Trusted Translation Services

Yacine Atif, Mohamed Adel Serhani, Piers Campbell, and Sujith Samuel Mathew

College of Information Technology
UAE University
{Yacine.Atif,serhanim,P.Campbell,s.mathew}@uaeu.ac.ae

Abstract. Administering multilingual Web sites and applications reliably, involves interconnected and multipart tasks, where trust in the involved parties and content translation sources is paramount. Published Web sites may reflect content from databases, content management systems and other repositories to manage related Web content. But a Web site mirrored wholly or selectively onto a target language version requires streamlined trusted processes. Traditionally, files are translated and transferred via FTP, e-mail, or other communication means. Similarly, translation instructions are communicated between involved parties through verbal instruction, e-mail, and instruction files lead to a variety of inconsistencies and lack of trust in the translation process. This paper proposes a Web service approach to streamline the translation processes and an integration of trust properties in the proposed translation Web services. Web Services have been instrumental in handling problems inherent to systems integration, allowing web-based systems to converse and communicate data automatically. The OASIS Translation Web Services Technical Committee has released a standard way for Web Services to serve the translation and localization business. This article proposes a framework to centralize translation services at a reputable source providing a workflow and a mechanism to quantify service trust. An implementation of the framework is also described in the context of a localization case study.

1 Introduction

Any multilingual site owners would realize the soaring cost in translating content, as well as the opportunity cost of having it localized and culturally adjusted. The substantial effort involved in integrating a translated content to produce target language pages entails a certain challenge in making sure the translated sites meet localization standards, particularly when the source language sites are dynamic. Furthermore, given the regional operations' requirements on regional sites to reflect in a timely manner offshore content, one may need to define a model of trust whereby these operators can accomplish some of their processes themselves in updating their own content. The need for language translation in the appearance and operations of web contents and services is as appealing as challenging to modern industries. Yet, those aspirations have to respond also to a certain level of trust as they involve Web exposure of businesses and client-facing interfaces. A framework for a reliable and

E. Bertino and J.B.D. Joshi (Eds.): CollaborateCom 2008, LNICST 10, pp. 778–791, 2009.

automatic translation processes using current infrastructure, to respond to today and future trusted localization requirements of global web contents, forms the scope of this paper.

Our particular collaborative internet-based translation framework, involves dozens of expert researchers to translate prominent web contents from source languages into a targeted language. It allows multilingual content site owners to translate source-language articles to a target-language destination which is relevant to localized content for prominent regional market. The Translation Management System (TMS) isolates the effected articles and invokes a translation Web service [1] which route the translation request across trusted translation parties. Thus our framework is based on a Web services translation engine which is compliant with OASIS Translation [2] standards. The need to design standard Web Services for translation was highlighted at several instances in the context of web content with global reach intent. For example, the Olympics Web site [4]. Yet, Web content owners are reluctant to solicit translation services due to the long term partnership requirements to translate dynamic content reliably. TMS provides an on-line translation factory based on standard document tagging structure and involving field expert translators.

Earlier, Oasis standardization body developed XLIFF (XML Localization Interchange File Format), an "XML-based format that enables translators to concentrate on the text to be translated" [6]. XLIFF supports a full localization process by providing tags and attributes for review comments and tracking translation status. XLIFF aggregation recognizes several source document formats, and will translate them to XLIFF counterpart destination. Any translation service or application that understands XLIFF can open and change the communicated content. XLIFF grew as a result of significant work from the industry to develop localization within OASIS. Created at the beginning of 2003, the OASIS Technical Committee includes representatives from major IT industries including Oracle, Microsoft and IBM. This standardization embracement facilitates the propagation of translation engines such as the one proposed in this paper.

While web services technology's short span of life has seen extensive applications in business information, with rich content such as airline reservation, its exploitation in globalizing reliably web contents has not been fully investigated. The approach adopted in this paper aims at reducing cost of translation and localization process when globalizing web contents. The proposed framework mediates the interaction instances between a translation solicitor and provider using standard translation related web services. These interactions may involve contract agreement on translation terms and the actual translation job itself, in trust-building environment. The proposed architecture empowers corporate with worldwide operations to meet effectively a global audience. TMS is a business partner providing this far-reaching audience to corporate and businesses.

