

A Web Services approach to software localisation.
Bringing software localisation tools from the desktop to the
cloud

Jaime Mateos

A dissertation submitted to the University of Dublin,
in partial fulfilment of the requirements for the degree of
Master of Science in Computer Science

2010

Declaration

I declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed: _____
Jaime Mateos
13/09/2010

Permission to lend and/or copy

I agree that Trinity College Library may lend or copy this
dissertation upon request.

Signed: _____

Jaime Mateos
13/09/2010

Abstract

Software localization tools available today are, by and large, implemented as desktop applications. The leading ones have been in the market for well over 10 years. During this time they have expanded their functionality, naturally trying to add more value. As a result, they are function rich applications that target a variety of users with different roles.

This research first investigates what cohesive pieces of functionality are involved in software localization; and secondly explores how they can be provided as web services. The potential benefits of using a Service Oriented Architecture include lessened vendor and tool lock-in, a more flexible and agile software architecture for developing innovative and tailored localisation tools, and increased automation by implementing workflows through web service orchestration. This research studies how and to what extent this benefits are realized and identifies a number of barriers to adoption.

The research looks at the current landscape of localisation and web services standards to identify the ones that can be used to support this architecture. It studies possible shortcomings in those standards as well as new areas where standards would be highly desirable.

The research includes a survey analysing the support for a Service Oriented Architecture found in the principle software localisation tools in use today. This evaluates the tools degree of automation, network access to functionality, expandability, platform independence, standards use and composability.

The research also includes a summary of a set of interviews conducted with leading industry experts from Independent Software Vendor localisation divisions, Localisation Service Providers and tool vendors. These interviews were used to gauge their attitudes to the benefits and barriers to the adoption of web services, and covered issues such as scalability, availability, use of standards, performance and security.

Table of Contents

1. Introduction.....	1
2. State of the art.....	5
2.1 Related work.....	5
2.2 Web services standards.....	6
2.3 Localisation Standards.....	7
2.4 Localisation tools.....	8
2.5 Interviews with Industry experts.....	12
3. Reference implementation.....	17
3.1 Web Service interface design.....	19
3.1.1 XLIFF.....	19
3.1.2 Non standard schemas.....	24
3.2 Web Service interfaces.....	26
3.2.1 Archiving.....	26
3.2.2 Leveraging.....	26
3.2.3 Machine Translation	28
3.2.4 Parsing.....	29
3.2.5 Pseudo translating.....	30
3.2.6 QA checking.....	31
3.2.7 Reporting.....	32
3.2.8 Language review.....	32
3.2.9 Target generation.....	33
3.2.10 Testing.....	34
3.2.11 Translation.....	35
3.2.12 Updating.....	36
4. Evaluation.....	36
4.1 Standard support	37
4.2 Software localisation workflows.....	37
4.2.1 Translation Workflow.....	38
4.2.2 Machine Translation Workflow.....	39
4.3 Performance.....	40
4.4 Security considerations.....	41
4.4.1 Confidentiality.....	42
4.4.2 Data integrity.....	42
4.4.3 Authentication.....	43
4.4.4 Authorization.....	43
4.4.5 Non repudiation.....	43
5. Conclusion and future work.....	43
References.....	46
Appendix 1: Questionnaire used for the interviews with Industry Experts.....	49
Appendix 2: Interfaces for Software Localisation Web Services	50
1. Archiving Service.....	50
1.1 Archiving.wsdl.....	50
2. Leveraging Service.....	51
2.1 Leveraging.wsdl.....	51
2.2 LeveragingXliff.xsd.....	54
3. Machine Translation Service.....	56
3.1 MachineTranslation.wsdl.....	56
3.2 Machine TranslationXliff.xsd.....	58
4. Parsing Service.....	60
4.1 Parsing.wsdl.....	60

5. Pseudo Translation Service.....	62
5.1 PseudoTranslation.wsdl.....	62
5.2 PseudoTranslation.xsd.....	64
6. QaCheck Service.....	66
6.1 QaCheck.wsdl.....	66
6.2 QaXliff.xsd.....	68
7. Reporting Service.....	70
7.1 Reporting.wsdl.....	70
8. Review Service.....	71
8.1 Review.wsdl.....	71
8.2 ReviewXliff.xsd.....	73
9. Target Generation Service.....	75
9.1 TargetGeneration.wsdl.....	75
9.2 TargetGenerationXliff.xsd.....	77
10. Testing Service.....	80
10.1 Testing.wsdl.....	80
11. Translation Service.....	81
11.1 Translation.wsdl.....	81
11.2 TranslationXliff.xsd.....	83
12. Update Service.....	85
12.1 Update.wsdl.....	85
Appendix 3: BPEL scripts for Software Localisation Workflows.....	87
1 MtWorkflow.bpel.....	87
2 TranslationWorkflow.bpel.....	89
Appendix 4: Is XLIFF in the right place?.....	93
Appendix 5: Extract from XLIFF v1.2.....	101
Appendix 6: Performance measurements.....	103

Index of Tables

Table 1: ISV localisation tools.....	1
Table 2: SOA support in software Localisation tools.....	10
Table 3: Software localisation Web Services.....	16
Table 4: Software localisation Web Services implementation.....	18
Table 5: Leveraging Web Service performance measurements.....	41

1. Introduction

Internationalisation is the process of coding and testing software to ensure that it can both function properly in international computing environments, and be localized in an efficient and cost effective manner. Localisation is the process of adapting a piece of software to a different culture which includes primarily the translation of the user interface. Back in the 1980s, the support for internationalisation and localisation provided by Operating Systems and programming languages was extremely limited. Internationalisation issues such as locale information representation or character set support were not standardized, localisation service providers (LSP) were just appearing and there was a general lack of expertise in these areas.

