedded in graphics (and that therefore may not be accessible in the form of ASCII or Unicode characters). Information types embedded in graphics are also dealt with in the Functional Element "Multimedia". However, it must be emphasised that 'text embedded in graphics' may pose a real challenge, even if it does not lead to a problem. In many cases that text will go undetected, with the obvious implication that it will not be translated. The problem with text embedded in graphics will affect the operation of both fully automatic and human translation, especially if the latter is supported by tools like multi-lingual HTML editors, which again are likely to miss the text, and therefore will not offer it to the human translator. For the time being this finding should result in a recommendation for web site designers not to embed or display text beyond logos and names (which do not need to be translated) in the form of graphics. Information search is covered in section 2.2. If the search returns full text documents, the application must be able to display its contents, irrespective of the language. This may require automatic recognition of language and character encoding.

2.1.1.1 Architectural Implications

The very general remarks made above have several implications for the architecture of a web site that must be able to support multilinguality. The following aspects must be kept in mind:

- The design should specify that all documents that are under direct control of the site manager must have a standardised language identification (LID) tag. Such a tag makes language identification superfluous.
- Unless it can be guaranteed that each file has a standardised language identification tag, the architecture must contain a language identification tool. The data flow must be designed in such a way that each file/document to be displayed either comes with a LID tag, or can be processed by the LID tool.
- The design should plan at least for the use of ISO-8859-x, or preferably, for the use of Unicode.

In all what is written below it is assumed that these general requirements are fulfilled. Therefore, we will not reiterate the need for the use of Unicode each time this may be relevant.

For almost all operations related to information provisioning the designers and managers of multilingual web sites will find a need for multilingual HTML editors.

In order to be able to identify the architectural implications of textual information provisioning it is necessary to draw up a taxonomy of 'linguistic information types'. In drawing up such a taxonomy, we focus on those issues that have a direct impact on the processes involved in the design, construction and maintenance of multi-lingual web sites. In this section only the general architectural implications related to specific information types are discussed. The tools to support the use of specific information types are described in section 5.2.

Five major features must be distinguished because of their impact on web architecture:

1. Linguistic Structure of the message
2. Presentation of the information: published vs. interactively queried
3. Rate with which the information is updated and accessed
4. Source of the information
5. Personal or group characteristics of the receiver(s)

These features/factors are not necessarily orthogonal.

2.1.1.2 Linguistic structure of the message

Here the most important criteria are the length of a typical 'message' in terms of the number of phrases and the stage (or point in time) where the translation is performed.

Length and complexity of the messages determine the type of tools that are necessary for its processing. 'Processing' depends heavily on the source of the information (i.e., whether it is monolingual, or perhaps even non-linguistic data). 'Processing' also depends on the moment when it must be performed (i.e., during the integration of the information in the web site, or each time the information must be displayed. In this section we will limit the discussion to tools and architectural elements that are directly related to the need to make a web site multilingual. More specifically, we focus on tools which support some kind of machine or human translation.

The **simplest form of text** is made up of the menu items that can be selected by clicking, or the names of fields in forms that must be completed (cf. the section on 'User interface' below). Here, a multi-lingual terminology database is probably sufficient to support the designer of the web application. Perhaps the terminology database must be connected to a thesaurus, to cater for the risk that the most obvious equivalent is too long to fit properly in the graphical design of the page. The thesaurus can then be used to find a suitable alternative expression. It is assumed that the selection of the menu items will be made by the designer/implementor of the web page. This implies the assumption that these menu items are fixed for the lifetime of a given version of the web application. Thus, tools that support the designer to find suitable translations of the menu items are sufficient.

Here it is explicitly assumed that it is only necessary to translate terms in menus and forms. If the structure of the menus or forms must be adapted to fit with the requirements of a particular language or locale, these adaptations are outside the scope of the tools implied below. If such structural adaptations are necessary, a completely different set of tools, procedures and expertise is called for.

2.1.13 Architectural Implications

Tools that are needed to handle short phrases in a multilingual environment include:

- multilingual terminology databases
- multilingual thesauri

Multilingual terminology databases and thesauri are also essential for the processing of more complex text types. However, we will not repeat this need for the text type discussed below: the (human) translator should rather plan to collect terminology databases and thesauri relevant to the contents of a web site under all circumstances.

If the **texts consist of short phrases**, a multi-lingual phrase book that covers the domain is probably adequate. One can also envisage effective use of translation memory systems in this context. Phrases may, for instance, play a role in on-line catalogues, where product names and a specification of the major features of the products are given. Contrary to what was said above for menu items, the multi-lingual phrase book may need to be accessible on-line, during the use of the application. This is the case if a monolingual catalogue (database) is used which is maintained independently of the web site, so that the translations must be performed on demand. Alternatively, the tools can be used to support the creation of a multi-lingual catalogue (database). The choice of the most appropriate architecture and procedure(s) will depend on many issues. If the translation is done on-line, one must provide tighter quality control for the phrase book, and its completeness must be guaranteed. For off-line preparation of a multi-lingual catalogue (database) the requirements may be less strict, because there is always a competent human being who makes the selection.

Architectural Implications

Tools for use with short phrases include:

- multilingual phrase books
- translation memories

Both multilingual phrase books and translation memories will also be needed with the more complex sentences and texts discussed below.

To facilitate access to phrasebooks and translation memories it is probably essential to have a tool which removes HTML (and other mark-up) codes from the text taken from the web site; the codes must, of course, be restored in the output of the translation tools.

It may prove to be convenient to have a tool that can monitor the minimum width and height on the screen necessary to properly display the output of a translation action.

If **the texts consist of complete sentences**, translation poses more requirements. Not only the phrases must be correct, but also the overall syntactic structure. Here too, one must distinguish between applications where a multi-lingual database is generated off-

line, and applications where translation is done on the fly. For both types of applications, potentially suitable translation tools are on the market. In fact, all translation tools that feature the requested language pairs are potentially useful, including -but not limited to- translation memories.

Architectural Implications

- Handling complete sentences in a multilingual environment requires access to full machine translation systems.

Alternatively, or in combination with completely automatic machine translation, one may need tools to support human translation.

If **the texts consist of paragraphs** that exhibit some kind of discourse structure, fully automatic translation becomes questionable. Consequently, there is a need for tools in the field of computer assisted translation. At the same time, tools that propose a fixed structure for a text are likely to be very useful, if only because they support the alignment between versions of the text in multiple languages.

As was said several times before, text translation can be performed on-line (each time the document is requested and viewed) or off-line.

Architectural Implications

- Handling full paragraph texts require the same translation tools as mentioned for the simpler text types.

It is conceivable that Content Classification tools are a useful complement to fully automatic machine translation. Whether content or domain classification makes a difference for the quality or speed of machine (or human, for that matter) translation depends on the capability of the translation tools to adapt to specific contents or domains.

2.1.1.4 Presentation of the information: published vs. interactively queried

By “**published information**” we mean texts that were carefully designed for readability and clarity, so as to make the most important pages in a web site maximally attractive. On the other hand, “**interactively queried**” information refers to documents that are retrieved through some on-line query, from a database of documents which may turn out to be monolingual, and in a language different from the text to be shown in the browser.

All consultants in the field of multi-lingual publishing (and multi-lingual web sites) recommend that the translation of the top level pages (i.e., the pages that a visitor is most likely to see first when entering the site) must be performed by trained human translators. Many are even stricter, and recommend that these translators must be native speakers of the target language. Of course, the human translation process can be supported by the usual tools for computer assisted translation. These are the same tools as mentioned above. One will want to build up domain specific terminology databases, and domain specific translation memories.

