

Web Service Integration for Next Generation Localisation

David Lewis, Stephen Curran,

Andy Way

Reinhard Schäler

Kevin Feeney, Zohar Etzioni,

John Keeney

Centre for Next Generation Localisation

Knowledge and Data Engineering
Group

School of Computing

Centre for Localisation
Research

Trinity College Dublin, Ireland

Dublin City University,
Ireland

University of Limerick,
Ireland

{Dave.Lewis|Stephen.curran|Kevin.Feeney|etzioniz|John.Keeney}@cs.tcd.ie

away@computing.dcu.ie

Reinhard.Schaler@ul.ie

Abstract

Developments in Natural Language Processing technologies promise a variety of benefits to the localization industry, both in its current form in performing bulk enterprise-based localization and in the future in supporting personalized web-based localization on increasingly user-generated content. As an increasing variety of natural language processing services become available, it is vital that the localization industry employs the flexible software integration techniques that will enable it to make best use of these technologies. To date however, the localization industry has been slow reap the benefits of modern integration technologies such as web service integration and orchestration. Based on recent integration experiences, we examine how the localization industry can best exploit web-based integration technologies in developing new services and exploring new business models

1 Introduction

Research and development of natural language processing technologies are leading to a variety of advances in areas such as text analytics and machine translation that have a range of commercial applications. The Localization Industry in particular, is strategically well placed to make good use of these advances as it faces the challenge of localizing accelerating volumes of digital content that is being targeted at increasingly global markets of this content. It needs to exploit the benefits of NLP technologies to reduce the cost of translation and minimise the time to market of this digital content. Furthermore, where the localization industry best learns how to efficiently and flexibly employ NLP

technologies in the localization of digital content it will be ideally placed to develop new services and exploit new business opportunities offered by the WWW. In particular, today's localization techniques are not able to keep pace with the WWW's ability to dynamically compose and personalize existing content and to support rapid development of large volumes of user generated content. To meet this challenge, localization processes must effectively employ NLP to move from manually centered, professional batch activities to highly automated, highly participative continuous activities. To do this, the technologies of the WWW need to be employed to dynamically combine NLP technologies and leverage different levels of human linguistic abilities and knowledge to best accomplish the task at hand.

In this paper we examine how this vision, which we term *Next Generation Localization*, can be supported by current web-based, service-oriented software integration techniques such as web service integration and orchestration. Based on recent integration experience we review the current issues in using open interoperability standards and web services to the integration of commercial localization platforms and NLP software. We then describe some generic definitions for NLP web services and how these provide flexibility in developing new localization service compositions. Finally, we outline the major software integration challenges facing the localization industry and describe how these are being addressed at Ireland's Centre for Next Generation Localization (CNGL).

2 Next Generation Localization

Traditional localization technologies and workflows are no longer able to cope with the escalating growth in volume. Traditional localization methods are not adequate to manage, localize and personalize unpredictable, on-line, multilingual, digital content. Machine Translation (MT) needs to be integrated into translation and post-editing workflows together with human translators. Novel machine-learning-based language technologies can automatically provide metadata annotations (labels) to localization input in order to automate localization standardization and management.