One central problem for software localisation was the lack of standard file formats designed for localisation purposes. This meant that, in many cases, translators had to work directly with the source files containing the developers code. Some companies came up with rudimentary and proprietary file formats where they would place localisable strings. In either case the choice of translation tools was normally limited to standard text editors with no specific localisation support.

The lack of standard localisation file formats plus a localisation market that was very immature made development of commercial software localisation tools impossible. As a result, in the early 1990s, most large Independent Software Vendors (ISVs) started developing in-house tools to support their localisation efforts (see Table 1):

Company	Tool
Microsoft	Espresso
Oracle	Hyperhub
Symantec	Pebbles
Lotus	Red
Novell	Nile
Corel	Trinity

Table 1: ISV localisation tools

Eventually the success of Microsoft Windows provided a widely used platform with a public standard file format: Common Object File Format (COFF), used for Win32 applications. This provided a fixed target that allowed development of commercial localisation tools. In some cases

these initial tools were developed by Localisation Service Providers (LSP) as a way to obtain a competitive advantage over other LSPs.

It's interesting to note that one of the current leading commercial software localisation tools, Catalyst (www.alchemysoftware.ie), started originally as an in-house tool used by Corel. It is also interesting how many large ISV still use in-house tools for most if not all their software localisation efforts. In many cases these tools are evolutions of the ones first developed in the early 1990s. There are several reasons for this but one of them is that these tools are tightly tied to the ISV's internal localisation workflow.

Currently there are two broad camps when it comes to software localisation tools: in-house proprietary tools used by large ISVs and a number of commercially available tools that are used by small and medium size ISVs. Commercially available software localisation tools include Catalyst, SDL Passolo (www.sdl.com/en/sites/sdl-passolo), RC-Wintrans (www.schaudin.com), etc.

The software localisation tools available today are, by and large, implemented as desktop applications. The leading ones, both in-house and commercial ones, have been in the market for well over 10 years. During this time they have expanded their functionality, naturally trying to add more value. As a result, they are function rich applications that target a variety of users including linguists, localisation engineers and localisation project managers. This broad user base makes numerous and varied functional requirements from these tools. This is reflected in the types of functionality these tools offer, which typically include:

- File parsing: parse the supplied files looking for localisable content.
- Leveraging: use a translation memory to pre-translate.
- Updating: use the previous localised version of the software to pre-translate.
- QA checking: run automatic QA checks on the localised content.
- Generation of localized versions of the original files.
- Reporting: generate statistics on the supplied content.
- Translation environment for localisable content.
- Linguistic review: assess and rate the quality of the translation.
- Pseudo translation: generate a pseudo translated version of the localisable content.

Normally these tools are designed as one-stop solutions. Once you choose a given tool the expectation is that it will be used for all the functions it provides. In fact, each of these tools use proprietary file formats for their project serialization. Porting project data between these proprietary file formats is normally not possible without an unacceptable loss of project metadata. The exported content generally consists of just the list of source strings and their translations missing other associated data such as the source string unique identifier, coordinates and size, translation status, available fuzzy matches, notes, etc. So, once the choice of tool is made, it is quite difficult if not impossible to use a different tool even though they may have a superior implementation of some of the functions involved in software localisation. This tool lock-in normally implies other lock-ins:

- A certain degree of LSP lock-in for whatever the unit of work (normally a project-language pair). It is possible to coordinate the transfer of files between LSPs but this adds overhead as well as increasing the risk of file handling errors. It also assumes that all vendors are proficient with the tool, which may be true for the commercially available ones but not necessarily for the in-house developed tools.
- Platform lock-in. Most of the software localisation tools are not cross platform. A choice of tool implies a choice of the operating system where localisation will happen. Even though Windows is the prevalent OS in the market, when localizing software targeted to other platforms (e.g. Linux, MacOS) it may be useful to perform some of the localisation functions in the target platform.

In an attempt to address the problems arising from such lock-in, this project investigates what advantages, if any, a Service Oriented Architecture (SOA) can bring in the context of software localisation. Erl in [12] explores the design principles for specifying Web Services for a SOA: standardized contract, reusability, autonomy, statelessness, discoverability and composability. According to Erl [1] a SOA design based on those principles would achieve the following goals:

- Business driven: the technical architecture should be aligned with the organization's business direction and requirements
- Vendor neutral: the software system should allow for the replacement or recombination of vendor technologies in order to maximize evolving business requirements.
- Enterprise centric: the architecture should be designed assuming that the services will be shared by other parts of the enterprise or will be reused in larger compositions.

- Composition centric: the architecture should be able to accommodate change through the “agile assembly of service compositions”[1].

There are some clear potential benefits of using an SOA design over the model currently used for software localisation. These include:

- Lessened vendor and tool lock-in which allows a choice of best of breed solutions for each function.
- Increased interoperability between localisation tools.
- A more flexible and agile architecture. Through the use of services as components ISVs and LSP can react more quickly to new business demands.
- Increased automation: reduction in process management overhead through the use of automated workflows.

This research project aims to evaluate whether these objectives are realistic. To do this the project first identifies cohesive pieces of functionality, involved in software localisation, that can be better provided as web services. For the set of software localisation functions identified as useful components, SOAP based Web Service interfaces have been developed. These interfaces have been used to build a reference implementation. The reference implementation includes the set of WSDL file as well as a number of BPEL workflows that are used to showcase how a system such as this could function. This reference implementation is used to assess the advantages and disadvantages of such as system compared with the model presently used in the marketplace.