2.1.1.5 Architectural Implications

- If the top-level pages must be translated by expert human translators, these pages cannot be dynamically generated each time the page is requested. At the very minimum, the translated text must be available for insertion in the page. One cannot count on on-line translation while the page is being generated and formatted.
- If pages or documents must be translated on-line, direct access to a translation engine in the web must be available.
- The details of this access depend on the origin of the texts that must be translated. If the text is identified by a URL, it should suffice to submit that URL to the translation engine.
- In order for a translation engine to be suitable for on-line translation of web sites, it must come with software that analyses the page layout of the source text, and that formats the translation results in a way that is visually/graphically close to the visual appearance of the source text.
- Machine translation tools used for the translation of web sites must contain procedures for the detection and subsequent proper handling of all XML/HTML mark-up.

The **need to translate** pages in a web site is not always equally urgent. It is obviously impossible to translate all documents in all potentially relevant monolingual databases into all relevant languages. Often, this is not necessary either. For instance, the technical documents in EURESCOM need only to exist in English, since it is safe to assume that all intended readers have a sufficient command of English to digest the documents.

If translation on demand is necessary, one must rely on one of the few existing tools for fully automatic machine translation, like Systran, Reverso, and iTranslator. It is well known that the quality of the translations is difficult to predict, and that a fixed minimal quality cannot be guaranteed. However, one may expect that the reader knows the domain, and that (s)he has some knowledge of the source language, so that (s)he will be able to interpret the translated text in the large majority of the cases. In the design of a web application one must carefully and explicitly decide which part of the documents can be left to on-line translation on demand, and which part must be pre-translated with some kind of human intervention.

Architectural Implications

- It may happen that one cannot afford to have specific pages in a web site translated by expert humans, even if these pages are requested quite frequently by visitors who would prefer to read them in a different language. In these situations repeated on-line translations may generate an undesirable load on the network and the CPU cycles of the translation engine. Thus, one must decide whether machine translated versions of frequently accessed pages must be kept in the same way as human translated pages.

One key issue that distinguishes between web sites is the **degree of parallelism** between languages, i.e., the degree to which all information is available and accessible in the exact same form, irrespective of the language. One way to provide complete

parallelism is to make all documents available in all languages, with the exact same information and the exact same hyperlinks. Another way to implement full parallelism is by means of some form of on-line translation of all relevant pages and documents. In principle, browsing may then be done in a monolingual environment.

We must consider the impact of partial translation of documents that can be accessed directly by selecting a link. Fully equivalent (i.e., language independent) behaviour may only be possible if all documents are translated. Here too, it is necessary to decide whether part of the documents need only to be provided in a subset of the languages.

Architectural Implications

- If it necessary to make browsing exactly equivalent between the multilingual versions of a web site, the complete hyperlink structure of all language versions must be identical.

If pages are submitted to a machine translation engine, that engine must be able to handle all mark-up codes correctly.

The degree of parallelism that can be obtained is limited by the extent to which a site contains links to external sites. If an external site maintains multiple versions of documents in multiple languages, the translation engine is probably not able to substitute the URLs.

2.1.1.6 Rate at which the information is updated and accessed

The way in which documents and texts are best presented in multiple languages also depends very much on the rate with which they are updated. Information that exhibits a high rate of change is probably best left to some kind of machine translation, except when that information is closely linked to pages that must be of very high quality (i.e., the pages that one would like to publish). This implies (once again) the need to **classify all documents and information types** during the design of the web site: at the very minimum, **two classes of documents** must be distinguished, those that **change only occasionally** and that need “published” quality, and those that **change frequently**, and might get by with lower quality translation and presentation.

Architectural implications

- In the design of a web site a distinction should be made between dynamic and static information. Static information can be generated in all relevant languages once; if necessary, intensive human involvement is possible. Dynamic information will most likely require some kind of machine translation.
- For the generation of dynamic information tools are needed which help to ease the process of machine translation. To some extent, tools to enforce the use of controlled language during the writing of dynamic texts may help to improve the readability, both of the text in the source language, and of the translated versions.

Frequency of access is another issue that must be considered. Information that may never be requested before it is updated probably does not need to be translated with human intervention. As an example, take the information about the Dance and Opera Theatre in den Haag. In periods in which there are no EURESCOM meetings in den Haag, human supervised translation of this information would be wasteful. Moreover, an interested visitor will most probably be able to figure out the relevant details of a performance, even if the machine translation is not perfect. It may be necessary to shift documents between classes, but this has a deep impact on the processes and the architecture of the web application. If it is decided that documents must be moved

from the “interactive access” class to the “published” class (for instance in connection with a special event) these documents must be moved to other positions in the hyperlink tree. Most probably, the complete access procedure will then have to change (e.g. local storage instead of on-line retrieval from another site). Of course, the same problems will occur if a set of documents is moved to another category on a permanent basis. This only underlines the importance of the initial design of the architecture.

Architectural implications

- Frequency with which information is (expected to be) accessed must be taken into account during the design of the web page. The same requirements apply as for the frequency of update above. The most important difference is perhaps the fact that frequency of access is more difficult to predict during the design phase. Therefore, it would be nice to have tools that make it easy to migrate documents or information classes between the categories that qualify for human or machine translation.
- The need to have information translated on demand requires that text, and/or URLs, can be submitted for on-line and real-time translation.

2.1.1.7 Source of the information

The most important distinction here is the **storage of information as texts, or as data that must be converted to text and/or graphics for presentation**. If data is present in non-linguistic form, one must consider whether presentation in the form of text (sentences or paragraphs) is most appropriate. For instance, if the EURESCOM site would give access to five days weather forecast in Porto, presentation of the data in the form of some kind of graphics might be as appropriate as presentation in the form of a text paragraph. Of course, graphical presentation is assumed to be language independent. Obviously, graphical presentation is only attractive if tools for the generation of the pictures are available. For more information on the architectural requirements and the tools related to graphical presentation of data, the reader is referred to the section on multimedia.

If information which is stored in non-linguistic form must be rendered as text, tools are required for Natural Language Generation. Moreover, those tools must be able to generate text in all relevant languages. These tools have already been summarised in PIR.3.2. If the information to be presented exists in the form of text, the same considerations apply as described above.

Architectural Implications

- If the non-linguistic data to be presented are stored in a database that serves the function of a catalogue, text generation requires tools for on-line translation of short utterances (typically words or phrases). Multilingual phrase books, lexicons or thesauri are probably adequate, provided that they cover the domain of the application.
- If database information relates to prices or measures, the generation tool must be able to convert prices and measures into the most appropriate units. It remains to be decided what ‘the most appropriate’ is (should the site adapt to the language, or to the country from which the information is accessed).

For a limited category of non-linguistic information (weather reports, reports about sections of the stock market, etc.) it may be necessary to generate paragraph length

texts. Few tools presently exist that are able to support this generation task, let alone in multiple languages.

The source of information may also have another impact on the architecture of an application. If a web site provides links to many remote sites, and if the information on many of these sites is updated at high rates, it will become unattractive (if not unfeasible) to collect all information on a single local server for translation and storage. For instance, if the EURESCOM server would provide access to weather forecasts, hotels, restaurants, theatre programmes and special events in all relevant cities, keeping that information up-to-date in all relevant languages in the server in Heidelberg might easily lead to congestion. This is the more wasteful if there is a non-negligible probability that a large part of the information will never be requested through the EURESCOM server.

These considerations seem to imply another issue that must be investigated and on which decisions are needed during the design of a multi-lingual web application. The open character of the global Internet will make it impossible to pre-translate all information that might be accessed via hyperlinks. The only way to create an environment in which all the information is available in all languages in perfect parallel would be a closed application, which does not provide hyperlinks to URLs outside the application proper. Obviously, this is not very attractive, at least for many applications. On-line high quality translation would be a good alternative, but it has already been said that high quality machine translation in unlimited domains is not feasible now, and will not become feasible in the intermediate future.