Traditional localization processes incur lack of trust because notorious issues such as redundancy, communication overheads, high translation costs, errors and delays have created reluctance to translation and Web content globalization. To alleviate these shortcomings, we propose TMS, a Web portal giving clients quick access to a

wide range of trusted translation services. TMS offers Web Service interfaces with simple and direct programmatic access to existing content repositories (such as CMS). This seamless integration increases further level of trust in the translation process and reliability of the translated content. TMS Web Services use XLIFF to specify the organization of translation content, and Translation Web Services. This approach streamlines the communication between translation clients and service providers. Using these services, web source content is translated and republished directly into the production website, while translation status can be tracked during translation until completion is unveiled. All data retrievals are handled without tedious manual intervention. In addition, once extracted from a CMS, TMS employs a variety of QoS processes to ensure the reliability, precision, and speed of the translation process. For example, translated resources may be referred to a translation review service prior to commit the delivered translations. This added-value service further strengthens the level of trust in the translation process.

The remaining sections of this paper are organized as follows. First, we discuss some earlier related work to justify the proposed web translation model in this paper. Then, we reveal the translation workflow and related processes followed by the communication architecture of the proposed TMS framework. We then present a case study of an ongoing TMS-based implementation named Murshid, which aims at validating the proposed model in producing Arabic web content out of existing prominent web sources. Finally, we conclude the paper with a summary of results and suggestions for future work.

2 Background and Related Work

There are a number of different approaches to translating a website, each with its own advantages and disadvantages. To successfully cross the language translation divide, many suggestions have been reported [7] for both writing clear, translatable content as well as choosing a suitable translation service. In many cases, it is not simply enough to translate words into another language. Languages with scripting systems that read from right to left, for example, should have layouts and menus adjusted accordingly when translated. Furthermore, cultural dimensions may be considered in replicating interfaces for countries of different cultures as look and feel may differ from one country to the other [8]. Actually, designing interfaces for international use has been extensively addressed in the past [9, 10] where usability principles and cultural issues are considered in translating an interface.

There are a number of ways to go about providing translations of websites. One such way is to perform translation with an automated translation program, though quality of these results can vary greatly. An example of such an automatic system is AltaVista's Babel Fish Translation [11] in which users may receive translations for words, sentences, or paragraphs by entering these directly, or translating a full web page by providing a URL to a page. However, automatic translation is still in its infancy, especially when multimedia contents is involved [13].

Another approach to web content translation is for the designer of a website to perform translation of content and interfaces by hand, which can be a cumbersome and time-consuming. The problems which can arise in this approach are discussed in 'Internationalizing Online Information' [12]. Furthermore, developers tend to ignore internationalization issues when designing a website and must reconsider these at the time of deployment.

A third option for Web content translation is to appeal to a site's user-base and allow this select community of volunteers or professional translators to translate a website. This approach has been successful in translating online newsletters and entire websites. This approach depends on an online community of users to volunteer their time to do the translation. One example of community-based translation can be found in TidBITS [14], and SOL [15]. Teams of translators and editors work on the translation process. One team member may acts as coordinator to assign articles to the other four, and assemble the finished work. This administrative hierarchy has been found to work well, allowing an extra level of style and quality checking. The coordinator is in charge of maintaining the quality of the translation. Similar initiatives prove the effectiveness of the community-based approach. OmniWeb [16], a browser created for Mac OS X which includes support for more languages than any other web browser. Another example is the Translation Project [17] which aims to translate free software packages into various languages. "Google in your language" [18] is another example of community-based translation. Users of Google register to translate into one of one hundred and forty-eight languages, and are then presented with items (strings of words) which appear on Google's pages that could need translation.