This approach provides practical experiences in an area with few, if any, existing implementations. Also, the design of the Web Service interfaces provides lessons that can be of interest to future standardisation attempts. Some of the choices could be different though. The choice of Web Service technology is one example. SOAP was used but REST could be as good if not a better fit for localisation. Another example is the level of localisation functionality included in the reference implementation. This was kept to a minimum and mock functionality was used instead. The intention was to focus on how the interfaces work. But performance factors can have an effect on the design of an interface and those factors become more apparent once a more realistic functionality is in place. Having chosen a more comprehensive development of localisation functions may have revealed additional information.

This dissertation is structured as follows. Section 2 presents the current state of the art for software localisation Web Services. Section 3 describes the reference implementation giving details

on the the different Web Service interfaces, the packages implementing those interfaces and the localisation workflows. Section 4 explains the results of the evaluation and section 5 offers a conclusion and points to future work in this area.

This research will be presented in the 15th Annual Internationalisation and Localisation Conference organised by the Localisation Research Centre (LRC) in September 2010.

2. State of the art

This section presents an overall picture on how and where Web Services are used in software localisation today. It describes both the prior research in this area as well as research conducted for this dissertation. It also presents any relevant standards that can be used to support SOA in the context of software localisation. The following areas are covered:

- Section 2.1 covers the little research I have found on the topic of Web Services applied to software localisation
- There is a long list of Web Services specifications generally referred to as the WS-*. Section 2.2 describes the ones relating to localisation such as Trans-WS.
- The software localisation industry has produced a number of standards aimed at increasing tool interoperability such as XLIFF or TMX. Section 2.3 gives an overview of the ones relevant to this project.
- Section 2.4 covers the current software localisation tools. Who they are and how much support they provide for a Service Oriented Architecture.
- Because of the lack of substantial academic research in this area, or publicly available industry analysis, I decided to conduct a number of interviews with industry experts to more fully capture current requirements and expectations for SOA in the industry. Section 2.5 presents a summary of those discussions.

2.1 Related work

Lewis et al. in [2] propose an Open Service Framework for Next Generation Localisation. The aim of the framework is to establish a Unified Localisation Factory that would allow for easy development of localisation solutions based on the integration and orchestration of Web Services. The framework is a work in progress encompassing a set of core principles, a process map, a set of methods and techniques, a system services architecture and a set of reusable elements. The framework proposed by Lewis et al. is at a higher level of abstraction than the work presented in

this dissertation. But it is complementary and provides an conceptual umbrella for our focus on software localisation Web Services.

In [3] Lewis et al. explore the benefits of using web service integration and orchestration in the context of localisation. Specifically they investigate how best to use them to integrate natural language processing functionality, such as text analytics and machine translation (MT), into localisation workflows. The authors implemented Web Services for MT, text analytics and crowd-sourcing and used BPEL to orchestrate sample localisation workflows. According to the authors the main challenges facing such a solution are the maturity of BPEL and the design of the Web Service interfaces.

2.2 Web services standards

There are many standardised specifications associated with Web Services. They are referred to collectively as WS-* even though they are not necessarily a coherent group. They are maintained by different standard bodies (the main ones being the W3C and OASIS), they are at different stages of development from draft to ratified and , in some cases, they are in competition with each other.

Web Services specifications build on a number of XML specifications such as XML Schema, XML Namespaces or XPath. There are Web Services specifications for a wide range of areas such as:

- Messaging: SOAP, WS-Notification, WS-Addressing, etc.
- Metadata Exchange: WS-Policy, Universal Description, Discovery and Integration (UDDI), Web Service Description Language (WSDL), etc.
- Security: WS-Security, WS-Trust, WS-Federation, Security Assertion Markup Language (SAML), etc.
- Reliable Messaging: WS-Reliability, etc.
- Interoperability: WS-I Basic Profile, etc.
- Business process: WS-BPEL, WS-Choreography, etc.
- Transaction: WS-Transaction, etc.

The only Web Service specifications directly related to localisation are the ones maintained by the Translation-WS (TWS) OASIS technical committee. TWS was set up in 2002 “to create a standard way for Web services to be used within the translation and localisation industry” [5]. In

2007 it released the OASIS Translation Web Services 1.0.3 draft specification [6]. This specification is far less ambitious than the initial goal. It focuses on automating the communication between localisation content creators and localisation service providers (LSP) through Web Services. It specifies a simple interface for submitting and retrieving localisation jobs between the parties whose primary aim is to facilitate the outsourcing of localisation work to the LSPs. The interface only provides visibility in regards to the project status but does not address how to expose localisation functions as Web Services that can be dynamically integrated into different workflows. The specification treats an LSP as a black box and assumes the client is looking for a one stop solution and is only interested in access to a quote and some project status information. This is certainly one real scenario but not the only possible one. It is nevertheless the only scenario contemplated in the specification. As such, even though useful, the specification is far too limited to allow for disruptive and innovative changes in the localisation marketplace.

It is not clear how much this specification is being used in the industry. Lionbridge, the largest LSP in the market, has implemented it for their Elcano web portal (<http://elcano.Lionbridge.com>) in conjunction with XLIFF. Reynolds in [5] explains how it was used to automate the localisation process for a large customer.