It is obvious that machine translation in the web requires tools which are able to detect phrases with a special status, that should not be translated. In more general terms, tools are needed that are able to handle HTML and XML mark-up, so as to prepare texts containing those mark-ups for machine translation. These tools are not necessarily trivial. For instance, some expressions that consist of contiguous words in the source language may translate into words that are not contiguous in the target language. This identifies a class of tools which are needed irrespective of the details of the translation process.

Architectural Implications

The architectural implication related to the presentation of external web pages are not different from what has been said above with respect to the rate with which the information is updated and accessed.

2.1.1.8 Destination of the information

Part of the information that must be conveyed through a web site may come **from documents that are already prepared for presentation in a suitable browser**. However, we are likely to see an increase in the number of applications that separate browsing and navigation from presentation. Examples of such applications include services that can be accessed using different types of terminals (PCs when at home or in the office, mobile terminals with small screens and without keyboard for mobile access, or access through WAP and VoiceXML). Separating presentation from the other layers in a web application will have a deep impact on the tools for text generation.

In the generation of the presentation for multiple devices it is necessary to distinguish between static and dynamic pages. If all pages are essentially static (and only a selection is made between 'parallel' pages according to the capabilities of the terminal) only tools are needed to generate the pages in all relevant languages off-line.

However, if it is necessary to generate (part of) the pages dynamically, suitable generation tools for all relevant languages must be available on-line.

In an information service where (part of) the messages must be summarised to allow display (for instance because they must be presented on a mobile terminal) summarisation can be done in one language (viz. the source language of the message), after which the summary can be translated into the relevant language for presentation. The only drawback of this approach is that it may be more difficult to translate a summarized sentence, since there would be less context for choosing the best terms.

Considering all things related to the impact of display devices with varying capabilities it is fair to say that the impact mainly comes from the need to represent 'information' in several different textual presentation forms.

Architectural Implications

The requirements made by the display device do not differ significantly between static and dynamic pages. The only difference is the stage(s) during which the tools must be accessible.

As far as we could make out, no suitable tools are available to automatically generate textual representations of information that are suitable for display on specific terminals.

2.1.2 Authoring Systems

Authoring Systems or Authoring Tools are software packages which aim to support the creation of pages. Both the creation of text, HTML and page layout can be supported. In the context of BabelWeb only authoring tools that support the creation of Web pages are relevant.

From a quick scan of the information on the Internet it appears that the large majority of the web authoring tools are either freeware, or low cost. Most of the tools which are offered are geared towards the English language. However, there are a number of tools that claim to support other languages, and notably languages with writing conventions different from English.

A quick analysis of the tools which are available does not show clear distinctions between "authoring systems" on the one hand and "(multilingual) HTML Editors" on the other. Yet, an attempt is made to keep the two concepts apart.

2.1.2.1 Multilingual Authoring Systems

In fact, few if any applications that deserve the name "Authoring System" were found which deserve the epithet 'multilingual'. At best, multilingual text editors can be found. Here is a sample:

<http://www.aramedia.com/uwop.htm> offers two text processors which can handle multiple languages (among which Arabic).

<http://www.lastech.com/products/indoweb/indoweb.htm> lists a multilingual authoring tool for creating web pages in an number of (12?) Indian languages.

<http://www.lang.duke.edu/unieddll> advertises a Unicode compliant multilingual text editor (UniEdit) which comes as a DLL. UniEdit comes as a by-product of a language learning and authoring system WinCALIS, developed at Duke University. As far as one can tell from the brochure, WinCALIS is mainly meant to support the creation of computer assisted language learning.

2.1.2.2 Multilingual browsing and search

On <http://www.unn.ac.uk/~cggh1/webguide/Index14.html> a list of web tools can be found, including multilingual browsers.

Here there is a quote from that site, on multilingual browsers:

Although English is pretty much lingua franca (that is a switch) of the Web, it is not the only language being used on the Web. Here are some Web browsers with multilingual capabilities, so you can cruise the Web in a native or adopted language other than English."

Accent Multilingual Mosaic
<http://accentsoft.com/Main/moseng.html>

"Supports 30 different languages, including Russian, Arabic, Greek, or Japanese, even while using the U.S. version of Windows. Designed to work under any language version of Windows 95/98 or Windows 3.1 (requires Win32s, available at site). 30-day evaluation version"

HMView by Bersoft Hypertext Systems
<http://traviata.nta.no/index.html>

"Supports Dutch, Spanish, French, German, Italian, English. Supports frames, and is an off-line browser as well. Registered users can redistribute freely (as CD-Rom frontend or whatever)."

The EU funded project *Desire* is aimed at making scientific information sources and services in the web multilingual. A paper entitled "Developing multilingual subject gateways", by Emma Worsfold, Paul Hofman, Debra Hiom contains a section on multilingual search engines, that is quoted in full here:

A number of the Internet search engines are now offering some form of multilingual service. DESIRE reviewed nine of these to see what could be learned (the results of this review can be found in Appendix 2 of this report). It was found that few of the services made any detailed attempt to describe their provision for multilingual information retrieval and resource selection. Services that offered some form of multilingual provision left unanswered questions about which resources were included and which were excluded. The engines reviewed are: AltaVista, Euroseek, Excite, Lycos, Hotbot, Infoseek, Magellan, Metacrawler, and WebCrawler.

(<http://www.sosig.ac.uk/desire/lang/language.html>)

<http://www.uyip.org/> is the home page for Understanding Yiddish Information Processing. This site lists a number of web browsers which support Hebrew.

2.1.2.3 Multilingual HTML Editors

We start this section with a quote from a document on Internationalisation, written in the framework of the *Desire* project:

For example web browsers and servers have been able to negotiate character sets and encoding for some years, but it was only in January 1997 that RFC2070 proposed the introduction of an <LANG> element to allow for author supplied hints as to the language that a section of a document is written in. Even now this element is not part of the W3C Recommended HTML 3.2 DTD and serious deployment of an I18N ready version of HTML will only really begin with the recent introduction of HTML 4.0. Without such an

element, software that relies upon knowing the language for correct operation (such as automated translators, speech synthesisers and in some cases even the rendering engine) have a much harder, if not impossible, task ahead of them.

Note that <LANG> does not exist even in HTML 4.01 ... The most sensible way to embed language information in an HTML page is to add a element with a corresponding style.

An early reference to the specification of multilingual HTML was found in a report on a Symposium on Web Internationalisation & Multilingualism, 20-22 November 1996, organised by the IFO2000 project.

A search in AltaVista for "Multilingual HTML Editor" returned no matches. An advanced search with {"HTML Editor" AND "Multilingual"} did return a fair number of matches, among which pages with lists of HTML Editors for download (e.g., <http://library.thinkquest.org/11341/htmledit2.html>). Many of these tools are also referenced by <http://www.unn.ac.uk/~cggh1/webguide/Index14.html>. It appeared, however, that many (if not most) of the links in this list end up in a Page Not Found Error. None of the editors on which information could be found feature true multilinguality. At best, they promise that one can exchange sets of files with a single command. If one can provide multilingual copies of the texts in parallel files, the multi-file update capability. The best one can get seems to be automatic conversion of special characters to proper display/rendering.

Through http://web-developer.com/html/html_editors.html an HTML editor for Chinese/ English was found.

LanguageWare.net (through WholeTree.com) promises multilingual HTML editors, but rather appears to offer services for the translation of web sites and e-commerce applications.

2.2 Description of the information retrieval functional element

2.2.1 Information retrieval techniques

A survey of the major techniques for information retrieval is presented in a section entitled "*Multilingual/Cross-Linguistic Information Retrieval*" of the PIR.3.2 document "*Inventory of language related tools*". In the first part, we presented an overview of the traditional techniques (full text scanning, inversion, signature files and clustering). In the second part we discussed attempts to include semantic information (natural language processing, latent semantic indexing and neural networks). In this text related to the PIR.4.2 document "Define possible architectures for the pre-selected services", we will discuss the integration of information retrieval tools to multilingual web sites.

