

# BACHELOR THESIS

Internationalization and Multilingual Web Development  
with practical application to “Historical Trees and Forests”  
web application

Conducted within the bachelor program  
Information Technology and Systems Management  
Salzburg University of Applied Sciences

Submitted By:  
Alex Taylor



Head of Department: FH-Prof. DI Dr. Gerhard Jöchl  
Supervisor: Karl Entacher

Salzburg, February 2011

## Affidavit

Herewith I declare that I have written the present master thesis fully on my own and that I have not used any other sources apart from those given.

---

Location, Date

Student name and signature

# Contents

1. Introduction
2. Multilingual Web Application Requirements
  - 2.1 Overview
  - 2.2 Web Standards
    - 2.2.1 Unicode
  - 2.3 Languages to support
  - 2.4 Separation of Interface and content
3. Site Structure
  - 3.1 Separation between interface and content
  - 3.2 Different types of multilingual sites
  - 3.3 Design Considerations
    - 3.3.1 Colors
    - 3.3.2 Default Language
4. Actual Translation
  - 4.1 Which Content should be translated?
  - 4.2 Supporting Different Dialects
5. Practical Application
  - 5.1 Introduction to the Application
  - 5.2 Joomla
  - 5.3 Integration of Multilingual features
  - 5.4 Choosing the default language
6. Conclusion

# 1. Introduction

At its heart, the World Wide Web is fundamentally an intercultural and multilingual experience. With such a vast array of different languages and communication styles, it is increasingly important that we are able to provide each group of people with this bounty of information that is contained within the internet. Languages are the most prominent example of providing support for different countries and regions, as they are visible and have clearly defined structures. We will come to see that not only are languages important when creating web applications, but many cultural, social, and geographic factors come into play.

Today the internet is expanding to greater and greater limits, and will continue to demand more solutions to problems and challenges that arise. Since many websites currently exist in just one language, they are seeing the need to transition into multilingual frameworks. This can be quite challenging, even more so than creating a new site from the ground up. This makes it extremely important to account for the possibility of a future transition to multilingual support when designing a new project.

Early on, developers had to come up with ways to create multilingual interfaces from scratch, and it was quite a difficult process in the beginning, with many options available, some being easy to develop, but hard to maintain, or vice versa. Now it is getting much easier, with many different solutions being available. One type of platform that is widely taking the stage is Content Management Systems. These systems help users create and manage content, and often even a community, online. In the beginning, they were only popular with small groups of developers, and small businesses. Now they are taking the industry by storm, and even large web corporations such as Popular Science, Sony Music, and Amherst College are joining in on CMS solutions, according Karen Coombs in her article "Drupal Done Right."

Most of these systems provide convenient ways to manage web site's architecture and content added by users and editors of the web site. Although these systems are developed by international communities, multilingual features are not always integrated into them. These systems are used in situations with widely different needs, from powering blogs to major government web sites. Many of these different use cases share the requirement to support multilingual features, even if the exact needs in these use cases are different. A single user blog could have content in different languages, while a complex government site requires parts of its web presence in multiple languages at once.

I will begin my thesis with an introduction to the requirements of a multilingual site, and what needs to be taken into consideration when transitioning. Also I will look at the standards that come into play, and what role each provides for multilingual display of web applications to be possible. Then I will move into the technical part of what tools we can implement, and the actual process of implementation. Then I will transition away from the theory part of this thesis, and into the practical application of the work we actually did for our Historical Tree Mapping web application.

## 2. Multilingual Web Application Requirements

### 2.1 Overview

Before I start with the requirements I think it is important to clarify the terms Web Site, and Web Application. Although the World Wide Web was designed to offer interconnected web sites to visitors, more traditional applications also found their way onto the internet, which resulted in web applications. There are slightly different descriptions for these two terms. Web sites can be considered collections of interlinked web pages managed together that allow you to read their contents. Web applications, on the other hand, are applications accessed through a web server that allow you to do something. While there are clear examples of both, most web sites are now a mix of pages with application-like functionality. The multilingual principles I will discuss here are equally applicable to traditional web sites and web applications, so these terms are used interchangeably.