2.3 Localisation Standards

There are a number of standards in the area of localisation that have been developed to support interoperability between localisation tools. The Organization for the Advancement of Structured Information Standards (OASIS – www.oasisopen.org) has a number of them including:

- XML Localisation Interchange File Format (XLIFF) [8] is an XML based file format first standardized in 2002. It is used to represent localisable content and their corresponding translations. It can also be used to store other related information such as process information, possible translations, context information or binary data. Most localisation tools implement some limited support for XLIFF. Its current specification is v1.2 released on February 1st 2008.
- Open Architecture for XML Authoring and Localisation (OAXAL) is a recently set up technical committee exploring the integration of several localisation standards (XLIFF, TMX, SRX, etc.) into an environment for document creation and localisation. In December 2009 it released v1.0 of the OAXAL Reference Model[22].

The Localisation Industry Standards Association (LISA) has also developed a number of localisation standards including:

- Translation Memory eXchange (TMX) [9] is an XML standard first released in 1998. It can be used to exchange translation memory data among Computer Aided Translation (CAT) and localisation tools. Support for TMX is quite common among localisation tools. The current official version is v1.4b released on October 2004.
- Term Based eXchange (TBX) [10] is another XML standard that is used to exchange terminology data among localisation tools.
- Segmentation Rules eXchange [11] is a standard for describing the rules used to segment text for processing by a Translation Memory System (TMS). It's a companion standard to TMX as it allows the owner of a Translation Memory (TM) to reproduce the leveraging results obtained with the TMS that first performed the segmentation. It has been sparsely implemented as TMS vendors regard segmentation rules as part of their competitive advantage.
- Global information management Metrics eXchange (GMX): family of standards on globalization and localisation related metrics.

The W3C, which maintains some of the core specifications for Web Services such as SOAP or WSDL, also has an Internationalisation Activity (<http://www.w3.org/International/>) that works on making possible to use Web technologies with different languages, scripts and cultures. This includes, among others standards:

- Internationalisation Tag Set [4] used to support the internationalisation of XML Schemas.

2.4 Localisation tools

There are many Computer Assisted Translation (CAT) tools available. Searching through the most common localisation forums revealed close to a hundred and there are probably some extra ones. As can be expected these tools are a very mixed group. They include translation memory managers, terminology managers, spell/grammar checkers, project management software, etc. Some are freeware while others are commercial offerings and their development model can be open source or close source.

Another divide is the one separating CAT tools that focus on software from the ones specializing in documentation and help. Software and documentation localisation present different challenges in term of volumes and the different metadata associated with each content type. This results in tools that specialize on one type of content or the other. This project focuses on software localisation and so the evaluation is centred on CAT tools for software content.

There is no reliable data on what the market share is for these tools. Localisation tools vendors make claims about their customer base but there aren't neutral studies to verify those claims. The opinions of industry experts have been used to select the tools that are regarded as more widely used. The set of tools being evaluated is not a comprehensive list of the software CAT tools available but it is representative of the current state of the art.

This research tries to evaluate whether the current CAT tools can be used to implement a Service Oriented Architecture. To do it each tool has been rated using the following criteria:

- Automation: do they expose an API that can be used by other software applications?
- Network access: do they exposed an API through a network interface?
- Expandability: do they expose a plug-in API that can be used to expand their functionality? What areas can be expanded?
- Platform independent interface: can their API be called from programs developed in different programming languages/operating systems?
- Standard support: do they support standard file formats such as TMX, TBX, XLIFF?
- Interoperability: can they interoperate with other software localisation tools?
- Composability: can different localisation functions (e.g. leveraging, pseudo translation, QA) be accessed individually?

See the results on Table 2.

Name	Automation	Network Access	Expandability	Platform Ind. Interface	Standards support	Interoperability	Composability
Alchemy Catalyst www.alchemysoftware.ie	Yes; supports scripting & Event API	No	Yes, core operations through Event API	No	Limited; TMX, XLIFF	No	Limited
SDL Passolo www.sdl.com	Limited; supports scripting	No	Limited; through scripting	No	Limited; TMX, XLIFF	No	Limited
RC-Wintrans www.schaudin.com	Yes; supports scripting & COM	No	Limited; through VBA macros	No	Limited; TMX	No	Limited
Multilizer www.multilizer.com	Limited; command line support	No	Yes; Delphi/C++ SDK	No	Limited; TMX, XLIFF	No	Limited
Sisulizer www.sisulizer.com	Limited; command line support	No	No	No	Limited; TMX, XLIFF	No	Limited
Visual Localizer www.visloc.com	Limited; supports scripting	No	No	No	Limited; TMX	No	Limited
Lingobit Localizer www.lingobit.com	Limited; command line support	No	No	No	No	No	Limited
Heartsome www.heartsome.net/EN/xlfedit.html	No	No	No	No	Yes; XLIFF, TMX, TBX, SRX	No	Limited
VB Language Manager Pro www.whippleware.com/vblm.htm	No	No	No	No	No	No	No

Table 2: SOA support in software Localisation tools

As can be seen from Table 1 most of the tools do not expose an API. Support for automation is limited to scripting or just command line parameters. This rules out expanding their functionality, access through a network or a platform independent interface. An interesting area is standard support. Most tools allow for import/export of translation memories using TMX. But TMX is not equipped to represent all the metadata associated with software localisation (unique ids, resource type, coordinate information, etc.). Exporting a software localisation project to a TMX file can result in an extensive loss of metadata. As a result, though useful in some scenarios, support for TMX does not offer a route to interoperate with other localisation tools. XLIFF on the other hand it is better tailored to represent software resources. It could potentially be used to transfer a project repository from one tool to another. It is interesting to note that, even though most tools handle XLIFF as a source file that can be added to a project, none offer the possibility to export a project repository as an XLIFF file.