2.2.2 Mono-lingual and cross-language information retrieval

2.2.2.1 Definition

By "*Cross-Language Information Retrieval*" we mean the retrieval of documents based on explicit queries formulated by a human using natural language, when the language in which the documents are expressed is not the same as the language in which the queries are expressed. It is the ability to issue a query in one language and receive a document in another that distinguishes cross-language information retrieval

from monolingual information retrieval. Although monolingual information retrieval is outside the scope of this definition, cross-language and monolingual retrieval functionality can certainly both be provided by a single system. Cross-language information retrieval implicitly includes written text, speech, and perhaps sign language as possible modalities, although many writers limit their attention to text without further qualification when using the term "information". DARPA has adopted the term "*Translingual Information Management*" to describe a set of functions that include both cross-language retrieval, visualisation of multilingual document collections, and other issues that involve management of multilingual information. In some ways "translingual" is actually more descriptive of the underlying problem, that is to transcend the specific languages used by authors and searchers. But "cross-language" is now so well established that it is not clear whether "translingual" could displace it. In some of the early works on this topic the term "*Multilingual Information Retrieval*" was used. But the term "multilingual" is too seriously overloaded to be useful in this context. In particular, systems that are capable of monolingual information retrieval in more than one language correctly claim to be monolingual, and the increasing attention being paid to information retrieval in languages other than English has resulted in development of several such systems. Some of this research has investigated issues which will be important to cross-language information retrieval systems as well, such as segmentation, morphology, stopword lists, and user interface localisation.

2.2.2.2 Cross-language information retrieval and machine translation

In this section we discuss the effectiveness of machine translation in cross-language information retrieval systems. There are two main strategies of translation: query translation and document translation.

Users seeking information from a particular information source could benefit from the ability to query large collections once using a single language, even when more than one language is present in the collection. If the information they locate is not available in a language that they can read, some form of translation will be needed. On the other hand, support for free text searching across languages is not yet widely deployed, and fully automatic machine translation is presently neither sufficiently fast nor sufficiently accurate to adequately support interactive cross-language information seeking. An active and rapidly growing research community has coalesced around these and other related issues, applying techniques drawn from several fields - notably information retrieval and natural language processing - to provide access to large multilingual collections.

A controlled vocabulary information retrieval system can be very useful in the hands of a skilled searcher, but end users often find free text searching to be more helpful. Free text retrieval systems typically rely on matching features derived from query terms with features derived from the terms that appear in the document collection. Cross-language free text retrieval thus requires that either the representation of the query or the representation of the document (or both) be translated so that the two representations are compatible. When storage is limited or several languages must be accommodated, translating the query is more practical than translating each document into every language. On the other hand, a strategy based on document translation can permit the translation workload to be performed at indexing time. These alternatives and variations on them, such as mapping both the queries and the documents into language-independent representations, present fundamental tradeoffs that designers of cross-language information retrieval systems must consider.

Document translation essentially reduces cross-language retrieval to its monolingual equivalent, but query translation strategies can impose unique requirements on the five retrieval system components shown in Figure 2 [8].

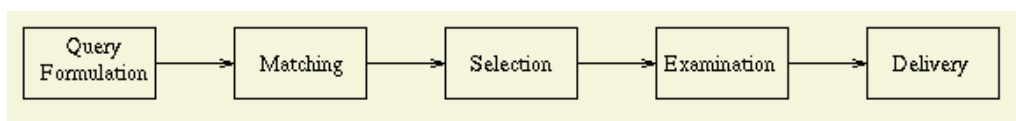


Figure 2: Retrieval system components

Free text queries posed by end users are often quite short, making it difficult to use context to limit translation ambiguity. Providing facilities for interactive disambiguation in the query formulation interface thus offers the potential to improve retrieval effectiveness [9]. Here we present some examples of use of machine translation in cross-language information retrieval systems.

a. Cross-language information retrieval using document translation

The most known machine translation system associated with information retrieval engines is *Systran* system associated with the *AltaVista* research engine (<http://www.altavista.com>) and with the international *Voila* portal of *France Telecom* (<http://www.voila.com>). More recently the *Reverso* machine translation system (developed by *Softissimo France*) has been integrated to the local *Voila* portal of *France Telecom* (<http://www.voila.fr>).

The principle of cross-language information retrieval using document translation is shown in the following example:

URL:

1 Select a translation:

- English to French
- English to German
- English to Italian
- English to Portuguese
- English to Spanish
- French to English
- German to English
- Italian to English
- Spanish to English
- Portuguese to English

2 Enter a URL or type plain text to translate here:

3 Translate

The Systran machine translation system associated with the "Voila" web portal and search engine (<http://www.voila.com>) can translate textual parts of web pages (with indication of URL's) from English to five other languages and vice versa. Translation of web pages can also be executed on each of the URL's proposed by the search engine as search results. Moreover translating of free plain texts is also available but limited to only a certain length of text.


b. Cross-language information retrieval using query translation

One good example of a system that allows cross-language information retrieval using query translation is the *Arctos* system developed at the CRL laboratory (Computing Research Laboratory) at New Mexico State University. Arctos is a cross-language, interactive retrieval system that uses a combination of automatic and user-assisted methods to build and improve cross-language queries. Arctos is a conjunction of the URSA retrieval engine with a cross-language technology for the interactive creation of queries, multilingual document retrieval and translation. The queries can be sent to Web search engines for interactive document visualisation and translation. URSA (Unicode Retrieval System Architecture) combines Unicode display technology developed at CRL with translingual information retrieval, multilingual collection visualisation and document management [10].

A demo of Arctos is available at the URL: <http://messene.nmsu.edu/ursa/arctos/>

The following example¹ uses this demo to show the principle of cross-language information retrieval using query translation.

URL: <http://messene.nmsu.edu/ursa/arctos/>



ARCTOS uses URSA (Unicode Retrieval System Architecture)
CRL - Computing Research Lab at New Mexico State University

----- **SCREEN 1** : -----

(a) *English Query:*
translation

Query

Select a target language:

| | |
|--|-----------------------------|
| | German French Italian |
|--|-----------------------------|

Example: Query = keyword 1, keyword 2, keyword 3, ...

----- **SCREEN 2** : -----

- Default translation of keyword 1 = translation 1.1

¹ Authorisation to include this example has been given via email by CRL (Computing Research Laboratory) at New Mexico State University, by April 6, 2000

| | |
|--|---|
| <ul style="list-style-type: none"> - Suggestion of similar forms = translation 1.2, translation 1.3, translation 1.4, ... - Possible choice of a different translation = translation revision. - Default translation of keyword 2 = translation 2.1 - Suggestion of similar forms = translation 2.2, translation 2.3, translation 2.4, ... - Possible choice of different translation = translation revision. - Default translation of keyword 3 = translation 3.1 - Suggestion of similar forms = translation 3.2, translation 3.3, translation 3.4, ... - Possible choice of different translation = translation revision. - etc. | <div style="border: 1px solid black; padding: 5px; display: inline-block;">Submit revised query</div> |
| <p>----- SCREEN 3 : -----</p> | |

Search results in the target language.

| |
|--|
| |
|--|

The query submitted in a source language is translated in a target language. The translation can be revised. Then the query is submitted and search results are displayed.