To begin, I believe it is necessary to establish what exactly a multilingual web site is. According to the World Wide Web Consortiums (W3C) web site, there are two different classes of multilingual sites, one of which is a Multilingual Web Site, and the other is International Web Site. A Multilingual Web Site is a web site available in multiple languages. Several countries, for example Canada and Belgium have more than one official language, so a multilingual web site is not necessarily an international one. An international Web Site is A web site intended to be used internationally. This type of web site is not necessarily a multilingual one because residents of multiple countries can speak the same language.

As you can see, a web site can both be multilingual and international, serving people in multiple countries with different languages available. Unfortunately language alone is not

always enough to consider when presenting information to a web site visitor. The World Wide Web Consortium also takes into account the geographic location of a particular user base. For example in the United States we use miles as a measure of distance, but in other countries, even English speaking ones, they may use Kilometers for their unit of measurement. The W3C defines this particular set of differences as a Locale. According to the W3C, in computing the locale concept refers to a set of rules for presenting information to a user. Locale includes date formatting, currency, the language variety used, order of sorting and so on.

This brings us to a next set of terms that comes into play, Internationalization and Localization. The W3C defines internationalization as follows: internationalization is the design and development of a product, application or document content that enables easy localization for target groups that vary in culture, region or language (locale). And in turn, localization is the adaptation of a product, application or document content to meet the language, cultural and other requirements of a specific target market (locale). So to clarify, internationalization is just adding the ability support different locales, and localization is the actual act of converting the site to the selected locale.

Now that I have covered the fundamental basis of a multilingual web site, I will move on to some of the technologies and standards that are required to make multilingual sites possible.

## 2.2 Web Standards

When building multilingual web sites, we need to first consider technical requirements and possibilities. Web standards (recommendations and specifications) define our communication means between web servers and clients, so these must be examined first. It is important to note that these standards are applicable to single language web sites too,

although they are not widely known in English speaking areas of the world because the defaults provided are adequate there, so there is no immediate reason to think of these building blocks, however in our case, it is important to look at these.

The first standard we will look at is internationalized resource identifiers, otherwise known as IRI's. To begin, web addresses are typically expressed using Uniform Resource Identifiers or URIs. The URI syntax restricts addresses to a small number of characters: upper and lower case letters of the English alphabet, European numerals and a small number of symbols. Unfortunately a URI does not allow for non-English characters, which limits its usability internationally. Internationalized Resource Identifiers allow for domain names and paths to contain any Unicode character (Unicode will be discussed later on), thus allowing for fully localized web addresses.

For IRI's to work there needs to be support tied in to the web browser, the server hosting the site, and the underlying protocol, such as HTTP. In the past this has not been widely adopted, but now many web browsers are coming out with default support, and many servers are already capable of handling it. A basic example of an IRI could be for a website called `http://märz.at`. Now normally, this would not be allowed, because the umlaut is not supported by the URI standard, but IRI makes this possible by Using Internationalized Domain Names in Applications (IDNA's). Internationalized Domain Names in Applications are mappings of Unicode strings to special US-ASCII strings, which map to IP addresses in the standard domain name system. If we return to our above example, of `http://märz.at` and convert it IDNA then we would have `XN--MRZ-QLA.at`. This is what the browser does automatically, and allows for a transparent experience of the user. This feature is a great convenience for users because it allows for localization that has not been seen before with many different languages. Next, once we have made a connection to the server, we need to be able to communicate in the desired language, and in the beginning there was only support for English characters, so we needed a



solution for encoding characters. This standard is called Unicode, which brings us to our next section.

## 2.2.1 Unicode

At their most basic level, computers do not understand letters; they only understand numbers, in order for them to display readable text we had to develop a system for translating the binary information to readable characters. The way we did this was with character sets, basically an index that mapped symbols and letters to their respective binary numbers. This system only worked for one particular region however, and because development was taking place all over the world, there were many different character sets. This caused a big problem when computers started communicating with each other on the internet, and technologies were being shared across different parts of the world.