Composability is achieved only in limited fashion. In so far as the tools can be scripted they can be asked to perform different tasks on the files. But this is only supported while using the native, proprietary file format used by each tool. The implication is that, prior to perform any task, the project would need to be recreated for each tool -repeating steps such as parsing already performed. It would involve generating the localised content from one tool and re-parsing the source files and aligning them with their localised versions. All this, as explained earlier, without good support for interchange file formats. Apart from the technical risks just the overhead would be quite considerable.

Even though our focus is on software localisation is interesting to note that the landscape for documentation CAT tools seems to be more dynamic. To traditional desktop offerings such as SDL Trados there are new products using cloud computing models. Examples of products offered as Software-as-a-Service (SaaS) include Lionbridge's Translation Workspace, Google Translators Toolkit or XML-Intl's XTM Suite. There may be several reasons for this situation. Because of the higher volumes, documentation costs make the lion share of the ISVs localisation budgets. This may lead companies to invest more heavily on documentation CAT tools than their software counterparts creating a more profitable market for localisation tools providers. Another factor may be the relative simplicity of documentation compared to software both in relation to content (documentation hasn't got the myriad of input file formats software has) and process (documentation doesn't require a functional testing/fixing cycle).

2.5 Interviews with Industry experts

The lack of extensive research material lead me to perform a number of interviews with people with significant experience in the localisation industry. Although the list of interviewees is small they represent the major players in the industry. These are localisation service providers (LSPs), localisation tools providers (LTPs) and their customer base. It should be noted that the difference between localisation service providers and localisation tools providers is being blurred as more and more tools vendors are being acquired by LSPs. The customer base for software localization services is expanding. More and more companies are developing software as more and more products nowadays include microprocessors. But still the reference customers for localisation services are ISVs.

The people interviewed are:

- Stephen Holmes, Director Technical Operations, Sajan Software Ltd (21-03-10)
- Daniel McGowan, Group Project Manager, Novell (26-03-10)
- Paul Leahy, Senior Director of Infrastructure and Engineering, Worldwide Product Translation Group, Oracle (28-03-10)
- Phil Ritchie, Chief Technology Officer and Paul Magee Senior Software Architect, VistaTEC (31-03-10)
- Fred Hollowood, Director Research and Deployment and Robert Leyden, Senior Manager, Regional Development Group, Symantec (03-06-10)
- Enda McDonnell, Director of Engineering, Alchemy Software, (11-06-10)

The questionnaire listed in Appendix 1 was sent in advance and was used as a basis for the conversation. The interviews were recorded. At a general level the following five areas were covered:

a) Current adoption of Web Services for software localisation

The level of adoption of Web Services as part of software localisation is mixed. A large ISV such as Oracle uses Web Services pervasively through their localisation cycle. The translation group (WPTG) offers Web Services interfaces to their internal customers within Oracle. These interfaces are not exposed to third parties though, and neither the translation group consumes third party Web Services.

Novell does not use Web Services technology neither internally nor with external third parties. They do expose a number of resources through more conventional web portals.

Symantec uses an internally developed workflow engine (Orbit) to drive automation throughout the localisation process. The engine makes calls to functional components that provide updating, leveraging, build, translation, etc. Many of the functions exposed and consumed by Orbit already use Web Service interfaces and this is seen as the better technical choice. For example, the interface exposed to the build teams in the development groups to submit localisation requests uses Web Services. Currently Orbit is kept exclusively in-house and does not interact with any external Web Service.

The situation among the LSPs is also mixed. Sajan offers cloud based localisation services using their Global Communication Management System. This is a proprietary web based localisation solution.

VistaTEC is experimenting with Web Services in different ways. As consumers they sometimes use a third party Machine Translation engine that is offered through Web Services. They also use REST based Web Services with one of their clients, Google -these interfaces are used to subscribe to and publish data to VistaTEC localisation team via the cloud. Plus they have developed a workflow engine that exposes SOAP based Web Services not just internally but to other LSPs. This targets multi-vendor situations where a client is using VistaTEC for some functions (e.g. language review) while using a different LSP for others (e.g. translation). They want to allow other LSP to update status information so that the appropriate actions can be triggered by the workflow engine. This type of scenario is one that can easily be seen to benefit from the standardisation of the Web Service interfaces.

Enda McDonnell mentioned that Alchemy is following closely the demand for localisation Web Services. They already offer a product, Language Exchange, that offers Translation Memory services through a Web Services interface. They are ready to expand the Web Services support in their tools as demand expands but balancing it with continued support for their desktop applications.

Overall there seems to be quite a bit of interest in Web Services and, even though the usage seems to be low, there is a high level of attention to the subject.

b) Advantages of SOA for localisation

There are some advantages that were mentioned in most if not all the interviews. Scalability probably ranked at the top. As the workloads get bigger manual interfaces become bottlenecks. A

Service Oriented Architecture's ability to automate those interfaces is crucial. This is a concern shared by both ISVs and LSPs. For an LSP it equips them better to cope with unpredictable workloads coming from clients. It was pointed out that this unpredictability is becoming more acute as more customers start using agile development methods.

For a large ISV such as Oracle, where the translation department acts in a role similar to that of a Multi Language Vendor (MLV), it allows them to push activities such as internationalisation validation back to their internal clients. Oracle uses Web Services to offer localisation compliance technologies (Self-Validation, Self-Pseudo translation, etc.) to the different internal development groups. Scalable compliance services have been particularly necessary given the large amount of companies acquired by Oracle in recent times.