2.2.3 Information extraction

One of the typical features of standard search engines is their impermeability to user needs. Statistical techniques of relevance calculus can help a lot in refining and expanding queries, but they are hardly effective in the task of fulfilling more specific desiderata of the user. In a sense, they tend to interpret every query as a request of information about a certain topic, whereas the user might be interested in other (and more specific) forms of interaction, such as buying, selling, renting, downloading, talking, etc. Most Web search engines are based on keyword retrieval of text, however, where the intention is to identify a product or service from within a catalogue, other types of retrieval might be more appropriate, which take advantage of the structure of the product information. For example, when buying a car there are particular fields that can be used to capture the customer requirements, such as price range, model, colour, engine size etc.

It is assumed that a functional dimension has to be added to the indexing and retrieving machinery of a standard web based search engine. It is evident that this dimension can not be reached by using standard information retrieval techniques. This new dimension is called "Information extraction". Indeed, once user's expectations have been identified, information extraction techniques are able to provide much more effective results, as they can analyse small parts of documents just for the purpose of mining the kind of data in which the information seeker might be interested.

2.3 Description of the E-commerce functional element

2.3.1 General description of the E-commerce functional element

E-commerce is, of course, a strong motivation for companies to be interested in using the Internet. Small companies in particular can reach their (perhaps highly specialised) markets much more easily.

The role of the Internet Service Provider (ISP) in this is to provide the support an infrastructure to enable businesses to sell products over the Internet.

In this section we are concerned with the architectural design of a WEB site that provides a multilingual e-commerce facility. In general, an e-commerce WEB site will require all of the functional elements described in this document (information display, information retrieval, user interface, multimedia, e-commerce). In this section the focus will be on those architectural requirements which are more specifically required for selling goods or services on from WEB sites. Issues relating to the integration of the various architectural components will also be discussed.

It is important to note that producing a WEB site is necessary but certainly not sufficient for international e-commerce. In particular, this section will not cover legal issues relating to international commerce.

2.3.1.1 Internet Service Provider role

The role of the Internet Service (ISP) provider is not normally to sell goods over the Internet. The ISP is likely to need to offer an environment in which other businesses can conduct e-commerce. For SMEs, this means simplifying the technological aspects of selling goods on the Internet as much as possible.

At its simplest, a business may simply want its presence on the Internet. This may mean a few simple pages to show what the business does and give contact details.

At the other end of the spectrum, a business may require a full e-commerce solution including publicity pages, help with export legalities, goods cataloguing etc. The customer may also wish the ISP to co-ordinate translation of their site. Here the choice of a good architecture and appropriate middleware will be paramount. The fact that the ISP has an architecture that supports efficient localisation may well discourage the customer from attempting to manage their own multilingual site.

2.3.1.2 The process of e-commerce

The process of e-commerce normally begins when a potential customer has some requirement for goods or services. The supplier needs to make sure that his goods or services are advertised and available to the potential customer. The WEB is largely unstructured, so it can be quite difficult for a potential customer to find a supplier and similarly difficult for suppliers to target their advertising. Portals and search engines give a small amount of structure by providing a starting point. Once the potential customer has identified a possible supplier, then the customer will visit the supplier's WEB site.

The supplier's WEB site must give the potential customer confidence in the credibility of the supplier. Unlike a physical building, the customer will find it much more difficult to get an impression of the level of professionalism of the supplier. Typically, a WEB site will begin with an introduction to the company and a general

overview of the product range. For a multilingual WEB site, these first pages will need to be available in several languages. The translation quality here is of the utmost importance, and so it will need to be translated by a professional who is familiar with marketing in the target locale.

Once the potential customer has been reassured by their initial encounter with the WEB site, they will want to delve further to find more specific product and pricing information. If the company has a wide range of products, this will require some kind of searching facility, perhaps combined with hierarchical organisation of the WEB site.

Many WEB search engines are based on keyword retrieval of text, however, where the intention is to identify a product or service from within a catalogue, other types of retrieval might be more appropriate which take advantage of the structure of the product information. For example, when buying a car there are particular fields that can be used to capture the customer requirements, such as price range, model, colour, engine size etc. (e.g. <http://www.bmw.co.uk/>).

Having found a possible product, the customer is likely to want to know more about it. There may be some technical details, or instructions for its use that should be presented to the customer.

Now that the product has been identified, the customer will need to indicate that they are willing to buy the item. Then the user will need to enter information about themselves and their method of payment (normally a credit card for direct purchases on the WEB).

The payment will need to be verified and the goods dispatched.

In between placing the order and the customer receiving the goods, it may be desirable to keep the customer informed of progress with the order (positive confirmation of order acceptance, dispatch etc.).

There are numerous variations on the above buying/selling procedure. For example, when buying a used car, the WEB might be useful in placing the buyer and seller in contact, but will not normally play any further role in the selling process.

2.3.2 Architectural implications

There are numerous issues regarding the architecture of WEB sites that support e-commerce, this section will concentrate on addressing those that are affected when we wish to make the site multilingual. Also, the view will be taken that our main aim as ISPs is not to so much to create WEB sites for ourselves, but to create an environment in which our customers can create WEB sites.

2.3.2.1 Advertising and fixed text

Firstly, we need to make sure that the user can find our WEB site when looking for a product that we sell. This means that we need links to our pages to appear on the most used portals and we need to make sure that the keywords that can be extracted from our pages are appropriate. The titles of the pages need to be chosen very carefully as does the first paragraph of text. META tags need to be inserted which will make sure that the pages are appropriately classified by search engines that use these pages.

The user's first encounter with a vendor's site will be critical. The user will be presented with information which must re-assure them that this company is good to do

business with. From an architectural perspective though, this presents no special problems. This will be covered in the section on "Information Provision".

2.3.2.2 Catalogue

If the company has a wide range of products, or its products are highly configurable, then some kind of cataloguing system will be required. Typically, product information and pricing will be stored in a database. It may also be the case that the products themselves are localised and therefore may be considered as configuration of either the home market product, or some abstract, local-independent product.

Of course, product descriptions and pricing are locale-sensitive and therefore have an impact on the architecture. In general we may assume that some information is common to all locales and some is specific to each locale. The cataloguing system must allow separate control of locale-independent and locale-specific information (and perhaps some information specific to the home market).

In some cases it may be that the locale-specific information can be calculated from either another locale or from locale-independent information. For example, it may be that a US company produces goods which it also sells abroad. There may be a company policy for determining the price of goods in foreign markets which can be implemented as an automatic process. The architecture then needs to be able to support such rules which can calculate the values of certain fields either off-line or in real time.

It may be that, in some cases, some form of automatic translation is acceptable. In which case the ability to integrate a machine translation system in the creation of the locale-specific data from home product or local-independent data will be required.

Alternatively, it may be possible to generate some information in multiple languages directly from data. For example, the kinds of features that have to be specified when buying a car are likely to be common for all locales but, of course, the translations of the names of these features will be different. In these cases, it may be possible to produce multilingual text automatically using pre-translated templates, or using a machine translation system that is configured specifically for that task.

General principles of localisation become important here. There needs to be a clear separation of locale-specific and locale-independent information (see PIR 3.2 for an overview of localisation).

This has implications for the structure of the catalogue database, and the kinds of tools that will be required to maintain the catalogue. The database must, of course, be able to handle international characters. If the database can only store single byte characters, then it may be possible to use UTF-8 encoded Unicode. However, the application that accesses the text in the database will need to know how it is encoded in order to display it correctly.

Fortunately, the two main browsers in use (Netscape Communicator and Microsoft's Internet Explorer) can both display UTF8 encoded Unicode directly.

There are still numerous pitfalls though. For example, the Microsoft Access cannot store certain characters (e.g. 'ï') even though they are within the normal ASCII range and a UTF-8 encoded sequence will need to store some very unusual characters.

If the product range is large, then some kind of searching and classification method will be needed. Keyword searching is, of course specific to locale, so indexing terms need to be generated in a locale-specific manner. This is discussed further elsewhere.