The remedy to this problem, we developed an all-encompassing character-encoding scheme to replace them all, Unicode. Considered by the W3C as “the base architecture of the web,” Unicode stands for Universal Character-encoding system, and takes on the job of providing encoding support for almost entirely all of the world’s languages and characters. Currently Unicode uses a 4-byte scheme to represent each character, providing enough code points for over 1 million characters and symbols. This allows for information sharing all across the world, providing real multinational and multilingual resource sharing. This was a major breakthrough technology that helped the internet grow into what it is today. While Unicode provides us with the ability to display and share information across different languages, it is not able to provide any translation or conversion between them. Actual translation and implementation of supporting multiple languages is a much more complicated process.

## 2.3 Languages to Support

When creating a web site, it is important to consider the idea of providing support in multiple languages. It may sound most natural to provide support for only your native language, however there are many factors that you may need to consider before jumping in with a one language approach. The dominance of English in the Internet needs no arguing for, as Anthea Gupta puts it in her essay “The Internet and the English Language.” She also states that computers are English-oriented, the vocabulary of computing and of the Internet is overwhelmingly English, and most of the texts that are accessed through it are in English. As the only way a web site can survive and become successful is to generate traffic, it becomes more critical that the designer reach out to the largest user base possible. This typically leads to many web sites supporting English as a major language. The real challenge comes when you need to decide another language to support. Things such as geographic location as well as relevance to the intended languages user base must be taken into consideration. Another thing that must be considered is the number of languages that can be supported. The cost effectiveness can play a major part in the decision making process, as it takes a lot of time and effort to create, and continually support throughout the lifespan of the site.

## 3. Site Structure

### 3.1 Separation between Interface and Content

When designing a web site with multilingual requirements, separation of the interface and content becomes vital to the success of the project. First we need to define what interface and content really are. I will start with the interface, there are a set of application provided interface elements in some systems written in the “factory default” language. Then there are

possible modules or plugins added to the system with their own interface elements. Site designers refine and extend the built in interface of the application to be better suited for the actual web site's needs. Finally there are interface elements specified by the site administrator. Prime examples of interface elements are menu items and other navigational helpers, which are required to be customized by the maintainers of a web site. Content is much easier to define, as it is the actual information provided by the web site, for example on a blog, the content would be what the author has actually typed, and the interface would include things such as a "Next" button or a "View older posts" button that allows the user to navigate the site.

These two things become extremely important, because while it may be easy for a translator to translate the content of a website, it could be extremely difficult to change the language of interface elements because they are written into the code that makes up the web site. Also it can get even more complicated when there are items such as images, that are both part of the interface, as well as the content, when there are images that have words on them. You can see that translating a picture of words could get very difficult if you have to re-design the entire image collection of a web site just to change the languages.

There are many things that should be taken into consideration when designing a site or application that involves separation of interface and content. The first of which is to make sure that you use CSS extensively. For example when using HTML `<b>` or `<i>` tags for emphasis, and the desired language is Japanese, the characters need to be written in bold or italics. However the Japanese language uses dots above the text for emphasis or a different background color, instead of bold or italics. If we build on CSS, different style sheets for different languages can easily provide the functionality for emphasized text. This also can also be very important for bidirectional text presentation.

Another design consideration that must be taken into account is preparing for text expansion and contraction. This may not seem immediately clear, but in fact, languages can

greatly differ in word length and paragraph length after translation occurs. This can lead to problems, because it is common to design different parts of web pages to a specified width, such as a navigation pane or a sidebar. If the width of these parts is not adequately chosen, translated versions of the text written into them can easily not fit or leave an undesired empty area, causing problems for the end user. For example, it is very plausible that there could be a view counter below a series of photos, and English the word views is very short just 5 letters, however if the intended language was Italian, the translation for Italian is visualizzazioni. As you can see, if you created the site just for the English version, there could be many problems that occur when the second language is implemented.

One more thing that needs to be considered when taking into account separation of interface and content is cultural differences. One good example of this is images on a site, lets say you had a confirmation page on your site that displayed a photo of somebody making the “ok” symbol with their hands. To Americans this is the symbol for everything is ok, however if you site provided service for French users, they could interpret this as “empty” or “worthless” which could become very confusing or possibly insulting for them.

