GLOBALIZING INTERNET WEBSITES

Tomasz Müldner and Zhinan Shen
Jodrey School of Computer Science, Acadia University
Wolfville, NS, Canada B4P 2R6
{Tomasz.Muldner, 072805s}@acadiau.ca

ABSTRACT
The global nature and accessibility of the Internet has generated interest in *internationalization*, i.e. making websites available in various languages and cultures. While the interest in internationalization is growing, up to date, not much research has been done on various aspects of this topic. In this paper, we describe a framework for creating internationalized websites. Our objectives in designing and implementing this framework include developing techniques for efficient representation and storage of source data multimedia and their translations, and support for translation reusability. The framework is implemented as an Internet-based application, using XML and Java.

KEY WORDS
Internet technology, internationalization

1. Introduction

The global nature and accessibility of the Internet has generated interest in *globalization*, i.e. making documents such as websites available in various languages. Globalization consists of two phases: internationalization and localization. *Internationalization* entails generalizing the product so that, without the need for redesign, it can be *localized* to specific languages and cultural conventions. While the interest in globalization is growing, to date, not much research has been done on the various aspects of globalization.

The globalization process includes submitting a request for creating an internationalized website, entering source data to populate this website, specifying target languages (in which the product will be localized), performing translations, providing specifications for the internationalized website, and creating the website. There are a number of important research questions related to this process. First and most important question is how to structure the globalization process to accommodate the process of entering source data, storing them, submitting for translation, etc. Next, the source data may be modified, and if so, will need to be re-translated. This raises the question of how these source data should be stored so that only the modified data are re-translated, as well as how the translations themselves should be stored, so that they can be used to accommodate the process of performing new translations. Another question related to the translation process is whether translations should be performed by machine translation, or manually by translators, possibly with the help of automatic pre-translation. Finally, the question is how to create a single generic internationalized product, from which a website and various documents rendered in different formats can be generated.

Our preliminary research [1] described the Internationalized Faculty Website (IFW), the XML-based system for creating globalized products (both websites and documents) storing faculty CV (Curriculum Vitae), and showed how some internationalization activities may be structured. In this paper, we describe extensions of that research; the design and the implementation of the WGF system that can be used to globalize websites. Based on a data definition, the system generates a form that is used to enter only valid data, using a specific source language. Once the process of entering data has been completed, WGF takes care of submitting data to the translators and verifiers, and when translation is done, it guides the user in the process of creating the required website.

This paper is organized as follows. In Section 2, we provide the motivation and review the related work. Section 3 describes our system, and Section 4 its implementation. Finally, Section 5 provides conclusions and describes our future work.

2. Motivation and Literature Review

Growing interest in globalization and localization [2] has prompted the development of various standards specifically designed for globalization tasks, such as ISO language codes, the Unicode standard and related encoding methods [3, 4]. Several companies have introduced internationalized software; for example, Webmail [5] is a popular browser-based email client, which can be localized to over 20 languages. Some companies have tried to extend existing Content Management Systems (CMS), which typically focus on source content management, including content creation and publishing, and produce Globalization Management Systems (GMS), which focus on the translation and localization cycles. To date, however, there are very few research projects on globalization, and those that do exist are limited to internationalized websites that use dynamic web pages. For example, [6] suggests rules for websites internationalization, and [7] proposes a framework for the internationalization of *data-intensive* web applications. Finally, [8] describes the application of XSLT to localize XML documents; however, this work does not attempt to develop a uniform approach to globalization. As of now, there is virtually no research on GMS.

At the same time, the globalization industry has done some preliminary work on using automated translation methods using Computer Supported Translation (CAT). Using CAT, source documents are broken into *segments*. Translation Memory (TM) systems look up these source and target (translated) segments in persistent storage to help in the translation process. Some companies offering TM systems are described in [9, 10]. To make it possible to share existing translations of segments, TM software may use the **T**ranslation **M**emory e**X**change (**TMX**) format [11], which is a XML document type for storing collections of segments in multiple languages. Unfortunately, as a result of segmentation, the context for the source text is lost, and its translation may be incorrect. While TM systems merely help in the re-use of existing translations by finding the source segments that have already been translated, Machine Translation (MT) systems [12] use target language linguistic rules to help in automatic translation. The XML Localization Interchange File Format, XLIFF standard [13] has been created to store meta-data pertaining to the translation process, in order to support exchanging data between TM and MT. CAT is still in its early stage of development, and as of now there are a few applications that use it.