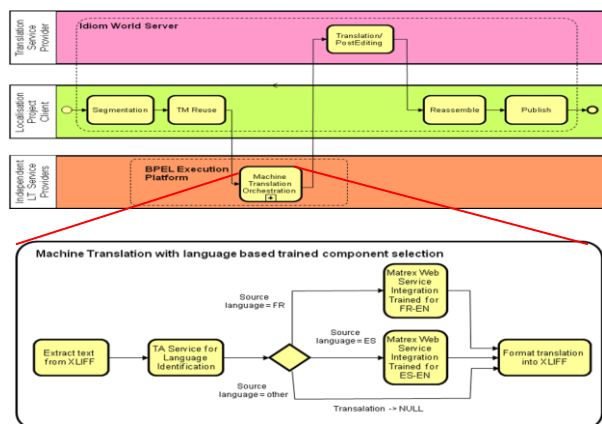


Figure 1: Example use of Web Service Orchestration in a Localisation Workflow

For Next Generation Localisation to be achieved, the individual components need to be interoperable and easily reconfigurable. The complexity of the resulting systems poses substantial software engineering challenges and crucially requires detailed user requirement studies, technical and user interface standards, as well as support for rapid prototyping and formative evaluation early on in the software lifecycle. Blueprints for an industrial environment for Next Generation Localisation, which we term a *Localisation Factory*, are needed to guide the development of localisation services systems integrating advanced language, digital content and localisation management technologies. However, in order to successfully achieve the goal of technical interoperability these services crucially needs to be supplemented by standardised localisation processes and workflows for the Localisation Factory. Figure 1 gives an overview of a typical localisation workflow, that would be used for translating the content such as

the use manual for a product, into multiple languages for different target markets. Typically this involves segmenting the content into sentences, looking up previously translated sentences from a Translation Memory (MT), before passing untranslated segments to a Machine Translation (TM) service to generate further candidate translations. Next, the job is passed to professional translators, who can accept automated translations or provide their own translations. Current practice in performing such workflows uses localisation platforms such as SDL's Idiom WorldServer to integrate Translation Memory databases, Machine Translation packages and the routing of jobs to translators who typically work remotely under the management of a localisation service provision agency.

The localization industry has already undertaken a number of separate standardization activities to support interoperability between different localisation applications. The Localisation Industry Standards Association (LISA – www.lisa.org) has developed various localisation standards:

- Translation Memory Exchange (TMX) for exchanging TM database content. Many TM tool providers have implemented support for TMX in their products.
- Term Base eXchange (TBX): XML Terminology Exchange Standard. An XML linking standard, called Term Link, is also being investigated.
- Segmentation Rules eXchange (SRX), for exchanging the rule by which content is originally segmented. There has been very little support to date for SRX because segmentation is the main component that distinguished TM tools. Segmentation has direct consequences for the level of reuse of a TM. A TM's value is significantly reduced without the segmentation rules that were used to build it.
- Global information management Metrics eXchange (GMX): A partially populated family of standards of globalization and localization-related metrics

The Organization for the Advancement of Structured Information Standards (OASIS – www.oasis-open.org), which produces e-business standards has had a number of initiatives:

- XML Localisation Interchange File Format (XLIFF): XLIFF is the most common open standard for the exchange of localisable con-

tent and localisation process information between tools in a workflow. Many tool providers have implemented support for XLIFF in their products.

- Trans-WS for automating the translation and localization process as a Web service. There has not been much adoption of this standard. Work on the development and maintenance of the standard seems to be at a stand-still.
- Open Architecture for XML Authoring and Localization: A recently started group looking at linking many existing localisation standards

The W3C, which develops many web standards, has an Internationalisation Activity (www.w3.org/International) working on enabling the use Web technologies with different languages, scripts, and cultures. Specific standardisation includes the Internationalisation Tag Set to support internationalisation of XML Schema/DTDs.

To date, therefore, standard localisation processes and workflows addressing common interoperability issues have not yet been widely adopted. Outside of proprietary scenarios, digital publishers and service providers cannot integrate their processes and technologies and cannot provide independent performance measures. This implies lost business opportunities for many and missed opportunities for significant performance improvement for most of the stakeholders. We now examine how web services may help improve this situation.

3 Service Oriented Localization Integration

The Centre for Next Generation Localisation [cngl] is developing a number of systems in order to investigate the issues that arise in integrating centralized workflows with community-based value creation. It aims to make full use of Service-Oriented Architecture [erl]. This advocates software integration through well defined functional interfaces that can be invoked remotely, typically using the Web's HTTP protocol with input and output parameters encoded in XML. The W3C have standardized an XML format, The Web Service Description Language (WSDL), for describing and exchanging such service definitions. Web services can be composed into more complicated applications using explicit control and data flow models that can be directly executed by workflow engines. This allows new