Now that I have clearly defined the difference and importance of content and interface, let’s take a look at the different approaches that you can take when designing a site for multilingual use.

## **3.2 Different Types of Multilingual Sites**

Now I will take a look at the different possible options that a multilingual site can take, starting with the easiest and most basic, and moving up to the more complex.

The most basic setup for a web site is a factory default English-only site that uses English for the content and interface. This is extremely easy because the language of the html and other web protocols are all in English.

Another approach could be a site that supports US English as well as British English. Changes could include different site design or different wording for text accounting according to the different language requirements. This approach is easy because it does not need change the interface language, and only the content needs to be changed.

An approach of having just a monolingual site in any other language than English is possible. This requires that the text of the interface be completely translatable and that the language of the resulting site be configurable so the generated web pages show the right content with the proper language. In this scenario both the content and the interface would need to be changed, but only one language needs to be supported.

The last, and most interesting setup is when a web site needs full support of two or more languages, this is considered the basic multilingual setup. It involves having the Interface be in two languages, and able to switch through the two, as well as having the content hosted in both languages also. There are many sub entities of this that can include sites that complete mirror in each language, or sites that have completely different content. Also there can be sites that are completely different entities, existing as two separate single-language sites. As you can see there can be many possibilities, each with its advantages and disadvantages.

## 3.3 Design Considerations

### 3.3.1 Colors

It may not seem completely obvious at first, but color choice in a web site can have a major effect on the message you are trying to get across to your users. This is because there are many different things people associate with colors, such as if you run an environmental web site in the western world you may have a lot of green on your site, or a water parks web site would most likely include a lot of blue. However, the colors you choose for you site should appropriately reflect your target audience. In his web article “Design Multilingual Website: A beginners guide,” Christian Arno points out that the color red can denote “danger,” “love” or “passion” in Western cultures; whilst it signifies “purity” in India and “good luck” or “celebration” in China. An additional example of these differences is that in many western cultures, the color black signifies “funerals” or “death” but quite interestingly white signifies these same things in many eastern cultures. This demonstrates that while it may seem irrelevant at first, when changing a web sites language, it may be necessary to also look at the color scheme involved at the same time.

### 3.3.2 Default Language Selection

A big decision that needs to be made when choosing to support multiple languages on your site is the default language that is to be displayed when a user navigates to your page. We can use an example of a site with two languages, German and English. Many factors can come into play when trying to decide which should be the default, as well as different technologies you can utilize in order to improve the user experience.

One solution to this would be to have an entrance page to your site, serving as a neutral page that clearly offers an option between the two languages. If the site you are running is an E-commerce site, this could be a very serious option, as it would let them select the desired country from which they would like to make purchases from. Also this could prevent a user from accidentally entering an order that is on the wrong country's site or possibly even in the wrong currency. A disadvantage of an entrance page is that you are preventing users from accessing content directly, and putting a barricade between your content and the user.

Another option is to use an auto-detect feature, which can use the users IP address as a method for determining the originating country, and then defaulting to that home page. This is possible using IP2Country services which are small pieces of dynamic code that run on your home page and determine what the best language should be. However these services are not always accurate and they can lead to users being directed to a totally different language than they can speak. Also I would like to point out that this would certainly lead to wrong language selections if the user were to be using a proxy server or similar technology.

Probably the most common, and easy to implement would be a site that just chooses a language that the designer determines, most commonly English, and provides clear options for easy language selection changes. It is advisable to select the language that most of your users speak as the default.

## **4. Actual Translation**

### **4.1 Selecting content to translate.**

Determining which parts of the website to translate needs to be a very thought out process. It is important to realize that the translation process of the interface component and

the content component need to be taken into account separately. First, let us look at translating the interface part. This is the easier component of the two, because most of the time the interface deals with literal translations, and short, simple phrases. As we looked at earlier, things in the interface that need to be translated are items such as the navigation menu, and any component that deals with navigating or interacting with the web page.