2.3.2.3 Capturing the sale

Once the user wishes to buy a product, they may either do it on-line or by contacting a human being. Either way, the user details will need to be captured. User interface is described in another section of this document.

Verifying user details, credit rating etc. is not an issue that involves multilinguality and so will not be discussed in this document. Similarly security (although a vital issue for e-commerce sites in general) does not impact on the multilingual aspects of the architecture, and therefore will not be discussed in this document.

2.3.2.4 Considerations outside the scope of this document

There are of course many more issues relating to the architecture of an e-commerce site. For example, the delivery of the goods may either be on-line (if the goods are information or software etc.), or they may need to be delivered by some physical means.

There are various legal and financial issues. For example, how should VAT be recovered on goods that are delivered on-line? Fortunately, these and other legal issues are outside the scope of this document.

Security is of course an extremely important consideration for e-commerce WEB sites. A general description of an e-commerce site would not be complete with out a comprehensive section on Secure Electronic Transactions (SET) and Secure IP. However, since these do not impinge in the multilingual aspects of WEB sites, they will not be discussed here.

2.4 Description of the user interface functional element

2.4.1 General description of the user interface functional element

The User interface functional element was identified to provide a powerful and contained structuring concept to deal with all aspects related with form filling and user profile generation and management in a multilingual information context.

The following text presents the main features covered by this functional element and its additional architectural components or extensions required by multilinguality when compared to a monolingual web site approach.

The main features to be consider in this section are:

1. Forms design
2. Generation and management of user profile
3. Personalisation
4. E-mail handling
5. Privacy and Security

2.4.1.1 Form filling

The most common way to interact with the user is through HTML forms. Form filling covers the situations when the user fills in, selects entries for, or modifies labelled

fields on a form presented by the system. Using tools like JavaScript or CGI, a form on a Web page can interact with, and react to, user input.

The aspects of multilinguality and localisation must be carefully considered in the conception and design of the forms. For instance, to a Portuguese user the “state” field, used in American forms, does not make any sense: an Italian user thinks instead in terms of “province”.

For a multilingual web site, one can see two approaches to deal with the form design and presentation: using pre-defined and translated templates for the site linguistic options or the (automatic) translation and the adaptation of the form is made dynamically, as a function of the language option. In both approaches it is necessary to pay special attention to all the issues related with handling characters, writing rules and data exchange (see PIR31, “Multilingual text processing”).

Another aspect to consider is the form processing and storing. The interaction with the user, namely in the input validation (formatting errors, missing or invalid information, and so forth), must be done according to the linguistic option.

2.4.1.2 Generation and management of user profile

The definition of the user profile results from the importance of knowing and understanding the needs and preferences of the user. This is particularly important for the commerce-oriented sites.

The generation of the user profile can be achieved requiring the user's involvement, typically through filling out a form or following a decision-tree set of questions. The direct request of user information through a form depends on site function and objectives.

It is also possible to get information from the user in an indirect way, by using cookies or by looking at an IP address and serving up content based on the user's browser.

The user information collected by either direct or indirect means must be stored in a database. The analysis of this information can be used to define and manage the user profile. User Profiles can be used to customise Web content for users logging onto the Web server.

2.4.1.3 Personalisation

Personalisation allows sites to offer personalised and possibly dynamic adaptation of content structure, navigation primitives, and presentation styles automatically to different users, based on the user profile. Personalisation can draw from many types of contents including files, databases, newsgroups, e-mail public folders, and so on.

Cookies are especially useful in personalisation in conjunction with dynamically generated pages. For example, if the site delivers information on a variety of subjects, one could use a cookie to store data about which topics the user has reached. When the user returns to the site, we can generate dynamic pages that highlight those topics.

As a good example of personalised sites we can refer the Amazon.com, one of the most watched e-commerce sites on the Net. It also has one of the best recommendation engines online. The system analyses past purchases and posts suggestions on the shopper's customised recommendations page. Amazon enables both active and passive behaviour from the user. Amazon's Recommendation Centre (dynamically served to returning customers on its home page) looks at users' past purchases to suggest books they might like. For new users and the relatively few

readers who want to actively rate content, Amazon also offers its BookMatcher, where visitors rate a few books they like and then receive recommendations themselves.

2.4.1.4 E-mail handling

For the site server it is possible to have an application to handle e-mail, so that e-mails that are received are processed 'automatically'. The application could extract the relevant information to store or process. It is also possible to generate and send a message to the user (useful on e-commerce sites, for advertising, to control purchase or confirmation orders, etc). In a multilingual web site context, such mail handling application should also be able to consider the language option of the user. Again automatic translation of email messages should be preferred to pre-defined email message templates for the various languages supported by the site (more complex content and interaction management).

2.4.1.5 Privacy and Security policy

Most personalisation technologies require some users personal data. It is critical that the users' information be transfer and stored with security. It is also important to inform users on how it will be used before collecting it.

With the cookies approach the procedure is the same: every site that gathers data from its users should include a disclosure statement with specific information about how the site uses cookies.

Sometimes the appearance of security is as important as the fact that it really exists. The information stored in a cookie is safe from prying by other Web sites, but users don't necessarily understand that. So it is essential to avoid storing any data that's even potentially embarrassing or costly to the user – like passwords, credit card numbers, or purchase authorisation numbers. The sensitive information must be saved in a server-side database.

Security also plays a role in the decision to run an application as a client-side script or a server-side CGI. When we run applications on a Web browser, the server is safe because the client application can't access server resources. But a poorly written server-side CGI application can open the site to malicious visitors. For example, if a user enters their name into an HTML form and the CGI prints that name, someone could potentially use that form as a back door to execute their own server commands, such as listing password files.

2.4.2 Motivation for multilinguality in user input

With the growing importance and marketing dominance of Internet web sites acting as portals (e.g. anchor sites for the users) the relation with the user is rapidly becoming the key element for success on the Internet presence of business companies. The personalisation factor is driven by the user or customer and techniques based on manipulation of the user profile information are enabling portals to deliver the information that the customer needs in the way the customer wants it to be presented. For such dynamic and global environment over the Internet, multilinguality can only be seen as a powerful capability to address the user in its selected language and that needs to consistently cover all the interactive communication experience of users with the web sites (both visual and written information).

2.4.3 Architectural components required for user interface

The user interface functional element requires several architectural components, technologies and tools, to provide multilingual functionalities:

- Tools for language identification.
- Multilingual tools for web pages creation, supporting multiple character sets, language mark-up and the linking of documents in different languages.
- Web browsers with multilingual facilities for navigation on multilingual web sites.
- Scripts with multilingual capabilities.
- Databases with multilingual capabilities supporting different character codes.
- Applications and tools to process (accessing and manipulation) the databases content with the knowledge of character encoding set order to display it correctly.
- Tools to provide web pages dynamically from the information stored in the database and to allow the generation of notifications to the user.
- Tools to generate pages which support tags and text for each language and language mark-up, and have the capability to generate and customise the pages
- Encoding of the information in portable formats such as XML (see PIR 3.1)
- On-line and off-line Machine Translation, in association with some linguistic resources to increase translation accuracy, like glossaries, spelling dictionaries, phrasal books, terminological databases in different domains, word taxonomies (names, cities, countries, ...).

2.4.3.1 Generation and management of the user profile

User profile generation

The direct request of user information through a form depends on site functions and objectives and is best described in the form filling point. Here, we only refer to the indirect way to get information from the user.

The generation of the user profile can be achieved by using cookies or by looking at an IP address and serving up content based on the user's browser.

A cookie is an HTTP header containing a string that a browser stores in a small text file on the user's hard drive. The file is saved in the Windows/Cookies directory (for Microsoft Internet Explorer) or in the Users folder (for Netscape Navigator).