It is unclear which technologies are the best for internationalization. Two major candidates are XML [14] and Java [15]. However, existing proposals of how these technologies can be used for internationalization are limited to specific, low-level issues, such as element encoding or usage of resource bundles. Our initial research [16, 17, 1] described various benefits of using various XML technologies for globalization, including XML Schema, and XSLT [18]. For instance, one of our initial research efforts involved the design and implementation of the Internationalized Faculty Website (IFW) system [1], which can be used to create internationalized products containing the Curriculum Vitae (CV) for a faculty member. We also designed and implemented a system for the Internationalization of Data using XML, IDUX [16], in which we extended the original IFW system by allowing the user to use a variety of data definitions (and not just CV data). We have implemented a preliminary version of this system using a relational database to store XML data, and we have designed a common API that allows the framework to communicate with different types of databases, including pure relational databases, XML-enabled databases and native XML databases.

3. Website Globalization Framework

This paper describes an XML-based Website Globalization Framework (WGF) for building internationalized websites. Below, we provide a complete list of high-level requirements for WGF:

- *Multilingualism:* support for issues such as right-to-left and vertical text rendering.
- *Separation of Concerns (SOC):* separation of roles that require different types of expertise.
- *Distribution:* implementation of the framework allowing for cross-platform development.
- *Platform Independence:* accessibility by users of heterogeneous systems.
- *Customizability:* of both the framework and the final website.
- *Scalability:* i.e., addition of new components to the framework does not hinder its performance.
- *Persistence and Reusability:* of both source data and translations.
- *Low Cohesion:* loosely connected components, allowing replacement of existing components and addition of new components without affecting the operation of existing components
- *Portability:* data can be moved between different kinds of persistent storage and network nodes.
- *Efficiency:* the framework will be efficient in terms of time and space.

3.1 Introduction

Following the “Separation Of Concerns” principle, WGF provides separate roles that require different type of technical expertise. There are five kinds of user **roles**:

- creators (enter source data, select languages, etc.)
- administrators (maintain accounts, etc.)
- translators (translate documents submitted by administrators)
- verifiers (verify translations submitted by administrators)
- end-users, who access the internationalized website

Fig. 1 shows some actions preformed by users in the first four roles. Creator is not involved in the translation process; she or he creates a translation task, submits it to the administrator and when the translation is completed, it is made available by the administrator to this creator. Here, by *translator* we mean a human being or a machine translation, MT program. Indeed, WGF is designed to make it possible to transparently switch between human translators and MT software.

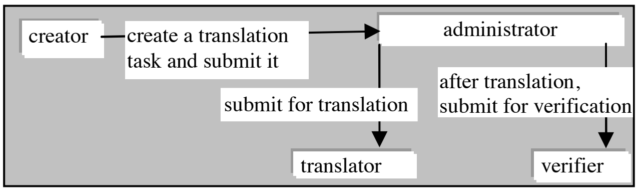

creator
create a translation task and submit it
administrator
submit for translation
after translation, submit for verification
translator
verifier

Fig. 1. Top-level view of a submission of the translation task

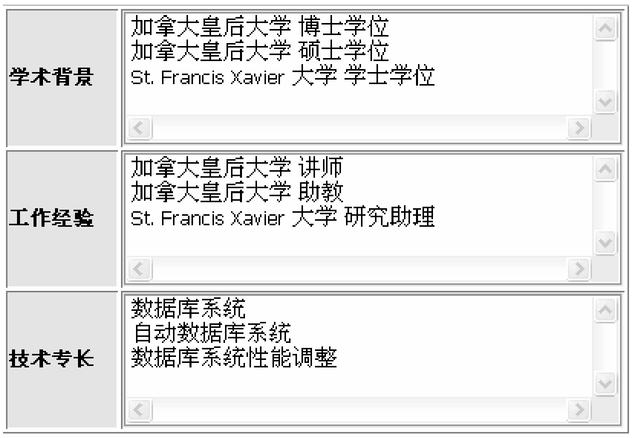
Each translator has a **profile** that lists pairs of languages, and a specialization; the source language is a language to translate *from*, the target language is the language to translate *to*, and a specialization specifies an area such as Computer Science. For example, a translator's profile may include (English, French, Biology) and (English, German, Computer-Science), while another translator's profile may include (French, Polish, Mathematics) and (Polish, French, Mathematics). We do not assume that the ability of translating from one language to another is reflexive. Note that all data, their translations and their status are persistently stored by WGF, and the translators can re-use previously accepted translations; see Section 3.5. The preferred way to translate a text is to use a direct translation; however, if there is no translator who can perform this task, WGF may choose to perform an *indirect translation*; first from the source language into an intermediate language, and then from this intermediate language into the target language. Based on availability of translators, WGF maintains for each language, the list **of translatable languages**.