Once the interface has been translated, the next step is to translate the content, this is a much more difficult procedure, because choosing which components to translate can be very hard to determine. Should all of the content be translated? Or possibly some information in one language is completely irrelevant to users of a different language. For the practical part of this thesis, we will be dealing with content that is relevant in all languages that need to be supported, so I will aim to focus on that application.

## 4.2 Support for Dialects

For many languages there are many different dialects spoken of the same language. This can become a problem for developers trying to host multilingual web sites. It is advisable to implement support for different dialects even with the extra resources needed, because it will give the user a feeling of comfort that they may not even notice, but will improve their overall user experience. This introduces the importance of having a native speaker translate the content. When a user visits a site, they expect it to be written by somebody who speaks that language, and it could become extremely distracting or annoying for a user to visit a site that has constant mistakes. Also the fact that some words don't even exist in other dialects of the same language, it could start to cause problems. This happens between German spoken in Germany, and German spoken in Austria, for example if you ran an online shop, and you were trying to sell stools to a visitor from Austria, the visitor may not even find what he is looking for because they are completely different words used for the same thing. You can see the



importance of providing different versions of the same language could begin to be rather important.

## 5. Practical Application

### 5.1 Introduction to the Application

The project that I will be applying the above information to is titled “Historical Trees and Forests.” This web application allows registered users to mark the position of historical trees, natural monuments and forests worthy of protection. It is possible to load images and meta-data for these geographical objects. This system is based on OpenStreetMap <<http://www.openstreetmap.org>> and the Google Maps-API <<http://www.google.com/apis/maps>>. My part in the project is to assist with adapting a multilingual interface to support English and German.

Currently the application is by default written for the German language, and we would like to fully integrate support for the English language as well, as to broaden the possible user base, and increase the size of the project. The main purpose for this application is to track, and keep record of individual trees and forests in a web based environment. The application supports the following features:

- Save Positions of Trees
- Save positions of forests
- Save positions of avenue with trees
- Upload photos
- Upload attachments
- Enter and save text information
- Edit objects

Figure 1 shows the default home page of the web application in its default language (German).

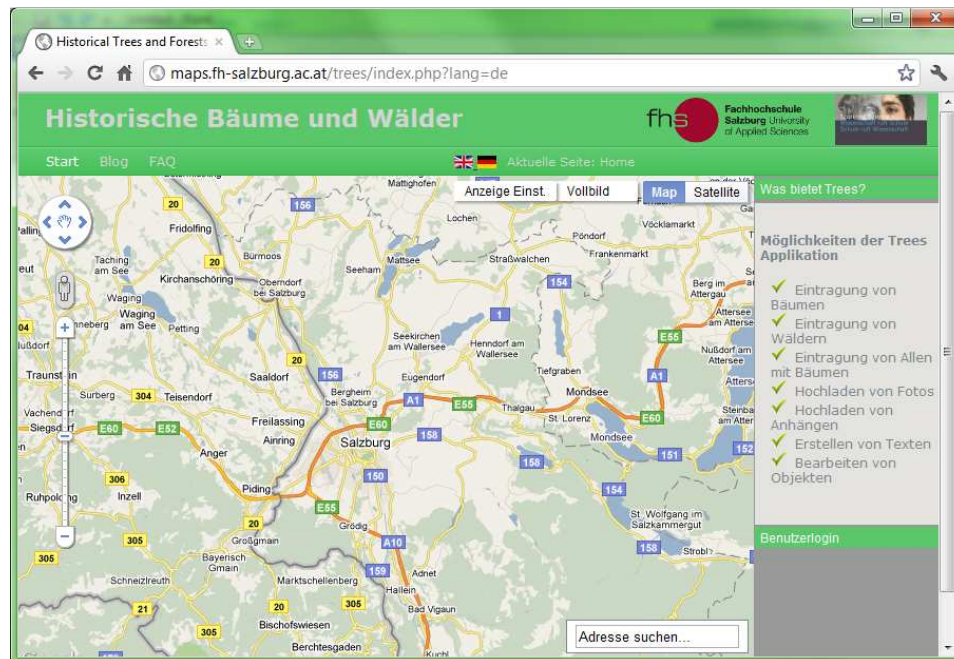


Figure 1

We will implement the feature to view the application in German or English. This will be done in large part by using a Content Management System known as Joomla.