SOA scalability is based on automated interfaces. Interestingly, Robert Leyden pointed out that Symantec's main focus is not on localisation tools but on automation. Symantec's experience is that the cost benefits of a fully automated workflow are larger than the ones obtained through improving the functionality provided by localisation tools that require manual interactions. Web Services is only one way a component can present an interface to an automation engine. Some of the components accessed by Symantec's Orbit offer command line or scripting interfaces. But these are viewed as technically inferior solutions normally belonging to legacy components.

Another advantage mentioned often and one that comes hand in hand with scalability is availability. Functions offered through Web Services are available 24x7. Even the ones that require human processing at least guarantee that the request has been queued. Increased availability is becoming more important as both companies and LSP have geographically distributed workforces living in different time zones and with different bank holidays. Scalability and availability are the basis for the type of enterprise solution LSPs and ISV want.

'Pay as you use' models were also mentioned as an advantage, pointing out that the standard 'license' model for software localisation tools is sometimes overkill when one plans to use just a portion of the tools functionality.

Composability, the ability to easily choose which components (services) to use in different scenarios, was also a common point. It was referred as the flexibility to “rip and replace” or “swap things in and out”.

Other advantages that were also pointed out included:

- Easy deployment and version consistency. Because software is deployed centrally it is easier to ensure that translators, localisation engineers and other users are using the re-

quired version.

- A point was made that ISVs can use localisation Web Services to manage costs at a more granular level based on the ability to choose providers of specific localisation services (such as MT, leveraging, translation, etc.). This may not be of interest for clients looking for one stop solutions.
- It was pointed out that having automatic workflows guarantees that certain activities, such as QA checks, will happen and their execution can be accurately monitored. Manually driven workflows do not offer the same level of certainty.
- IT departments in the ISV may see Web Services as a more acceptable mechanism to allow LSPs access to internal systems because it is more secure and it is more formalized than ad-hoc ones.
- It better supports the business case for specialist companies that deliver a 'niche' service as a Web Service.

c) Barriers to adoption

Both Daniel McGowan and Paul Leahy stressed the need for open standards. A solution based on proprietary interfaces, even if they are public, was not regarded as viable. Most ISVs use many LSPs and need to be able to reuse their tools solutions across them. The need for open standards makes a move to an SOA an industry decision, one that needs the support from most if not all the players. It was noted the difficulty involved in building this kind of consensus.

Related to the above it was mentioned a possible resistance from localisation tools vendors. According to Daniel McGowan an attempt to develop a Web Service interface, in conjunction with Moravia, for Idiom's WorldServer was rejected by Idiom. The interface would allow Novell's and Moravia's instances of WorldServer exchange files. But Idiom “felt that they were losing in their ability to make money out of the tool” and made changes to hide some previously exposed API.

Performance was also mentioned by most as a must for a system such as this to get widespread adoption. It was pointed out that attempts in the past to offer software that relayed on Web Services have failed because of this issue. This was blamed on the fact that often distributed software is still designed as if it was a desktop application without regard for network latency; and most of these systems are tested in a LAN but later used in a WAN. It was suggested that simple design practices such as caching, intelligent lookups using background threads and plain avoid-

ing synchronous calls can quickly improve the user experience.

Fred Hollowood and Robert Leyden stressed that, for a company such as Symantec to access a third party Web Service, it would require guarantees about the expected level of service as regard to turnaround time. This connects to the general subject of Service Level Agreements negotiation and monitoring.

Reliability and robustness were also a common theme. The need to handle failure gracefully was noted, specially if a services architecture is to compete with a manual process that, slow as it may be, it is very resilient (just make a phone call and ask for the files to be re-sent).

Interestingly security was also mentioned as a barrier. It was regarded as such because of the need to get buy in from the IT departments at both ISVs and LSPs.

Questions were raised about the maturity of workflow engines specifically BPEL and the maturity of the user interfaces built upon it. Daniel McGowan pointed out that workflows are normally administered by localisation project managers. Some of these are not highly technical (they rely on the localisation engineer for technical matters). If the system is aiming to provide an easy way to customize a workflow it should provide a simple to use user interface.

d) Web Services design

In general the list of proposed Web Services presented in the questionnaire was considered sensible. Suggestions for extra ones included a segmentation service and a service to auction small localisation jobs.

There was a consensus that, though not perfect, XLIFF is the sensible candidate to be the XML based format used as the payload for localisation Web Services.

It was noted that interfaces that required less contextual information, such as QA, would perform better and offer more value rather than the ones, such as updating, where additional metadata (previous version, etc.) is required. It was also mentioned that synchronous services that perform automatic tasks could be more easily integrated into existing localisation workflows. While asynchronous interfaces where human intervention is needed, such as translation would add less value. Based on these criteria the list of proposed localisation Web Services (see Appendix 1) can be prioritized as follows:

Table 3: Software localisation Web Services

- | |
|--|
| 1. Machine Translating: use machine translation to pre-translate |
|--|

2. QA checking: run automatic QA checks on the localised content
3. Reporting: generate statistics on the supplied content
4. File Parsing: parse the supplied files looking for localisable content.
5. Leveraging: use a translation memory to pre-translate
6. Segmentation: break the source string into segments for more effective leveraging
7. Pseudo translating: generate a pseudo translated version of the localisable content
8. Updating: use the previous localised version of the software to pre-translate
9. Generate localised file: generate a localised version of the original file
10. Archiving: permanent storage for finished localised content
11. Artwork/Multimedia processing: localise artwork/multimedia content
12. Translating: translate the localisable content
13. Language reviewing: assess and rate the quality of the translation
14. Testing: test for defects in the localised product

e) Non functional requirements

Across the non functional requirements that were listed in the questionnaire the only one that was generally agreed on was security. Most people opinion was that the Web Services interfaces should enforce security at a low level. This means that encryption and integrity protection should be mandatory. But it was also felt that higher level security, as in roles, users and access restrictions, should be left to the parties to decide outside the specification.