The reason that WGF provides *a verifier role* is that the translator may or may not see the context for the complete translation. Also, as we mentioned above, a translator may be a MT program, rather than a human being, and results of these program should always be checked. A verifier has a profile similar to that the translator. The verifier can accept a translation (possibly with minor corrections), or reject it.

While the administrator is responsible for assigning translation requests to specific translators and verifiers, various administrative tasks may be *automated*. For example, we have designed and implemented an algorithm, which selects translators based on their profile, availability, load, and acceptance of indirect translation. In the following section, we describe in details the process of creating and submitting data for translation.

3.2 Actions of the Creator

To standardize the globalization creation process, each source document must adhere to a so-called **data definition**. For example, a data definition may specify a CV (resume), or a collection of DVD movies. A data definition is presented to the user in a form of GUI, called a **data definition form**, which allows the user to enter only source data valid for this definition. Various fields in the GUI have *labels* in a language called **an interface language**. Each data definition form comes with one or more interface languages to facilitate the process entering source data in a preferred language. A very simple example of a data definition form that uses Chinese as an interface language and a source language is provided in Fig. 2. Here,"学术背景","工作经历"and"技术专长"are **labels** of the definition form. Of course, a preferred situation is the one in which the interface language is the same as the language used for the source data; however, if this interface language is not available, it may be enough for the creator to use a language, for which they can *understand* labels. We will elaborate on this issue below.



学术背景	加拿大皇后大学 博士学位 加拿大皇后大学 硕士学位 St. Francis Xavier 大学 学士学位
工作经验	加拿大皇后大学 讲师 加拿大皇后大学 助教 St. Francis Xavier 大学 研究助理
技术专长	数据库系统 自动数据库系统 数据库系统性能调整

Fig. 2 Simple data definition form

To create a new source document, users first select an existing data definition. WGF maintains two kinds of general repositories of various resources; in particular, *data definition resources*. A **global repository** is maintained by the administrator and accessible to any creator. A **local repository** is maintained and accessible to a specific creator. Therefore, the administrator is responsible for the quality of a global repository, and the creator is responsible for their local repository. There is also a **public repository**, which is used by creators to *publish* local resources. This public repository is accessible to all users; in particular, administrators may select its resources and, possibly with modifications, move to the global repository.

The creator knows at least one language, such as their native language, but the functionality of WGF facilitates the process of creating internationalized websites by creators who know more than one language. Specifically, WGF maintains the set of **source languages**, specified by the creator. It is the creator's responsibility to enter data, which are grammatically and semantically correct according to the selected *source language,* because these data are not sent for verification. Data entered in one of the source languages may be translated into one or more **target languages**.

Example.
The creator knows English and French (which are source languages for this example), enters data in English and in French, and specifies English, French, German and Chinese as target languages. Note that WGF will allow the creator to select German and Chinese as target languages, only if it is possible to translate source data from English or French into German and Chinese. The exact choice of languages in the

translation requests depends on the availability of translators and their current load; for example there may be two translation requests: from English to German and from French into Chinese. The resulting website will be internationalized in four languages (we provide more details of how this website can be customized in Section 3.4).

Note that if several source languages are used by the creator, then it is more likely that the translations may be provided. In the above example, if the creator knew Polish, and WGF did not provide translations from English or French into German, but it did provide translations from Polish into German, then WGF would ask the creator if they wish to provide source data in Polish, in order to satisfy the translation requests.