workflow applications to be defined declaratively and immediately executed, thus greatly reducing the integration costs of developing new workflows and increasing the flexibility to modify existing ones. Such web-service based service composition is known as Web Service Orchestration. OASIS has standardized web service orchestration language called the Business Process Execution Language (BPEL), which has resulted in the development of several commercial execution platform and BPEL workflow definition tools, which support workflow definition through drag-and drop interfaces. In CNGL, web services and web service orchestration are used for integrating components and operating workflows between potential partners in the commercial localization value chain. This provides a high degree of flexibility in integrating the different language technologies and localization products into different workflow configurations for the project, while avoiding reliance on any single proprietary platform. As an initial exploration of this space a system integration trial was undertaken. The use of BPEL for integrating NLP software has previously been used in the LanguageGrid project, but is a purely in support of academic research integration. Our work aimed flexibility instantiate commercial localisation workflow using NLP software wrapped in services that are orchestrated using BPEL, while, as indicated in Figure 1, still integrating with commercial localisation workflow tools. This exploration also included extending the human element of the localisation workflow by soliciting translations from a body of volunteer translators. This is seen as more appropriate if the required translation is not time constrained and it often forms part of a customer relationship strategy. Quality management may require involvement of volunteer post-editors, and incomplete or poor translations may ultimately still need to be referred to professional translators.

Thus our workflows can be configured to operate in parallel to provide alternative translations. In the professional localization workflow, after the MT stage, the candidate translation would be returned to the SDL Worldserver platform via which professional translators and post-editors are able to complete the task. In the crowd-sourcing variation, this manual step is instead performed by passing the job to a similar application implemented as a

plug-in to the Drupal collaborative content management system.

Our implementation uses the XLIFF format as a standard for encapsulating the various transformations that happen to a resource as it passes through the localisation process. It should be noted, however, that support for XLIFF is partial at best in most localisation tools. Where the standard is supported, there are often different, specific flavours used, and embedded elements within the XLIFF can be lost as the resource passes through various stages in the process. Another problem with incorporating current tools in our service-oriented framework is that some of them, such as IBM's UIMA, are designed to function in a batch mode – which does not map cleanly to services. Nevertheless, despite a range of practical problems, it was in general possible to engineer service front-ends for most of these tools so that they can be integrated into a composable service infrastructure. In the following section we proceed to detail the design of the generic web services we defined for this system and discuss the option undertaken in their implementation.

3.1 Web Service Definitions

The OASIS TWS working group remains the only real attempt to define web-services to support the localization process. However, TWS has a limited scope. Rather than aiming to support the dynamic composition of language services into flexible localization workflows, it concentrates on supporting the negotiation of “jobs” between service providers. It is primarily intended to support the efficient out-sourcing of localization and translation jobs and it does not address the composition of language-services to form automated workflows.

Therefore, in order to deploy web-services to support such composition, there is little standardisation to rely on. Thus, a first step in addressing the problem is to design a set of web-services and their interfaces suitable for the task. In designing these services, it is worthwhile to recall the general goals of service-oriented architectures; the services should be designed to be as flexible and general as possible and they should neither be tightly coupled to one another, nor to the overall system which they are part of. Furthermore, in keeping with the general trends in service designs [foster], variability

in service behavior should generally be supported through the passed data-structures rather than through different function signatures.

Bearing these design goals in mind, we can begin to analyse the basic requirements of localisation with a view to translating these requirements into concrete service definitions. However, in order to further simplify this task, we adopt certain assumptions about the data-formats that will be deployed. Firstly, we assume that UTF-8 is the universal character encoding scheme in use across our services. Secondly, we assume that XLIFF is employed as the standard format for exchanging localization data between different parts of the localisation process.

XLIFF is primarily focused on describing a resource in terms of source segments and target segments. Essentially, it assumes the following model: a localization job can be divided up into a set of translatable resources. Each of these resources is represented as an XLIFF file. Each resource can be further sub-divided into a sequence of translatable segments (which may be defined by an SRX configuration). Each of these source segments can be associated with a number of target segments, which represent the source segment translated into a target language. Finally, XLIFF also supports the association of various pieces of meta-data with each resource or with the various elements into which the resource is sub-divided.