3. Reference implementation

A reference implementation has been used to evaluate the advantages and disadvantages of using a Service Oriented Architecture in the context of software localisation.

The first task was to identify cohesive pieces of functionality, involved in software localisation, that could be better provided as web services. As presented in the previous section, the resulting list was discussed in conversations with industry experts and a prioritised list was drawn (see Table 2 in section 2.5). The list by no means tries to be exhaustive but it covers the main software localisation functions. For each function a Simple Object Access Protocol (SOAP) based Web Service interface has been designed (see section 3.2).

Implementations for these interfaces have been developed using Java version 6. These Java packages fulfil the interfaces but are not fully functional localisation components. In some cases they provide partial functionality; e.g. for the Parsing Service a parser for one file type (Java properties [24]) was developed. In other cases the Java packages are simply stubs; e.g. the Leveraging Service does populate the content but it does not implement any string matching algorithm.

The reference implementation includes the minimum localisation functionality required for a system such as this. This minimum includes a) handling at least one input data type and b) outputting the corresponding localised version. This led to the partial implementations for the Parsing and Target Generation Services. The focus for the other services was on how to represent the (mock) results of their processing in a way that could be returned efficiently to the requesting process. Table 4 lists the Services that were developed and their implementation level:

Table 4: Software localisation Web Services implementation

Service	Implementation
File Parsing	Implemented a parser for Java Properties files. Full implementation of the guidelines from the Representation Guide for Java Resource Bundles[24].
Leveraging	Implemented operations for context & context-less leveraging. Traverses XLIFF doc and inserts mock <alt-trans> elements.
Machine Translation	Traverses XLIFF doc and inserts mock <target> elements.
Target Generation	Implemented for Java Properties files.
QA Checking	Traverses XLIFF doc and inserts mock <qaisue> elements.
Reporting	Returns an XML document containing a few high level statistics.
Pseudo Translation	Traverses XLIFF doc and inserts <target> elements populated with pseudo translated versions of the source strings.
Translation	Implements operation for submitting and retrieving translations. Traverses the XLIFF doc and inserts <target> elements with mock translations.
Language Review	Implements operation for submitting and retrieving reviews. Traverses the XLIFF doc and inserts <note> elements with mock comments.

Testing	Implements an operation to receive a file and serialize it.
Archiving	Implements an operation to receive a file and serialize it.

Two workflows have been developed representing typical localisation workflow segments. These are used to showcase how these proposed services can be orchestrated to fulfil these workflows.

The Web Services and the workflows were deployed on servers running on the Trinity College network and a number of client applications were developed. These clients consume the Web Services and initiate the workflows and can be used to simulate how users would interact with such a system.

The Web Service interfaces and the reference implementation are included in the companion DVD annexed to the dissertation. Check the Readme.txt in the DVD for a description of the contents.

3.1 Web Service interface design

Standards compliance was one of the main principles used during the design of the Web Service interfaces. The goal was to leverage the efforts already made in the localisation area so to minimize the work required by a future standardisation process. The attempt to re-use existing standards was balanced with the need to provide non-ambiguous interfaces.

3.1.1 XLIFF

During the design of the Web Services used in the reference implementation there was a recurrent need for a schema defining localisation content. XLIFF (see section 2.2) was the obvious candidate. It is a relatively mature standard that has small but significant market penetration and tool support. [For a discussion on the role of XLIFF in the localisation industry see appendix 4 for a draft on an article that will be submitted to a localisation magazine to be considered for publication.]

In the reference implementation, most of the request and response documents used by the Web Services include XLIFF documents. They are used both as inputs to Web Service operations and as mechanisms to convey the output of those operations. This extensive application of XLIFF in the reference implementation has revealed a number of issues with the standard that can be categorized as follows:

a) Lack of definition

The transformations into XLIFF are undefined for most input file formats. Only three Representation Guides are provided by the OASIS XLIFF TC. These are:

- Representation Guide for HTML [25]
- Representation Guide for Java Resource Bundles [24]
- Representation Guide for Gettext PO [26]

There are many input file formats not covered by these. In fact, the XLIFF standard contains 53 pre-defined values for the datatype attribute of the file element (see appendix 5). The representation guides only cover 6. The list of input file formats without a representation guide is not made of niche formats but includes some of the most common localisation formats such as win32 or resx.

For the reference implementation we decided to develop a parser for a file format covered in one of the representation guides: javapropertyresourcebundle. Java .properties files are text files containing key-value pairs. There is a public file specification for them that is part of the Java Development Kit documentation. Java .properties files can represent only one resource type, namely a string-table, and are among the simplest input file formats in localisation.

The representation guide still needs to define guidelines for many aspects of the transformation. These are some examples:

- Structure of the resulting XLIFF file
 - Each .properties file is represented in a separate `<file>` element
 - The `translate` attribute of `<trans-unit>` element should be set to `no` when the value string is empty
- Encoding and character representation
 - Single quotes, double quotes, backslashes should use unescaped versions in the XLIFF file
 - Characters encoded using the Java Unicode notation (`\uXXXX`) should be replaced with their Unicode equivalents
- Order of Extraction
 - The extraction order should reflect the order of the data in the source document

- Key identifier
 - The identifier (used for leveraging, etc) is stored in the `resname` attribute. While the required `id` attribute is set to a sequential value.
- Parameter specifiers (message replaceable)
 - Variables and operators found in the strings should be encapsulated using either `<ph>` or `<x/>` elements

Even for a simple file format such as this the level of detail laid in the representation guide is quite formidable. Without it the possibilities for incompatible transformations of Properties files into XLIFF are very high. And those incompatibilities have a knock on effect on other Web Services. For example, different ways to represent the source string in XLIFF could cause a reduction in the number of matches found by a Leveraging Web Service with the accompanying increase in localisation costs.