The creator may also request a translation of *labels* used in a data definition form. Similarly to the request of translating source data, the translation may be provided by the creator, using one of the source language, and then stored in a *local* interface language repository. The creator may also request a translation of labels in one or more target languages, and in this case data will be sent for translation and verification, and when completed, stored in a *global* interface language repository.

Once the translation request is submitted to the administrator, the creator waits for the completed and verified translation. This verified translation is called a **generic website**, because it can be used to generate various kinds of websites. We describe this process in Section 3.4.

3.3 Maintenance and Customization of Creators Resources

The creator has access to the various resources, including source data, which have been previously translated. If these data have to be modified, for example they represent the list of publications and a new publication is added, then only a new or modified data will be sent for translation and verification. Data that have not been changed will not be re-translated.

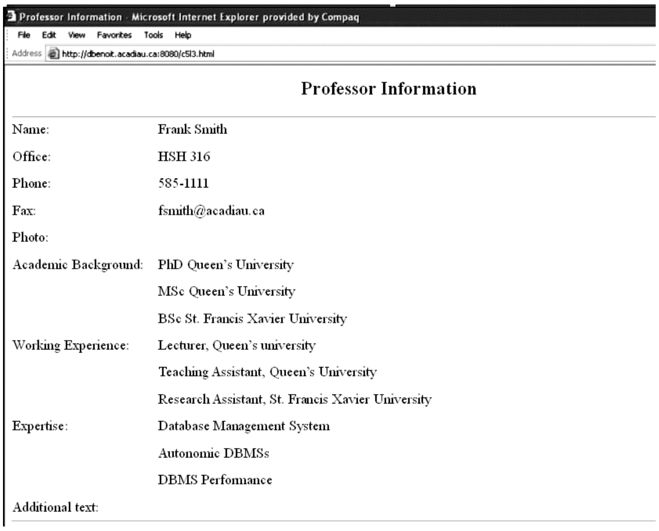
Another important resource accessible to the creator is a set of data definitions and associated forms and interface languages. We are working on taxonomies for data definitions for various domains, for example data definitions for various types of resumes. However, for a specific need of a creator, the required data definition may not be available, although there are "similar" data definitions. Therefore, WGF allows the creator to modify the existing data definition and store in a local repository. Examples of modifications are adding or removing a specific field, such as the list of supervised Master students. These modifications can take place at two levels. First, at a *higher level*, a data definition may be modified, effectively creating a new data definition. Second, at a *lower level*, a data definition form may be modified, to accommodate a single-use creation of a website.

3.4 Final Websites: Customization of Generic Websites

A generic website is used to generate one or more websites, for example by providing a **format** for rendering. Available formats such as PDF or XHTML are stored in a local or a global repository. In addition, the creator may generate customized versions of the generic website, for example by selecting *some* of the data to be excluded. Let's now describe the details of the customization process, leading to a **final website** (here, called a website).

A website has a **cover page**, which is the initial webpage shown in a selected **default** language. (If no default language is selected, then the locale is used to determine the default language.) The cover webpage provides a menu to move to webpages displayed in other languages. The creator may filter out data, i.e. exclude some data from specific webpages. For example, the English webpage may show all the data, while other webpages may exclude some data. Finally, different formats may be assigned to these webpages. For example, the website may be initially displayed using Polish, allowing the user to switch to webpages using these languages: Chinese, English, German and French. Webpages for the first three languages use XHTML, and the French webpage uses PDF. Figures 3, 4 and 5 show the website consisting of three webpages; respectively in English, Chinese and Polish.

The creator can assign different *widgets* to individual elements of a generic website, and modify the default placement of these widgets on the website. For example, the creator can choose a table for some elements and a list of other elements. The possibility of modifying the default placement of widgets is particularly important, because various languages use different alphabets and scripts, spacing rules, directions, etc. Therefore, placement of widgets has to consider issues such as word length, word positioning, font differences and other such items.