This simple basic structure allows us to define a very simple set of general web-services, each of which serves to transform the XLIFF in some way. These three basic classes of services transform the XLIFF inputs in the following ways:

1. Addition of target segments.
2. Sorting of target candidates
3. Addition of meta-data.

Thus, we adopt these service-types as the set of basic, general service interfaces that our services will implement. They allow us to apply a wide range of useful language-technology processes to localization content through an extremely simple set of service interfaces. To give some examples of how concrete services map onto these basic interfaces:

- A machine translation service is a manifestation of type 1. It adds translations, as target segments, for source segments in the XLIFF file

- A translation memory leveraging service is, similarly, implemented as a service of type 1. It can be considered as a special case of a translation service.
- Our basic service-design supports the application of multiple TM and MT services to each XLIFF file, potentially producing multiple translation candidates for each source segment. There are various situations where there is a need to order these candidates – for example to choose which one will actually be used in the final translation, or to present a sorted list to a human user to allow them to most conveniently select the candidate that is most likely to be selected by them. These services can be implemented using the common type 2 interface.
- A wide range of text analytics service can be implemented as services of type 3. For example, domain identification, language identification and various tagging services are all instantiations of this type.

Although these service types are generic, in terms of the transformations that they apply to the XLIFF content, they may be very different in terms of their management and configuration. Thus, it is neither possible nor desirable to devise generic management interfaces – these interfaces need to be tailored to the particular requirements of each specific service. Thus, each service really consists of two specifications – an implementation of the generic interface which allows the service to be easily integrated as a standard component into a workflow that transforms the XLIFF content, and a specific interface that defines how the service can be configured and managed. The following section provides several examples of specific services and their management interfaces.

Although XLIFF provides significant support for management of the transformation of resources as they proceed through the localisation workflow, it is not a universal solution. It is an inherently resource-oriented standard and it is thus not well suited for the aggregation of meta-data that has broader scope than that of the translatable resource. For example, in the course of a localisation workflow, we may wish to store state information relating to the user, the project, the workflow itself or various other entities that are not expressible as XLIFF resources. Therefore, a service-oriented localization workflow has a need for a service which allows the setting and retrieving of such me-

ta-data. The following section also includes a basic outline of a service which can provide such functionality across the localization workflow.

Finally, it should be pointed out that BPEL does not provide a universal solution to the problem of constructing workflows. It is primarily designed to facilitate the orchestration of automated web-services and does not map well to human processes. This has been acknowledged in the proposed BPEL4People extension and the incorporation of better support for human tasks is also a key motivating factor for the development of the YAWL workflow specification language – a BPEL alternative [vanderaalst]. To overcome this limitation, we have designed a general purpose service which allows components to query the state of human tasks within the workflow – this allows workflows to be responsive to the progress of human tasks (e.g. by redirecting a task that is taking too long).

3.2 An MT Web Service

As part of our work within CNGL in the development of a Localisation Factory we have engineered a web service capable of leveraging translations from multiple automated translation components. The service operates by taking in an XLIFF document, iterating the segments of the document and getting a translation from each of the translation components for each segment. These translations are attached to the segment within the XLIFF and the service returns the final XLIFF document back to the client. The service can be configured to use any permutation of the automated translation components depending on the workflow in which the service finds itself operating. Some translation components may be inappropriate in a given workflow context and may be removed. The service also allows for the weighting of translations coming from different translation components so that certain translations are preferred above others.

The service implementation leverages translation from two open web based translation systems Microsoft Live Translator [mslive] and Yahoo Babelfish [babelfish]. Microsoft Live Translator can be accessed through a web service interface. Yahoo Babelfish has no web service interface so getting back translations is implemented through a screen-scraping technique on the HTML document returned.