Community-based translation presented above appears to be most prominent and advocated by several initiatives as it uses a wide source of human translation expertise. However, these are ad-hoc approaches suitable for specific Website or tool. The processes are slow due to the absence of a common translation specification standard and a system that streamlines translation workflows. In this paper, we introduce a general approach to community-based translation where the Web acts as a natural middleware linking translation client and service providers. The proposed Translation Management System or TMS has a suite of remote translation-authoring, content reviewing and approving in integrated workflows. This framework allows content editors from different locations to request for language translations and for translators to perform translation online. The Translation Management System lowers the overall cost, reduces the administrative overhead and streamlines the entire operations. Furthermore, the adoption of open standards leads to the construction of powerful translation services. The proposed model represents a translation aid platform that content providers can use, with the prospects to offer translation web services to their clients, without the need to transfer valuable multilingual content. Finally, the implementation approach enables the integration of such service into larger business information applications with superior multilingual processing capabilities.

3 Translation Workflows and Web Services

The proposed service architecture is an attempt to increase the level of performance of producing correct translations. It joins a persistent translation engine to a dynamic

source for translations on the fly. For every thematic domain, participants cooperate around the TMS-provided services to accomplish a domain-related translation. TMS participants include six categories of roles: guests, translators, editors, translation project managers (or managing editor), reviewer and administrator, for every translation service project. Relevant translation-related web services are triggered for each participant category. Members of each category are able to access appropriate TMS services available to that category and all the preceding categories. Hence the scope of services increase hierarchically.

XLIFF is used as communication medium between clients and TMS, whereas Translation Web Services define the nature of translation related services for a given translation project. Our standards-supported approach allows source content of any media type to be posted directly into TMS's translation production process from the client CMS. Translations can also be tracked from the CMS engine to facilitate automatic integration into the client's portal when translation completes, without additional manual intervention. Further QoS attributes can include a variety of translation productivity tools such as accuracy, and timeliness of the translation. Figure 1 includes typical TMS and client modules of the framework architecture. This framework is applied to our TMS design and instantiated to our translation case study.

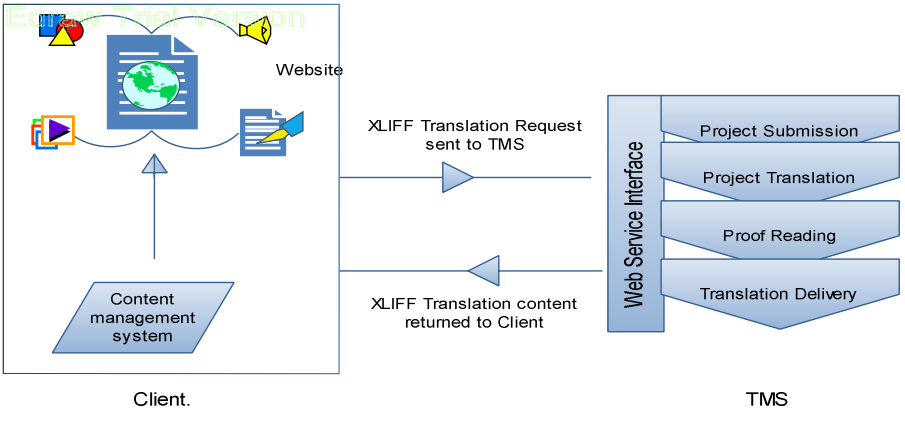

Fig. 1. TMS Architecture

This is the model which we have used for translating prominent web source to Arabic language as a case study of the proposed framework. Typical websites use a content management system (CMS) to store media and articles displayed on the website. Translation is processed through the TMS, an online translation service provider. Initially, content is taken from the source web site CMS and integrated into an XLIFF document (with relevant translation-related tags). Following a Service-Oriented Architecture, SOAP protocol wraps each XLIFF document and sends it to TMS via Web Services invocation. At the TMS side, Web service interfaces represented by corresponding WSDL define how the available services are invoked as