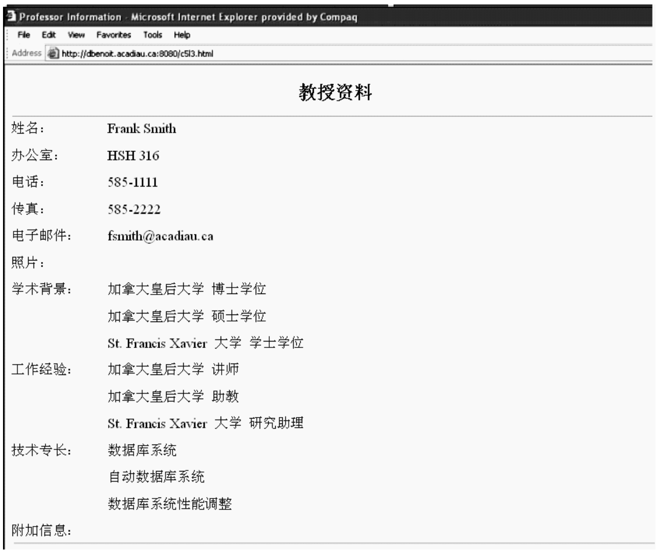

Professor Information - Microsoft Internet Explorer provided by Compaq
File
Edit
View
Favorites
Tools
Help
Address

http://dbenoit.acadiau.ca:8080/c5l3.html

Professor Information

Name:	Frank Smith
Office:	HSH 316
Phone:	585-1111
Fax:	fsmith@acadiau.ca
Photo:	
Academic Background:	PhD Queen's University MSc Queen's University BSc St. Francis Xavier University
Working Experience:	Lecturer, Queen's university Teaching Assistant, Queen's University Research Assistant, St. Francis Xavier University
Expertise:	Database Management System Autonomic DBMSs DBMS Perfomance
Additional text:	

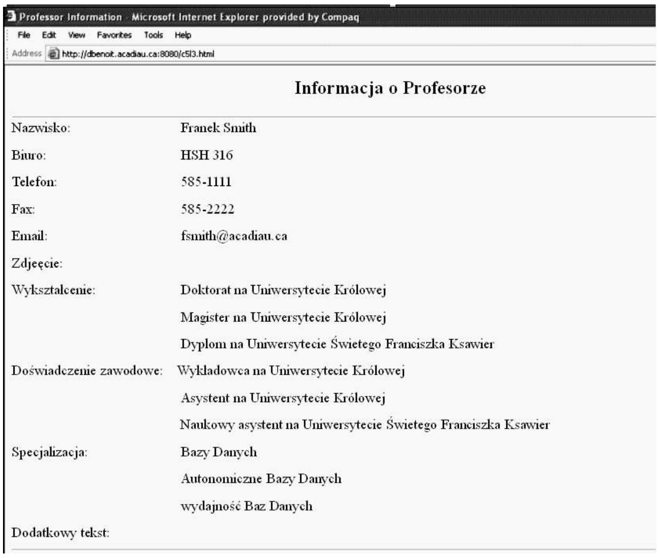
Fig. 3. English webpage


Professor Information - Microsoft Internet Explorer provided by Compaq
File Edit View Favorites Tools Help
Address http://dbenoit.acadiau.ca:8080/c5l3.html

教授资料

姓名：	Frank Smith
办公室：	HSH 316
电话：	585-1111
传真：	585-2222
电子邮件：	fsmith@acadiau.ca
照片：	
学术背景：	加拿大皇后大学 博士学位
	加拿大皇后大学 硕士学位
	St. Francis Xavier 大学 学士学位
工作经验：	加拿大皇后大学 讲师
	加拿大皇后大学 助教
	St. Francis Xavier 大学 研究助理
技术专长：	数据库系统
	自动数据库系统
	数据库系统性能调整
附加信息：	

Fig. 4. Chinese webpage


Professor Information - Microsoft Internet Explorer provided by Compaq
File Edit View Favorites Tools Help
Address http://dbenoit.acadiau.ca:8080/c5l3.html

Informacja o Profesorze

Nazwisko:	Franek Smith
Biuro:	HSH 316
Telefon:	585-1111
Fax:	585-2222
Email:	fsmith@acadiau.ca
Zdjeęcie:	
Wykształcenie:	Doktorat na Uniwersytecie Królowej
	Magister na Uniwersytecie Królowej
	Dyplom na Uniwersytecie Świetego Franciszka Ksawier
Doświadczenie zawodowe:	Wykładowca na Uniwersytecie Królowej
	Asystent na Uniwersytecie Królowej
	Naukowy asystent na Uniwersytecie Świetego Franciszka Ksawier
Specjalizacja:	Bazy Danych
	Autonomiczne Bazy Danych
	wydajność Baz Danych
Dodatkowy tekst:	