The service also makes use of MaTrEx [matrex], a hybrid statistical/example-based machine translation system developed by our partner university Dublin City University. MaTrEx makes use of the open-source Moses decoder [moses]. Translation models are created using MaTrEx and are passed to the Moses decoder which performs that translation from source to target language. We took the Moses decoder and wrapped it in a web service. The web service pipes segments for translation to Moses which responds with translations. This translation model is produced based on aligned source and target corpora of content representative of the content passing through the workflow.

Finally we have taken a translation memory product LanguageExchange from Alchemy, an industrial partner within the project, and added that to the list of automated translation components available to our service. This allows any previous human translations to be leveraged during the automated translation process.

The service is engineered using Business Process Execution Language (BPEL) to orchestrate the calling of the various translation components that compose the service. BPEL allows those managing the service to easily compose a particular configuration of the service. Translation components can be easily added or removed from the service. The tool support around BPEL means that the user does not need a background in programming to develop a particular configuration of the components.

One problem we encountered implementing the MT service as a wrapper around existing components was that they are unable to handle internal markup within the segments. Segments passing through a localisation workflow are likely to contain markup to indicate particular formatting of the text. The machine translation components are only able to handle free text and the markup is not preserved during translation. Another problem encountered in using free web services over the Internet was that implementations did not encourage volume invocations, with source IP addresses requesting high volumes being blacklisted.

3.3 A Text Analytics Web Service

We have implemented a generic text-categorization service to provide text-analytic support for localization workflows. It takes an XLIFF file as input and produces an XLIFF file as output, transforming it by adding meta-data (a type 3 transform). The meta-data can be added either on a file-basis or on a segment basis, depending on the requirements of the workflow as expressed in the service's configuration. The service provides a simple and generic XLIFF transformation as part of the localization workflow, while the management interface provides flexible configurability.

The management interface is designed in order to support multiple text analytic engines, each of which can support multiple categorization schema at once. Our implementation uses two text engines, the open source TextCat package [textcat] and IBM's Fragma software [fragma]. The following operations are provided by the service:

Operation createSchema: The createSchema function creates a new categorisation schema based on a provided set of training data, which can optionally be provided by an RSS feed for ongoing training data updates.

Operation getEngines: This returns a list (encoded in XML) of the categorisation engines that are available to the Service. This allows the client to specify that a specific categorisation engine be used in subsequent requests.

Operation viewSchema: This returns a list of the categories contained within a schema (and the details of the engine that was used to create it).

Operation addData: This operation adds a piece of training data to a categorisation schema - i.e. it allows components to tell the service that a piece of text has a known category of categoryID according to the schema with schemaID.

Operation categorise: This provides a categorisation of text provided as an XLIFF segment, according to a specified schema taken from the list supported by the service.

3.4 A Crowd-sourcing Web Service

In order to allow the localization workflow to incorporate crowd-sourcing, by which we mean collaborative input from a volunteer web-based user-community, we have designed and implemented a web-service interface. This interface is designed to

allow stages in the localization job to be handed off to such a community. From the point of view of the workflow, the important thing is that the localisation requirements can be adequately specified and that the status of the job can be ascertained by other elements in the workflow – allowing them to react to the progress (or lack thereof) in the task and, for example, to allow the job to be redirected to another process when it is not progressing satisfactorily.

Our service design is focused on supporting crowd-sourcing, but it is intended to extend it to offer general-purpose support for the integration of human-tasks into a BPEL workflow. It serves as a testbed and proof of concept for the development of a generic localization human task interface. The initial specification has been derived from the TWS specification [tws], but incorporates several important changes. Firstly, it is greatly simplified by removing all the quote-related functions and replacing them with the RequestJob and SubmitJob functions and combining all of the job control functions into a single updateJob function and combining the two job list functions into one.