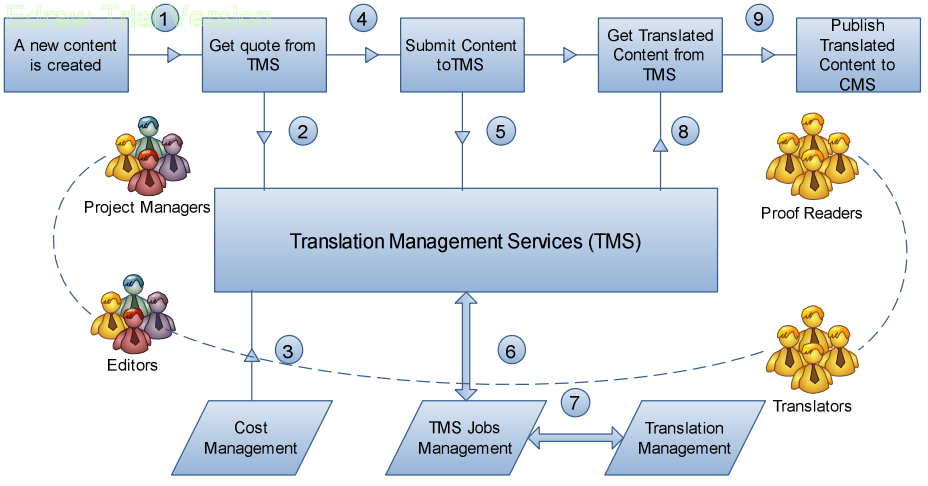

Fig. 2. Translation Workflows

well as their properties. A request carrying the XLIFF-wrapped document reach the TMS, where the translation process is managed and a translated version of the document is returned to the web site CMS via web services. Typical workflows are described in Figure 2. The execution of translation projects involves several document transfers between the project members whose roles are described in the subsequent section. These transfers are vulnerable to errors because they are processed manually. The proposed translation workflows are more effective than traditional email communications when the translation content is substantial. It controls the transfer of translation requests and returning back translated documents, while permitting members involved in the translation development to follow-up the state of every resource as well as identifying errors and delays punctually. Another advantage of this framework is its provision of accounting applications with respect to the size of a translation job (in terms of word count) as show in Figure 2 (Steps 2 and 3). The translation workflow is integrated with translation-related databases (Steps 6 and 7), and client CMS (Steps 8 and 9).

Web services offer several advantages in this translation processes streamlining framework [19]:

- Universal service repository available to anyone
- Selective access to any translation process
- Client can communicate with any translation vendor who support the service (such as the proposed TMS)

The details of the advertised Web Services used in TMS have some of the basic functionality that one might expect from a translation vendor service. Table 1 below illustrates the standard Web services used in TMS, and which specifications are

reported in [3] and [5]. Our implementations of these specifications are carried out along a workflow to be elaborated later in this article. As described in Table 1, `submitJob` initiates the translation process after cost negotiations triggered by the sequence of methods `requestQuote` and `acceptQuote`. Once status of a job has changed to completed, the method `retrieveJob` is called to integrate the translated contents into the client's CMS. Besides these basic translation web services, other mechanisms provided further service quality such as `reviewJob` which proofreads the quality of a translated content and returns an enhanced version of the translated resource.

Table 1. Translation Web services

Web Service	Specification
requestQuote	This request initiates the translation process by returning an estimated cost to the translation source. Rates may vary according to the translation volume and on volume and media type rates.
retrieveQuote	This call facilitates the retrieval of a quote.
acceptQuote	Using the proposed Job reference assigned when a request for quote has been issued, a translation request may be accepted, at which point translation process starts.
submitJob	Eventually combines requestQuote & acceptQuote, when a relationship between client and service provider has pre-established, or translation fees have been prepaid. Alternatively, this web service may be invoked when translation is deployed internally and hence bypassing quotes negotiation phase. This is the case for our case study implementation (i.e. Murshid) of the TMS discussed in this paper.
retrieveJob	When a translation is complete, this service is called to retrieve the translated resource.