Fig. 5. Polish webpage

The final website may include not only the text, but also any multimedia resources made available by the creator. The **multimedia repository** includes multimedia files, such as photos and videos, and their descriptions in various languages; for example "English video about DVD formats". The final website may appear in one of two available kinds:

- transient - all translations are stored in the database, and webpages accessible from the default page are dynamic.
- persistent - all webpages are static, generated by retrieving data from the database, and applying all transformations.

From the perspective of the client, who is accessing the website, there is no difference in what kind of final product is used. However, choosing a specific kind influences requirements on the server side, and efficiency of the website. The generation process is performed by the administrator. This process is time consuming and therefore should be performed only if the source data are not to be frequently changed. On the other hand, a persistent product, consisting entirely of static webpages, reduces the overhead caused by creating dynamic webpages, and does not impose any additional requirements on the website server. Therefore, the persistent product can be copied from the server to any site with the standard web server. A transient product requires a website with the server that can handle servlets, access a database, etc. (see Section 4).

3.5 Actions of the Translator

Because of space limitations, in this paper we describe only briefly translator's actions. The translator can reuse existing translations or provide new translations. If these new translations are approved by the verifier, then they are incorporated into the repository of translations. The standard implementation of the Translation Memory, TM uses a list of existing translations of segments. In our approach, we take advantage of a structure on TM to speed up the search for existing translations. In the future, we will use taxonomy of data definitions to allow the search to start within TM in the place where there is the highest likelihood of finding the match.

4. Implementation

We use XML [19] because of its support for Unicode, separation of concerns (XML describes content rather than formatting), and ease of conversion between various formats. XML Schemas are used to implement data definitions. As a programming language for the implementation, we chose Java [20], because of its support for internationalization, and various built-in tools, such as servlets JSP [21], and JAXB [22]. WGF is a distributed system, consisting of a central server (WGF server) and users accessing the server through the Internet through an arbitrary browser; see Fig. 6.

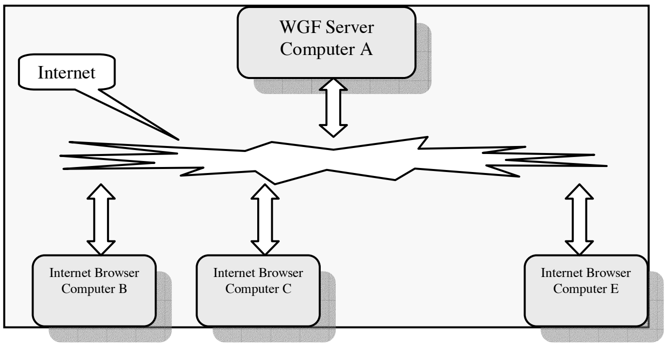


Fig. 6. Accessing a WGF server

Administrators of the WGF server create, modify and delete accounts for all users and maintain repositories of available translators and verifiers. Since one of our objectives is to support a cross-platform development, we use a multi-tier design for WGF; see Fig. 7, with a central WGF server, responsible for tasks such as communication with the database. Native XML databases are used for storing XML documents, and XQuery is used for the implementation of translation reusability. To support translation reusability, globalization systems use Translation Memories (TM). Current implementations of a TM require traversing the entire TM to search for existing translations, which is often inefficient. Since the current state of automated translation systems requires human intervention, initially, we do not intend to use automated translation. However, to make it possible to seamlessly incorporate automated translation systems into the framework architecture, we use XLIFF [13], which is used by the automated translation tools. In the future, we also intend to investigate the suitability of automatic pre-translations.