## 5.2 Joomla

Joomla is one of the most well known open source CMS solutions. It is regarded as one of the most user friendly tools and has won several awards including the Packt Publishing Open Source CMS Award in 2007. It is a free, open source CMS, written in PHP that uses object-oriented programming, storing data in a MySQL database, and does page caching, RSS feeds, printable versions of pages, news flashes, blogs, polls, search, and also has multilingual features built in. Packt Publishing reports that “Joomla! is possibly one of the biggest success stories in open source of late. Its first release came in only September 2005 and since then has grown to be one of the most downloaded Content Management Systems on the web.” Also, Rip Empson, a writer for TechCrunch.com states “Joomla has now passed 23 million downloads, and currently stands at just over 23.5 million, to be precise. It owns 10.3 percent of the CMS

market share, and BuiltWith shows it's powering over 1.4 million websites." As you can see, this CMS platform fits right in with our project, and was a natural solution for us. Now let's take a look at integrating multilingual features into our application.

## 5.3 Integration of multilingual features

In previous versions of Joomla, it was required to install a plugin feature to integrate multilingual support, but now, using the latest version of Joomla (Joomla! 1.6) it is not necessary, as the support is now integrated into the system, making it extremely easy to implement. The first step to adding multi language support is to create a "Content Language" in the Language Manager section of the Joomla interface. This defines what a language is, and you will be able to tag different things all throughout the website with this parameter we just created. For our application, we have created two "Content Languages." Figure 2 shows our language administration page, including the German and English languages defined.

The screenshot shows the Joomla! Administration interface for managing content languages. The page title is "Erweiterungen: Sprachen - Sprachen für Inhalte". The interface includes a navigation menu at the top with options like "Site", "Benutzer", "Menüs", "Inhalt", "Komponenten", "Erweiterungen", and "Hilfe". Below the navigation menu, there are status indicators: "Keine Benutzer im Frontend angemeldet", "1 im Backend angemeldet", "Nachrichten", "Vorschau", and "Abmelden". The main content area features a toolbar with icons for "Neu", "Bearbeiten", "Freigeben", "Sperrern", "Papierkorb", "Optionen", and "Hilfe". Below the toolbar, there are tabs for "Installiert" and "Inhalt". A search filter is present with a "Suchen" button and a "Zurücksetzen" button. A dropdown menu for "Status auswählen" is also visible. The table below lists the defined languages:

#	Titel	Titel für Hauptsprache	Sprach Tag	URL	Sprach Code	Bild Präfix	JSTATUS	ID
1	Deutsch	Deutsch	de-DE	de	de	de	✓	2
2	English (UK)	English (UK)	en-GB	en	en	en	✓	1

At the bottom of the table, there is a "Anzeige # 20" dropdown menu. The footer of the page reads: "Joomla!® ist freie, unter der GNU/GPL-Lizenz, veröffentlichte Software. Version 1.6.3".

Figure 2

Once we have clarified to Joomla which languages are available on our web site, we can then proceed to create different content, as we normally would in any website. The difference now, is that we can define a language parameter in each of our content files, and we will be able to choose between each of the languages that we defined earlier. This informs the CMS which language the content is in, you can see this demonstrated in figure 3 below.