TWS, as a standard focused on support for localization outsourcing – hence the concentration on negotiating ‘quotes’ between partners. Our requirements are quite different – we cannot assume that there is any price, or even any formal agreement which governs crowd-sourcing. Indeed, in general, a major problem with TWS which hindered its uptake is that it assumed a particular business model – in practice localization jobs are not so automated, nor so quick that automated price negotiation is a particularly desired feature. Such information can be incorporated into a Job Description data structure, but a generic human-task interface should not assume any particular business model – hence the significant changes between our API and that of TWS. Nevertheless, there is much clear and well-structured thinking contained in the TWS standard – how best to describe language pairs, jobs and various other commonly referenced ideas in a localization workflow. By using TWS as a base, we can take advantage of all of that work rather than designing our own data-structures from scratch. The main operation are as follows:

Operation requestJob: The JobDescription input parameter is an XML format which contains details of the job that is being requested. The returned

datatype is the details of the job that is offered by the service. These are not necessarily the same. For example, the requested job might contain several language pairs, but the returned description might not contain all of these language pairs as some of those requested might not be available in the service. Generally, it can be assumed that the service will make its “best effort” to fulfill the requirements and the returned data will be as close as it can get to the requirements submitted.

Operation submitJob: This operation works exactly as the one above, except for the fact that it submits the job to the service with the particular JobDescription required and receives back the JobDescription that will actually be carried out.

Operation retrieveJobList: This accepts a JobDescription input parameter, an XML format which contains a ‘filter’ on the various active jobs. The operation will return a list of all of the jobs which match that specified in the JobDescription argument.

Operation updateJob: A JobDescription input parameter is an XML format which contains a description of the various changes to the job that are being requested. The function will return a description which details the new, updated state of the job (note that the service does not have to follow all the requested changes and might ignore them).

Operation retrieveJob: A JobDescription input parameter is an XML format which contains a ‘filter’ on the various jobs. The operation returns a URI from which the client can retrieve the localised content corresponding to the filters.

Operation associateResource: This functions associates a resource (TM / Glossary / etc) with a particular job. The returned value is the URI of the resource (which may be different than the passed ResURI). The types of resource supported will need to be decided upon.

4 Future Work: Translation Quality

The next challenge to applying these techniques to workable industrial workflows is to fully address the metrology of such workflows. The current approach does not support the instrumentation of web services to provide quality measurements. Further, such quality measures need to be provided in a way that is relevant to the quality of the workflow as a whole and the business-driven key performance indicators which it aims to support.

However, the integration of translation quality metrics across different forms of workflow and different industrial workflow components and linguistic technologies has been widely identified as requiring considerable further investigation. Even the most basic metric used in commercial workflow, the word count against which translation effort is estimated, is calculated differently by different workflow systems. This particular case has already been addressed by LISA though its proposal for Global information management Metrics eXchange (GMX) [gmx].

It is hardly surprising, therefore, that closing the gap between the metrics typically used by MT system developers and what is needed to support the use of MT in commercial localization workflows is likely to be even more challenging. For example, metrics such as BLEU [bleu] are well-understood by MT developers used to participating in large-scale open MT evaluations such as NIST; a BLEU score of 0.8 (say) means either that one's MT system is extremely good, or that the task is quite simple, or both, or even that there are a large number of reference translations against which the system output is being compared. On the other hand, a score of 0.2 means that the quality is poor, that there is probably only one reference translation against which candidate translations are being evaluated, or that the task is a very complex one.

However, neither score means anything (much) to a potential user. In the localization industry, Translation Memory is much more widely used, and there users and vendors use a different metric, namely fuzzy match score, i.e. how closely a previously translated source sentence matches the current input string. Users typically 'know' that a score of around 70% fuzzy match is useful, whereas for a lower scored sentence it is likely to be quicker to translate this from scratch.

One of our research goals in the CNGL is to bring these two communities closer together by developing a translation quality metric that speaks to both sets of people, developers and users. One step in the right direction might be the Translation Edit Rate metric [ter], which measures the number of editing commands (deletions, substitutions, and insertions) that need to be carried out in order to transform the MT output into the reference translation(s). This is being quite widely used in the MT community (cf. the Global Autonomous Language Exploitation (GALE) project) by MT developers,

and speaks a language that users understand well. User studies will very much inform the directions that such research will take, but there are reasons to believe that the gap can be bridged.