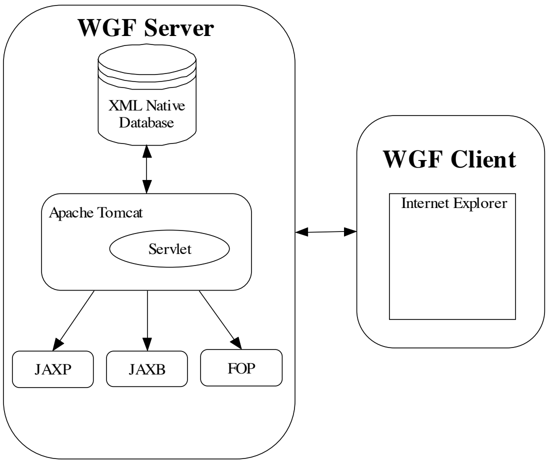


Fig. 7. A general architecture of WGF

5. Conclusion and Future Work

In this paper, we described the design and the implementation of the WGF system that can be used to globalize websites. Based on a data definition, the system generates a form that is used to enter only valid data, using a specific source language. Once the process of entering data has been completed, WGF takes care of submitting data to the translators and verifiers, and after the verification is done, it guides the user in the process of creating the required website.

Our future work includes creating another version of our system, in which various users will be able to work *offline*, using specialized client applications, rather than browsers. The process of entering data and their translations in various languages will be further accommodated by using specialized editors. We will develop standard taxonomies for data definitions for various domains, and work on tools to facilitate the creation of these taxonomies. Translation memory plays an important role in translation reusability and automatic pre-translation. We will develop standards for translation memory and, based on our experience, populate them with useful examples. The later phase of our research project will tackle the issue of globalization of legacy systems (websites and documents), which have not been created using XML. For example, Microsoft Word documents may be converted into XML using Word 2003. We will investigate techniques to convert various non-XML documents into XML and then apply the globalization process defined in our paper.

References

1. T. Müldner, F. Wang and D. Benoit, *My webpage can speak many languages*. EDMEDIA'04; Lugano, Switzerland. AACE Proceedings, pp. 2040-2046

2. Localization Industry Standards Association, (1999-2004), *LISA Standards Directory Map*, http://www.lisa.org/

3. Unicode Organization, (1991-2004), *Glossary of Unicode Terms*, http://www.unicode.org

4. W3C 2002, *URI encoding programs*, http://w3.org/International/O-URL-code.html

5. Webmail (2003) webmail http://www.webmail.co.za

6. N. Aykin (1999) *Internationalization and Localization of the Web Sites*. HCI (1), pp. 1218-1222

7. A. Belussi, R. Posenato, (2004) *Internationalizing Data-Intensive Web Applications*, Department Computer Science, Rap. di ricerca RR 16/2004, University of Verona.

8. Y. Yu, J. Lu, J. Xue, Y. Zhang, and W. Sun (2003) *Localizing XML Documents through XSLT*. Applied Informatics, pp. 1059-1064

9. Trados (2003) *Translation Software* http://www.trados.com/

10. Transit (2003) *Translation Software* http://www.starag.ch /eng/software/sprachtech/transit.html

11. TMX (2004) Translation Memory eXchange (TMX) http://www.lisa.org/tmx/

12. G. Dennett, (1995). *Translation Memory: Concept, products, impact and prospects*. http://www.star-uk.co.uk/About_us/People/Gerald_Dennett/msc.pdf

13. XLIFF (2003). *XLIFF 1 Specification*. http://www.oasis-open.org/committees/xliff/documents/xliff-specification.htm

14. Y. Savourel (2001), *XML Internationalization and Localization*, Sams Publishing

15. A. Deitsch and D. Czarnecki, (2001), *JAVA Internationalization*, pp 49-61, O'Reilly & Associates

16. D. Benoit and T. Müldner, *IDUX: Internationalization of Data Using XML*, IADIS WWW/Internet 2004 Conference pp. 469-476

17. T. Müldner and D, Benoit, *Generic Approach to Internationalization of Websites*, IASTED International Conference Software Engineering and Applications SEA 2004 pp. 465-470

18. World Wide Web Consortium (W3C), (1994-2004), *XSL Transformations (XSLT) Version 1.0*, http://www.w3.org/TR/xslt

19. W3C(2004) Extensible Markup Language (XML) http://www.w3.org/XML/

20. Sun Microsystems(2005) Java Technology http://java.sun.com/

21. Sun Microsystems(2005) JavaServer Pages Technology http://java.sun.com/products/jsp/

22. Sun Microsystems(2005) Java Architecture for XML Binding (JAXB) http://java.sun.com/xml/jaxb/index.jsp