This process maximizes automation the translation process and the communication between involved stakeholders, and therefore shortens the turnaround times. Access to TMS via web services is also made available to service provider personnel. For example, TMS typically uses a range of translators, who may be internal or remote. Translators use a Web service to retrieve pending translation documents. Using this service translation requests could be checked out, retrieved and later on submitted back, from within a translation service provider context. TMS actually involved a number of personnel to fulfill the requirement of a translation (as shown in Figure 2). These include:

a) *Guests*: Guests are allowed to browse parts of TMS translation database. Since the translated documents are meant to be published anyway, guests may search for text phrases, Boolean combinations of phrases, particular keywords, and various field-specific searches to index translated articles or resources. These resources are returned

with their target URI links, to invite guests to visit the target websites for which the articles were translated. TMS maintains two translation databases: the test and production database. The test database provides translators with an experimental environment of translation jobs. Committed translation works are sent to the production database. While the test database could be considered as a temporary working memory, which will be erased upon committing translation jobs, the production database is a sustained memory environment of all delivered translation by TMS.

b) *Project Managers*: Maintain interaction with a client and also get the source for translation, ensures that the translations are complete, reviewed and the final version is published. They also have the responsibility of assigning the translation work to editors. This hierarchy provides a modular approach for large translation contents. For example, Murshid involves a large Web source partners which are candidate for translation. In one specific source, contents is a collection of scientific articles in various domains but categorized into say Geography, Chemistry, Computer, etc.. Murshid assigns an expert Editor for each domain area. A project manager can view the current progress of the entire Website translation project. They have a view over each editor's section. Once the collection of content in each section of the website has been vetted by the Editor, the translation project is said to be complete.

c) *Editors*: They are best seen as area consultants and subject experts in the area and the target language for translation. Their job is to receive content from project manager and examine translations returned by translators. This operation is labeled as "vetting". Editors have expertise in specific areas. Multiple translations may be vetted by a single editor. Editors can examine the history of a candidate article, including all translations, revisions, and vetting. However, their vetting only applies to the most recent version of the translation. Editors can view the current progress of the Website translation project in their particular area. They can see which articles have been translated but not yet vetted, and they can pinpoint exactly which translated article they wish to vet. Editors are also responsible to assign source content to translators. Content is shown to editors in both input and output format. They may then make changes to the translation or send back to the original translator. So the editor and translator can discuss the changes in a cyclic process. Once the editor approves the translation, he states the resource as "vetted". A vetting mark is made in the translation database and a notice is sent to the translator. Translators work is stored in test version of the translation database. Once materials have been vetted by an editor, they are stored in the production version of the translation database. Subsequent emendation or expansion of a vetted article can occur in a new workflow.

d) *Translators*: They must be subject experts i.e. they are experts in the content to be translated. Translation can be carried out in a Push or Pull process. A translator may be assigned a set of article entries by the editor. Alternatively, translators may request assignments either in the form of a range (of pending articles for translation), or in the form of a topic, such as "Biology". This request indicates whether the translator wants

to work in the test or the production database. Web services are invoked upon requests generation. Editors are notified and may subsequently initiate or modify translation assignments. Given an assignment the corresponding translator proceeds with the actual translation task. The translator sees a placeholder to upload his translation in the expected format. The input is initially checked for completeness before giving an opportunity to make last changes prior to storing the translated item in the translation database. A translator may not modify a translation previously submitted. A submitted translation is vetted by an editor. Prior to submit a translation, translators may solicit the service of proofreaders or reviewers.

e) *Reviewers*: They are responsible for ensuring the quality of translation. They will proofread content and make grammatical or other linguistic or cultural amendments to the translated content. Eventually reviewers may instead make suitable comments and suggestions to be considered by translators. Reviewer may either approve or reject the translation.

f) *TMS Administrator*: The TMS Administrator would have overall responsibility of the TMS environment. He provides user access rights, sets up new user accounts and roles, supports and trains TMS related personnel.

4 Trust Specification

Trust remains a crucial component in any system. It is essential that users be able to appraise service reputation. This is particularly important in a translation service, where the quality of the service outcome can often be measured only after the content deployment. To this end we propose a reputation metric to measure trust of our proposed TMS services. We define reputation to be a quality property, composed of three sub-properties: Service Availability, Quality of Translation and Frequency of Service Invocation. Each of these components is associated to a weighted value. All values are then aggregated to produce a measure of reputation. Service availability represents the probability that the Web Service is accessible (available for use) quantified by the percentage of time the service is operating. Quality of translation is based on the speed, accuracy and presentation of translated material and is a composite value based on end users feedbacks. The frequency of invocation represents the number of times a service operation is invoked. The higher the level of invocation, the higher is the level of customers' trust in the service provider reputation.