The HTTP protocol does not have the concept of session, since it is fundamentally stateless: HTTP cannot identify the user or which other pages have been delivered to that user. The purpose of using cookies is to help web sites overcome this fact by providing a solution to trace users navigation on a site.

That kind of information is important because, unless you know something about the users, it is not possible to build a web site that responds to customer preferences, language option or even the browser and operating system used.

Cookies can be created by using a CGI (Common Gateway Interface) program or JavaScript. JavaScript is simpler and does not require server-side programming. As it runs on the user's machine, no information can pass from the client to the server. To pass such information we must use a CGI program to store information in a server

database or run server applications (to automatically generate and send email, for example) and also to collect information about the Web site's visitors.

General Aspects

- Cookies and other indirect ways of getting information from the user can register information about the language and character encoding.
- Capture the users system configuration (environmental variables) using client side scripts to personalise the access and navigation.
- In the process of registration of users and users preferences, store and retain the options and settings made by the user. This kind of information - context variables - can be supported by the use of cookies or session tracking variables.
- The scripting languages like Javascript or VBscript can be used for the validation of the data introduced by the user during form filling. To store and retain the information about the user profile obtained in a direct or indirect way cookies or session tracking should be used. These cookies and/or session tracking variables can be created and processed with CGI, Servlet or scripts.
- The implementation of scripts in a multilingual environment should follow some principles:
 - separation of the strings from the code,
 - clear identification of the variable names (avoid common words),
 - avoidance of breaking sentences,
 - definition of a coherent use of variables that store text translatable text.

User profile management

To store in a persistent way the user profile information it should be used databases with multilingual capabilities supporting different character codes. There are several databases, like Oracle 8i or the Microsoft SQL Server 7.0, that follow these requirements.

This information can also be stored in files with a suitable format such XML. XML is a powerful tool for data representation, storage, modelling, and interoperation and has a huge potential to be used in the context of multilingual web sites.

General Aspects

- Create a user profile repository with all the relevant information – language, character encoding, and user preferences to define the user profile, with the capability of handling international characters.
- Avoid the use of static fixed-length storage or ensure that is enough to the different languages.
- Set up applications and tools to process (accessing and manipulation) the repository content with the knowledge of character encoding set
- Tools to provide web pages dynamically from the information stored in the database and to allow the generation of notifications to the user. There are several tools like the Microsoft Active Server Pages (ASP) which uses mechanisms like ADO (Microsoft ActiveX Data Objects) or OLE DB to access the data stored in the database. The Java Servlets using the JDBC package to access the information stored in the database can also generate web pages dynamically. The languages for

the development of CGI programs like Perl also provide modules to access information stored in databases.

- Tools with the capability of retaining the application status between two consecutive user requests. The session tracking variables available in ASP or in Servlets retain the state of the client between consecutive requests during a period of time. The cookies can also be used to support this issue.

2.4.3.2 Form filling

Form design

The creation of a web page containing a form can use predefined templates. These templates must contain all the information necessary to the construction of the web pages.

The Cascading Style Sheet (CSS), as part of the dynamic HTML, is a simple mechanism for adding style (e.g. fonts, colours, spacing) to web documents. CSS may prove very useful in the context of the multilingual sites because it permits the separation between the document appearance and its content. This separation allows the development of simpler HTML documents, containing the translatable data, which facilitates the translation process.

There are several possible approaches to generate the forms in a multilingual environment.

- The first approach is to have a repository with the forms for each language. The layout and the linguistic and local dependent information are previously translated (off-line) and the pages are constructed with an HTML editor.
- The form can be created dynamically accessing several predefined templates available in a repository, supported by a database or XML files, with the information about the contents and the layout design. It is possible to define a format supported by tables or by XML tags which allows a separation between the layout information and the content of the form. The layout information contains all data to construct the form (location of the labels, input fields, buttons or images, length of the labels, input fields, buttons or images) and is defined according to the particular language characteristics. The repository must contain all information in each language supported by the site.

All the tools like ASP, CGI or Servlets may be used to generate dynamically the web pages.

- Another possibility is the existence of a single predefined template in a specific language like English. The data is translated on-line by a translation machine, according the linguistic option of the user and the page is dynamically generated. The tags and formatting elements of HTML pages must be preserved through the automatic translation process. In this situation the definition of the template should take into account the characteristics of the different languages that will be supported by the site. For instance it is necessary to provide enough space for the translatable fields to accommodate them after the translation.

Currently it is not possible with the translation tools available in the market to guarantee a high quality translation. However there are some areas where it is possible to obtain a high quality translation using machine translation tools. One of those areas is the Web page form, since usually the translatable data in this functional element is most of the times very simple (words or short sentences).

There are some companies like Systran and iTranslator which provide products in the automatic translation area. These products contain an interface to be easily integrated with applications developed by third party developing entities. The Systran machine translation engine provides an interface in Java and the iTranslator machine translation engine provides an OLE DB interface.

The design of a form in a multilingual Web site should have in consideration several aspects:

- Whenever possible the text should be separated from graphics;
- The size of dialog box and buttons must be adjusted to the different language based on character expansion (generically 30% more than English);
- Adjust the space and format of the layout to the user input, considering character expansion, presentation of numeric data, measurements units, currency, accented characters;
- Adjust dialog box, radio buttons and menus sizes to accommodate the translated texts;
- Ensure the consistency of layouts – margins, tables, graphics,;
- For alphanumeric text entry, avoid improper settings of the wrap attribute for Web form text fields.

Form filling and processing

The user details will need to be captured and processed. An aspect to consider is the user-interaction error: formatting errors, missing or invalid information, and so forth.

Essentially, there are two basic approaches to capture errors: on the client side (browser) and on the server side. In the first case, the form elements are checked with a Javascript function before the form is submitted to the server and the user is informed which fields are erroneous and why. For server side errors the processing necessary to identify the error is done on the server.

Capturing errors on the client side provides a more quick response (if the Web server is involved in processing the page for identifying errors, it is necessary to submit the page to the server, process the page to capture the error(s) and return the "error" page back to the browser) but we need to use scripting languages which may be either unsupported or inadequately supported by some of the older browsers.

General Aspects in form filling and processing:

- Adjust the space reserved to user input. to the different language; generically 30% more than the corresponding length in English,
- Select characters set/codepages with different codification to accentuation characters,
- Use character encoding identical to the received in the data,
- Define hyphenization rules,
- Define data validation mechanisms and inform the user about special input methods.

User presentation aspects:

Here there is a series of helpful hints for writing pages which are clear and appealing to users.

- The sentences should be concise and unambiguous. Avoid ambiguous solicitations or classifications – what is the “state” or the “middle name” ? what name you write first, the family name or the proper name?
- Use fonts that support special characters.
- Use descriptive field labels.
- Prepare pages HTML-based using forms and hidden fields to enable the dialog with the user.
- Define the user presentation, namely interface language, user layout preferences (based on user profile), the presentation of the results of users requests.
- Create a repository of all the presentations – HTML files, templates for forms and e-mail – databases.
- Ease user input and validation: mark required information, show formatting requirements (e.g., date must be in yy/mm/dd format).
- Different formats (names, dates, currency, ...) must be supported both for entry and display of information.
- Define the general interface and language navigation – all the pages with the linguistic option or, after the initial decision, all the pages in the same language.

2.5 Description of the multimedia functional element

2.5.1 General description of the multimedia functional element

Today it is very difficult to speak about the Web without "thinking multimedial". Even if the W3C Consortium struggles to remember everyone that the most important thing is accessibility for everyone, in practice it is very common to find sites full of graphics, music and videoclips. Moreover, a version of the site for visually impaired people could use a text-to-speech system to present the information: this could also be considered part of the multimedia component.