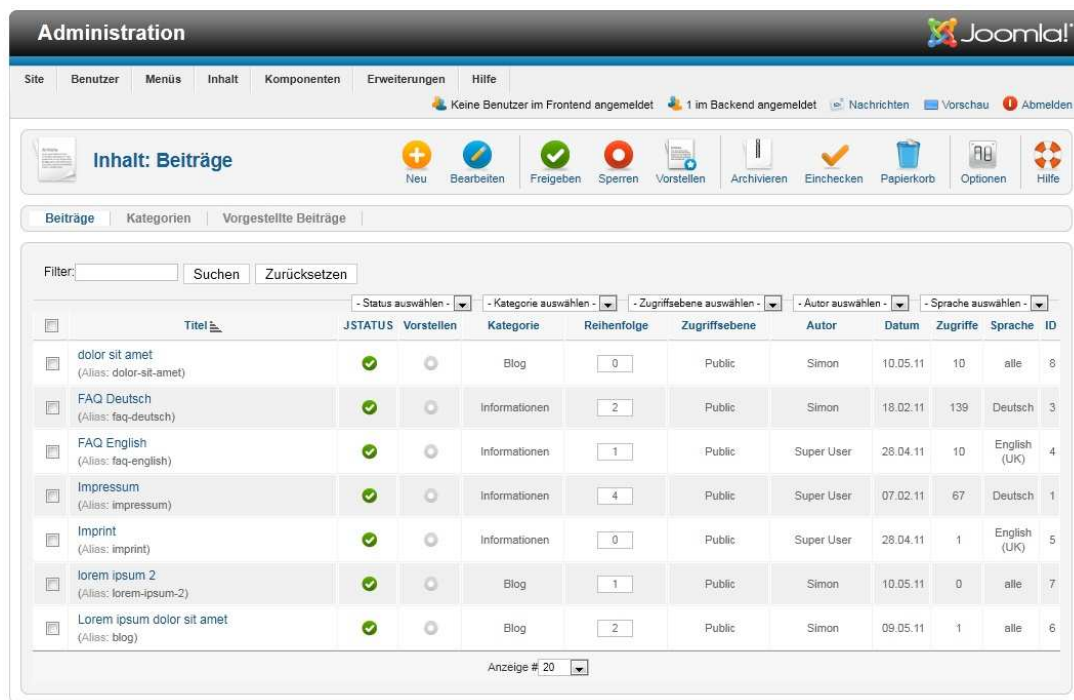


Figure 3

Now that our CMS can choose between which language file is necessary, for the content, we have to assess the manner in which we can translate the interface language. This is done by adding modules to the Joomla system, and it tells the interface what each translated interface item should be. The actual translations are stored within language files that provide the ID of each interface item, and then the translated alternative. This is quite beneficial,

because once the language file has been completed, then it handles everything from then on. It will be noted though, that the initial creation of each language file can be quite time consuming due to the fact that there are quite a large number of interface elements, including months, dates, date formats, buttons, labels, etc. In figure 4 you can see the many different modules that are required in addition to the “User Menu” and the “Benutzermenü” files.

<input type="checkbox"/>	Titel	JSTATUS	Position	Reihenfolge	Art	Seiten	Zugriffsebene	Sprache	ID
<input type="checkbox"/>	Bottom Menu	✓	footer	1	Menü	alle	Public	alle	19
<input type="checkbox"/>	Object Map Display	✓	position-1	1	Object Map Display	Nur ausgewählte	Public	alle	30
<input type="checkbox"/>	Breadcrumbs	✓	position-2	1	Navigationspfad (Breadcrumbs)	alle	Public	alle	17
<input type="checkbox"/>	Sprach Auswahl	✓	position-2	2	Sprachauswahl	alle	Public	alle	21
<input type="checkbox"/>	Was bietet Trees?	✓	position-3	1	Objects Overview	alle	Public	Deutsch	29
<input type="checkbox"/>	Possibilities with Trees?	✓	position-3	1	Objects Overview	alle	Public	English (UK)	31
<input type="checkbox"/>	User Menu	✓	position-4	1	Objects Usermenu	alle	Registered	English (UK)	34
<input type="checkbox"/>	Benutzermenü	✓	position-4	1	Objects Usermenu	alle	Registered	Deutsch	33
<input type="checkbox"/>	Banners	⊘	position-5	1	Banner	alle	Public	alle	18
<input type="checkbox"/>	Administration	✓	position-6	1	Objects Adminmenu	alle	Special	English (UK)	36
<input type="checkbox"/>	Administration	✓	position-6	1	Objects Adminmenu	alle	Special	Deutsch	35
<input type="checkbox"/>	Benutzerlogin	✓	position-7	1	Anmeldung	alle	Public	Deutsch	16
<input type="checkbox"/>	User Login	✓	position-7	1	Anmeldung	alle	Public	English (UK)	28