Based on the value of the reputation, we classify the Web service into three qualitative categories: Strong, Satisfactory, and Weak. We specify reputation property of a translation Web service by aggregating composing sub-properties, which are embedded in an XML representation that is included in the WSDL document of the TMS Web service. The properties' values are defined for each operation provided by the TMS Web service. The figure below presents the XML description of these quality properties for TMS Web service.

```
<Trust-WS name= "TMS">
        <Category name="Strong">
                <operation      name= "Translate"
                                Availability = 95%
                                Quality of Translation = Excellent
                                Frequency of Invocation = 1
                </operation>
                <operation      name= "Review"
                                Availability = 90%
                                Quality of Translation = Very Good
                                Frequency of Invocation = 1
                </operation>
        </Category >
        < Category name="Satisfactory">
                <operation      name= "Translate"
                                Availability =80 %
                                Quality of Translation = Very Good
                                Frequency of Invocation =1
                </operation>
                <operation      name= "Review"
                                Availability = 75%
                                Quality of Translation = Good
                                Frequency of Invocation = 2
                </operation>
        </ Category >
        <Category name="Weak">
                ……
        </Category >
</Trust-WS >
```

Fig. 3. XML representation of TMS Web service: Trust property

5 Prototype and Case Study

To validate some of the features of our TMS based Web services model, we provide a brief illustration of a case study for translating web content stored in CMS or content repository into Arabic language. We first show the XML specifications of implemented web services and the structure of the exchange XLIFF messages.

5.1 Murshid Web Service

Translation web services provide a range of operations such as `submitJob` and `retrieveJob` where the first operation allows a client to request a translation from

a web service and the second operation allows a client to get the translated content from a web service. An excerpt of WSDL specification of Murshid Web service is illustrated in Figure 4. It describes the set of available operation interfaces, and their related binding information.

```
<?xml version="1.0" encoding="utf-8" ?>
- <wsdl:definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" ... >
- <wsdl:types>
- <s:schema elementFormDefault="qualified" targetNamespace="http://www.Murshid.ae">
- <s:element name="Translate">
- <s:complexType>
      ..........
   </wsdl:types>
- <wsdl:message name="TranslateSoapIn">
        ......
   </wsdl:message>
- <wsdl:message name="TranslateSoapOut">
       .....
   </wsdl:message>
- <wsdl:portType name="TranslateServiceSoap">
- <wsdl:operation name="Translate">
 <documentation xmlns="http://Murship.ae/wsdl/"> Convert text from one language to Arabic language.
       Supported language is English to Arabic. </documentation>
 <wsdl:input message="tns:TranslateSoapIn" />
 <wsdl:output message="tns:TranslateSoapOut" />
     </wsdl:operation>
    </wsdl:portType>
- <wsdl:binding name="TranslateServiceSoap" type="tns:TranslateServiceSoap">
 <soap:binding transport="http://schemas.Murshid.ae/soap/http" style="document" />
- <wsdl:operation name="Translate">
 <soap:operation soapAction="http://www.Murshidwebservice.ae/Translate" style="document" />
- <wsdl:input> .... </wsdl:input>
- <wsdl:output> ..... </wsdl:output>
  </wsdl:operation>
  </wsdl:binding>
- <wsdl:binding name="TranslateServiceHttpPost" type="tns:TranslateServiceHttpPost">
 <http:binding verb="POST" />
   ..........
 </wsdl:operation>
 </wsdl:binding>
                  .......
 </wsdl:port>
 </wsdl:service>
 </wsdl:definitions>
```

Fig. 4. A part of WSDL document of Murshid Web services

5.2 XLIFF Specification for Murshid Web Service

To illustrate an example of XLIFF specification used to translate Web content to Arabic language, Figure 5 presents an itemized menu together with a login frame and sign up a new account as obtained by the proposed TMS web service implemented in Murshid case study.