2.5.2 Motivation for multilinguality in multimedia functional element

In a certain sense, multilinguality is not a key point for the multimedia functional element. Indeed, images - both still and in sequence - and movies are not language-dependent: ditto for MIDI music. There is however some use for multimedia components: for example, if sites want to add voice output as comment to a video presentation, or just as a friendly way to present some feature of the site. Another possibility is to add language-dependent text to the pictures, either as cartoons or as plain text (not captions, of course, since they may be treated as normal text). GIF files with words and sentences with special fonts fall in this category.

2.5.3 Architectural components required for multimedia information

If multimedia information is used within the site, we could have several architectural problems. Here there is a list.

- Multimedia files tend to be large: this means that it is not feasible to have a copy of them in each language section, if the pages are static. The simplest way to cope with this is when they are the same for each language version: in this case, it is

possible to have a directory where these files are stored, and to have the files inserted within tags in the HTML pages.

- In the case the multimedial content is language-dependent, but most of the file is not (for example, a MPEG file with audio parts in several language) it could be necessary to generate the complete message on-the-fly, or to resort to have many copies of that file. Besides the problems of space, it must be kept in mind that the tools for synchronising HTML links are different from those required for synchronising HTML pages.

2.5.4 Relation of multimedia with other functional elements

Multimedia component should be rather independent from information provision, since what applies to a monolingual site should also apply to a multilingual one.

There is some interaction with information retrieval component: the simplest way to cope with this is to label the multimedia content so that there is a text description which can be input to the current information-searching systems. This approach has also the side effect that the web pages could be exploited also by people with suboptimal web connection, and this gives a good overall impression to the site.

Probably there is no relation between multimedia component and e-commerce.

The strongest interaction for the multimedia element should take part with the user interface: is the input is vocal, for example, multimedia is of course necessary, and it has to be taken into account. But even if this were not the case, input in a specific language could lead to different content presented, if the creator of the site thinks that language is associated to the nationality.

3 Global architectural requirements for multilingual web sites based on functional elements

This section summarises the global architectural requirements for multilingual web sites management and exploration. These requirements are based on the individual architectural requirements that have been identified for each one of the functional elements described in the previous section. Results are presented in the form of a synthetic table where colons correspond to functional elements and lines correspond to architectural components. The architectural components comprise tools, procedures, static and dynamic components that are necessary to be added or adapted to monolingual web site architecture to make it multilingual. Some features are also considered in building and managing multilingual web sites. As it was said before, both of the following approaches fit in these figures: localisation of monolingual web sites, and creation of new multilingual web sites.

Moreover this table presents possible relations between functional elements in terms of architectural requirements. These relations are materialised by architectural components that are common to functional elements.

Architectural components are simply listed in the table. They are described more completely in the sections related to the corresponding functional elements.

Symbols used in the following table are :

IP: Information Provision functional element.

IR: Information Retrieval functional element.

EC: E-commerce functional element.

UI: User interface functional element.

MM: Multimedia functional element.

AC: Architectural Component.

X: Marks that an architectural component is required by a functional element.

| Architectural component (AC) | Type of AC | IP | IR | EC | UI | MM |
|---|-------------------|-----------|-----------|-----------|-----------|-----------|
| Languages and cultures | Feature | X | X | X | X | X |
| HTML / XML editor for multilingual documents | Tool | X | | | | |
| Off-line machine translation system | Tool | X | | | | |
| On-line machine translation system (document translation) | Tool | X | X | | | |
| On-line machine translation system (query translation) | Tool | | X | | | |
| Translation memories (content) | Dynamic component | X | | | | |
| Translation memories (queries) | Dynamic component | | X | | | |

| | | | | | | |
|--|-------------------|---|---|---|---|---|
| Language tags | Static component | X | X | X | | |
| Document/character coding standards with conversion mechanisms | Dynamic component | X | X | | | |
| Sets of fonts for different languages | Static component | X | | | | |
| Multilingual terminology database | Dynamic component | X | | | | |
| Multilingual phrase book | Dynamic component | X | | | | |
| Multilingual thesauri | Dynamic component | X | | | | |
| External localisation process (human translators) | Procedure | X | | | | |
| Tools for extraction and restoration of text from HTML | Tool | X | X | | | |
| Text generation (information) | Tool | X | X | | | |
| Text generation (selling reports, etc.) | Tool | | | X | | |
| Text summarisation | Tool | X | X | X | | |
| Multilingual authoring systems | Tool | X | | | | |
| Multilingual web browsers | Tool | X | X | X | X | X |
| Internationalisation for future localisation | Procedure | X | | | | |
| Several languages in top-level pages of web sites | Feature | X | | | | |
| Search engines | Tool | | X | | | |
| Language identification tools | Tool | X | X | X | | |
| Thematic analysis | Tool | X | X | | | |
| Cross-language Information Retrieval | Procedure | | X | | | |
| Information extraction | Tool | | X | | | |
| Query analysis | Tool | | X | | | |
| User profile information | Feature | | X | | X | |
| Databases (in general) | Dynamic component | X | X | X | | |
| Catalogues (of products) | Dynamic component | | | X | | |
| ISP (Internet Service Providers) | Procedure | X | | X | X | |

| | | | | | | |
|--|-------------------|---|---|---|---|---|
| Advertisement procedures | Procedure | | | X | | |
| Meta tags for appropriate classification of web pages | Dynamic component | | X | X | | |
| Meta Keywords tags that contain keywords in the appropriate language | Dynamic component | | X | X | | |
| Selling process | Procedure | | | X | | |
| Security process | Procedure | | | X | | |
| Conversion processes (currency, etc). | Procedure | | | X | | |
| Customer details | Feature | | | X | X | |
| Multilingual forms | Dynamic component | | | | X | |
| User personalisation | Feature | | | | X | |
| Multilingual email | Tool | | | | X | |
| Privacy and security mechanisms | Procedure | | | | X | |
| Graphic components | Static component | | | | | X |
| Multilingual texts inside images | Static component | | | | | X |
| Multilingual voice support | Tool | | | | | X |
| Access for visually impaired people | Procedure | X | | | | X |
| ASP (Active Server Pages) for dynamic Web server applications | Dynamic component | X | | | | |
| CGI scripts (Common Gateway Interface) | Procedure | X | | | | X |
| SSI (Server Side Includes) | Procedure | X | | | | X |
| JavaScript / VBScript for Applet local run | Procedure | X | | | | X |
| Others | | | | | | |

4 References

- [1] Vora, P., 'Managing errors in transaction-based Web Applications'
http://www.sandia.gov/itg/newsletter/dec99/workshop_errors.html
- [2] Fraternali, P., 'Tools and approaches for developing data-intensive web applications: a survey'. ACM Press (1999).
- [3] Berman, M. et al., 'E-Commerce solutions for the enterprise'. Microsoft. (1999)
<http://msdn.microsoft.com/workshop/>
<http://www.microsoft.com/siteserver/commerce/default.htm>
- [4] White paper from Lernout & Hauspie "Preparing Web sites for streamlined localization", 1999 <http://www.lhsl.com/>
- [5] <http://www.multilingual.com>
- [6] "JDK 1.1 internationalization overview"
<http://java.sun.com/products/Jdk/1.1/docs/guide/intl/intl.doc.html>
- [7] " The Java tutorial"
<http://java.sun.com/docs/book/tutorial/>
- [8] Douglas W. Oard, Serving Users in Many Languages, Cross-Language Information Retrieval for Digital Libraries, D-Lib Magazine, ISSN 1082-9873, December 1997.
- [9] Mark W. Davis and William C. Ogden, "QUILT: Implementing a Large-Scale Cross-Language Text Retrieval System," in Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Philadelphia, July 1997, pp. 92-98.
- [10] Mark W. Davis and William C. Ogden, URSA, The Unicode Retrieval System Architecture, <http://crl.nmsu.edu/Research/Projects/tipster/ursa/index.html>.