Figure 4

## 5.4 Choosing the default language

For our specific application, the main language is German, and English would be considered the second language. However there comes the question of which will be the selected default language when the user arrives at the home page. The solution for this in our specific project was again made much easier with the use of the Joomla CMS. Instead of just selecting to use English or German as a mandatory default, the application uses a Joomla plugin that is able to detect the visitor's default browser language setting. We have been able to adapt this to our application for our available languages, and it can detect the default language and adjust the settings accordingly. However this introduces the problem of other languages coming into play, for example, if the visitor's browser was in French, then we have to choose which language would be the default. This is easily accounted for by giving Joomla a default language to resort to in this scenario. For our purposes, the default language is selected as German, as can be viewed in Figure 1.

## 6. Conclusion

I have examined the requirements of multilingual web sites and presented technical and cultural difficulties and some existing best practices to the outlined problems. Also, I presented the practical solutions that were implemented into the web application “Historical Trees and Forests” as a real world example, and compared them to the proposed options.

As our world is becoming ever more internationalized, the need for multilingual web sites and web applications becomes increasingly important. Although it used to be much more difficult to employ features such as this, now days with the use of Content Management Systems such as Joomla, it is becoming much easier to deploy multilingual sites, even for the smallest of web sites. Many web sites, even though they may not currently need support for multiple languages, will begin to build features for multilingual support, because of the effort required to convert a site later, it is much more beneficial to build this support in at the beginning. There are many technologies that are in use today that we would not be able to do this without, and I believe that there will be more introduced every day, and improvements made on our current standards just as frequently.



## Works Cited

- Coombs, Karen. "Drupal Done Right." *Library Journal* 134.19 (2009): 30-32. Academic Search Premier. EBSCO. Web. 17 May 2011.
- Siemens, Phil A. "International & multilingual web sites." *W3C Internationalization*. N.p., 9 July 2003. Web. 17 Apr. 2011. <<http://www.w3.org/International/questions/qa-international-multilingual.en>>.
- "Design Consideration for Multilingual Web Sites." *Information Technology & Libraries* 24.3 (2005): 107-116. Academic Search Premier. EBSCO. Web. 17 May 2011.
- Ishida, Richard. "Handling character encodings in HTML and CSS." *W3C Internationalization*. W3C, 12 Aug. 2010. Web. 19 Apr. 2011. <<http://www.w3.org/International/tutorials/tutorial-char-enc/>>.
- Gupta, Anthea. "The Internet & the English Language." . N.p., 15 Feb. 1997. Web. 19 Apr. 2011. <<http://courses.nus.edu.sg/course/ellibst/poco/paper6.html>>.
- Ruffle, Paul. "Managing Multilingual Web Sites." *Seybold Report: Analyzing Publishing Technologies* 1.13 (2001): 12. Academic Search Premier. EBSCO. Web. 17 May 2011.
- Arno, Christian. "Design Multilingual Website: A Beginner's Guide." . N.p., 1 Jan. 2005. Web. 2 May 2011. <<http://www.hongkiat.com/blog/design-multilingual-website-a-beginners-guide/>>.
- Ishida, Richard. "Text size in translation." *W3C Internationalization*. N.p., 3 Mar. 2007. Web. 17 May 2011. <<http://www.w3.org/International/articles/article-text-size>>.
- Joomla!* N.p., n.d. Web. 7 June 2011. <<http://www.joomla.org/about-joomla.html>>.
- Packt Publishing. N.p., 31 Oct. 2007. Web. 7 June 2011. <<http://www.packtpub.com/article/joomla-wins-best-php-open-source-content-management-system>>.

Empson, Rip. "Joomla Quietly Crosses 23 Million Downloads, Now Powering Over 2,600 Government Sites." *TechCrunch.com*. N.p., 11 June 2011. Web. 13 June 2011. <<http://techcrunch.com/2011/06/11/joomla-quietly-crosses-23-million-downloads-now-powering-over-2600-government-sites/>>.