Fig. 5. Example of source content in English translated to Arabic using XLIFF specification

6 Conclusion

This paper has addressed many of the problems inherent to web content globalization through the provision of Translation Management Services (TMS). We also introduced a measure of trust to quantify the Quality of Service of the proposed translation services. Based on standardized web service specifications, this paper presented a streamlined approach to translating and localizing web content based on a consistent

workflow and well defined stakeholders roles. We first introduced the TMS model, as a systematic management of translating source web material to a given target language. We validated an instance of the TMS where the target language is Arabic in the context of a case study to demonstrate the operations TMS web services. The case study has further illustrated how TMS services could be used to effectively manage complex translation processes, while maintaining QoS attributes pertaining to a translation process, such as proof reading, traceability and transparency. Our solution presents a comprehensive enterprise approach to web translation, which seamlessly interoperates CMS functions with our proposed translation workflows, leading all the way to translation personnel whose roles have been defined in this paper.

To further specify trust, we plan to integrate the standard WS-Policy framework [20], which allows web services to advertise their Quality of Service (QoS), and for web service consumers to specify their QoS requirements. WS-Policy defines a policy to specify requirements and capabilities that manifest on the wire (for example, authentication scheme and transport protocol), as well as capabilities which are critical to proper service selection such as QoS characteristics. The advertised policy contributes to match a service requesters with the expected service quality, and hence the expected level of trust from the service outcome. Our future work explores further extensions of the TMS to provide mechanisms for trust based on WS Policy standard to further adhere to the OASIS prescribed standards.

References

1. Antonopoulios, V., Demiros, I., Cafatalayannls, G., Piperidis, S.: Integrating Translation Technologies Towards a Powerful Translation Web Service. In: Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems Singapore, December 1-3 (2004)
2. OASIS Translation Web Services Technical Committee, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=trans-ws
3. Translation Web services specification, Standard Specification Draft (June 2008), http://www.oasis-open.org/committees/workgroup.php?wg_abbrev=trans-ws
4. Reynolds, P.: Web Services and Globalization Management: A Case Study. Multilingual Computing & Technology, num. 73 16(5), July/August (2005)
5. Using Web Services for Translation, A white paper on the Translation Web Services standard (April 2006), http://www.oasis-open.org/committees/download.php/21054/TWS%20Whitepaper.doc
6. XLIFF Specification, http://docs.oasis-open.org/xliff/xliff-core/xliff-core.html
7. Dilts, D.: Successfully Crossing the Language Translation Divide. In: (ACM) Proceedings of the 19th Annual International Conference on Computer Documentation (2001)
8. Marcus, A., Gould, E.W.: Crosscurrents: Cultural Dimensions and Global Web User-Interface Design. ACM Interactions 7(4) (July 2000)
9. Levanthal, L., et al.: Designing for Diverse Users: Will Just a Better Interface Do? In: SIGCHI, pp. 191–192. ACM Press, New York (1994)
10. Nielsen, J.: Usability Testing of International Interfaces. Elsevier Science Publishers Ltd., UK (1990)

11. AltaVista Babel Fish Translation, http://babelfish.altavista.com
12. Merrill, C.K., Shanoski, M.: Internationalizing Online Information. In: SIGDOC, pp. 19–25. ACM Press, New York (1992)
13. Volk, M.: The Automatic Translation of Film Subtitles. In: A Machine Translation Success Story? Proceedings of Languages amd the Media, 7th International Conference and Exhibition on Language Transfer in Audiovisual Media, Berlin, Germany (2008)
14. TidBITS Translations, http://www.tidbits.com/about/translations.html
15. SOL: Byzantine Lexicography, http://www.stoa.org/sol/
16. OmniWeb, The Omni Group, http://www.omnigroup.com/applications/omniweb/
17. The Translation Project, http://www.iro.umontreal.ca/contrib/po/HTML/index.htm
18. Google in Your Language, http://services.google.com/tc/Welcome.html
19. Globalize your On Demand Business, IBM, http://www-01.ibm.com/software/globalization/topics/webservices/translation.jsp
20. Web Services Policy, http://www.w3.org/Submission/WS-Policy/