Supposing then that such hurdles can be overcome, broadly speaking, the quality of a translation process might be dependent on multiple factors, each of which could be measured both intrinsically and extrinsically, including;

- Source and destination languages
- Content domain
- Diversity of vocabulary
- Repetitiveness of text
- Length and complexity of sentences
- Availability of relevant translation memories
- The cost and time incurred per translated word

Often control of quality of the translation process can be impacted most directly by the quality of the human translators and the degree of control exerted over the source text. Different levels of linguistic quality assurance may be undertaken and post-editors (who are often more experienced translators and therefore more expensive) are involved in handling incomplete or missing translations. However, even in professional translation environments, translation quality is regarded as relatively subjective and exact measurement of the quality of translation is therefore problematic.

5 Conclusion

In this paper we have discussed some the challenges faced in taking a web service integration and orchestration approach to the development of next generation localization workflows. Based on our experiences of using these approaches to integrate both existing localization products and cutting edge research prototypes in MT, TA and crowd-sourcing, new, innovative localisation workflows can be rapidly assembled. The maturity of the BPEL standard and the design of general purpose, reusable web service interfaces are key to this success.

Acknowledgments: This research is supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Trinity College Dublin.

References

- [babelfish] Yahoo Babelfish Machine Translation <http://babelfish.yahoo.com/> 6th Feb 2009
- [drupal] Drupal Content Management System <http://www.drupal.org> 6th Feb 2009
- [bleu] Kishore Papineni, Salim Roukos, Todd Ward and Wei-Jing Zhu. 2002. In 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA., pp.311—318.
- [bpel] Web Services Business Process Execution Language Version 2.0, OASIS Standard, 11 April 2007, Downloaded from <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-0S.html> 6th Feb 2009
- [erl] Erl, Thomas, Service-oriented Architecture: Concepts, Technology, and Design. Upper Saddle River: Prentice Hall 2005
- [foster] Foster, I., Parastatidis, S., Watson, P., and Mckeown, M. 2008. How do I model state?: Let me count the ways. *Commun. ACM* 51, 9 (Sep. 2008), 34-41.
- [fragma] Alexander Trousov, Mayo Takeuchi, D.J.McCloskey, http://atrousov.com/uploads/TSD2004_LangID_word_fragments.pdf 6th Feb 2009
- [gmx] Global Information Management Metrics Volume (GMX-V) 1.0 Specification Version 1.0, 26 February 2007, downloaded from <http://www.xml-intl.com/docs/specification/GMX-V.html> on 6th Feb 2009
- [langexchange] Alchemy Language Exchange http://www.alchemysoftware.ie/products/alchemy_language_exchange.html 6th Feb 2009
- [matrex] MaTrEx Machine Translation - John Tinsley, Yanjun Ma, Sylwia Ozdowska, Andy Way. http://doras.dcu.ie/559/1/Tinsleyetal_WMT08.pdf
- [moses] Moses decoder <http://www.statmt.org/moses/> 9th March 2009
- [mslive] Microsoft Live Translator <http://www.windowslivetranslator.com/> 6th Feb 2009
- [ter] Matt Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, Cambridge, MA., pp.223—231.
- [textcat] Java Text Categorisation <http://textcat.sourceforge.net/> 6th Feb 2009
- [tbx] Termbase eXchange Format <http://www.lisa.org/Term-Base-eXchange.32.0.html> 6th March 2009
- [tmx] Translation Memory eXchange <http://www.lisa.org/Translation-Memory-e.34.0.html> 6th March 2009
- [tws] Translation Web Services Specification: <http://www.oasis-open.org/committees/download.php/24350/trans-ws-spec-1.0.3.html>
- [vanderaalst] Van Der Aalst, W.M.P. Ter Hofstede, A.H.M. “YAWL: Yet another workflow language” *Information Systems*, Volume 30, Issue 4, June 2005, Pages 245-275
- [xliff] XML Localisation Interchange File Format <http://docs.oasis-open.org/xliff/v1.2/os/xliff-core.html> 6th March 2009