The lack of representation guides is obviously a problem for a File Parsing Web Service, but it is also a problem for a Target Generation Web Service. Some of the changes that were needed to transform an input file into XLIFF need to be reversed by the Target Generation service. Uncertainty about which and how changes were made make target generation difficult and error prone.

b) Lack of coverage

During the design of the localisation Web Services listed in Table 2 decisions had to be made on the best way to return the information resulting from the processing performed by the Web Services. These decisions were based on a number of factors the primary one being the availability of standards. As explained earlier, if an applicable standard was available it was chosen, to minimize the need for future standardisation efforts.

In the cases of the Archiving or Testing Web Service the information is simply an acknowledgement of the request so it is returned in the form of an information message.

The other Web Services need to return more complex information and, in most cases, it was found that XLIFF provided a mechanism to convey it:

- The File Parsing Service returns XLIFF files representing the source files
- The Updating Service returns XLIFF files after being updated with the previous version
- The Leveraging Service inserts `<alt-trans>` elements (defined in the XLIFF schema) in the input XLIFF file and returns it.

- The Machine Translation, Translation and Pseudo-Translation services insert <target> elements (defined in the XLIFF schema) in the input XLIFF file and returns it.

But there are some cases in which XLIFF does not provide such mechanism. This is so for the QA and Reporting Web Service. The QA Service builds a list of potential issues affecting the localised content. For each one it identifies the <trans-unit> where the issue originated, the type of issue and other metadata. The Reporting Web Service builds a report containing statistical information: total wordcount, untranslated wordcount, number of files, number of dialogs, etc.

In these cases the following options were considered:

- Placing the information in a separate XML document defined in its own schema.
- Use existing, generic XLIFF elements such as <note>. The content of a <note> is text. Standards based on textual annotations are error prone and lack formal structure when compared with an XML Schema.
- Define new schema types and use the extension mechanisms in XLIFF to insert them into the XLIFF document

In the case of the QA Web Service the option to create a separate XML document required either:

- to place information to correlate each issue to the corresponding translation unit and send the input XLIFF file along with the document
- to duplicate large parts of the input XLIFF file

This created a great deal of processing overhead. Instead a new schema type was designed and, using XLIFF extension mechanism, instances are inserted in the <trans-unit> element. The schema type is:

```
<xs:complexType name="qaIssueType">
  <xs:attribute name="type" type="xs:string" use="required"/>
  <xs:attribute name="resolved" type="xlf:AttrType_YesNo" use="required"/>
  <xs:attribute name="severity" type="xs:string" use="optional"/>
  <xs:attribute name="defect-id" type="xs:string" use="optional"/>
</xs:complexType>
```

In the case of the Reporting Service the implementation for the reference model simply calculates some basic statistics for the contents of the XLIFF file. The correlation between the report

and the input file is thus quite trivial so it is returned in its own document (see Appendix 2 for the schema definition).

In both cases the solution involved creating non-standard schemas. The question that arises is how best to standardise them: as independent standards or as part of XLIFF. In the reference implementation they are part of the schema definition inside the WSDL file. This way users of the Web Service have access to the schema without the need to reference external sources. It could be argued that efforts to standardise the WSDL file will standardise the new schemas. But it seems inconsistent that Leveraging results can be expressed in XLIFF but QA results can't. The XLIFF standard reflects the industry consensus at the time it was published. Maybe that consensus needs to be widened to make room for the demands of localisation Web Services.

c) Excess flexibility

XLIFF flexibility comes from several sources:

- Structural: most attributes and elements are optional and the standard offers alternative ways to represent the same information. For example the `<trans-unit>` element has 29 attributes of which 28 are optional.
- Extensibility mechanism: XLIFF offers several places in the schema where elements defined in an external schema can be added
- User defined values: most attributes are either undefined or else allow for user defined values on top of the pre-defined ones.

During the development effort for the reference implementation we were regularly confronted with this flexibility. Some examples:

- the Leveraging Web Service needs to determine the target language for the operation. The `<file>` element has a `<target-language>` attribute, but it is optional. If it's not present one can check the `<xml-lang>` attribute for each `<target>` element. But `<target>` elements are also optional and the problem is compounded by the fact that, even though discouraged, the standard allows for multilingual files: “It is strongly recommended that content within the `<file>` element be uniformly bilingual” [8]. So, potentially, all `<target>` need to be checked to build an array of target languages.
- the `<skeleton>` element, needed by the Target Generation Web Service, is optional

Contending with XLIFF flexibility resulted in increased code complexity caused by:

- the need to check for alternative representations
- the error handling code needed where expected -but optional- information is not provided.

It also meant that the mechanisms for automatic schema validation offered by Java through the API for XML Web Services (JAX-WS) are rendered ineffective. If the schema is very flexible, having JAX-WS validating the input data does not eliminate the need for internal validations in the Web Services code.

XLIFF flexibility also increased the complexity of the WSDL interfaces for the localisation Web Services. In some cases the ambiguity in XLIFF forced moving some parameters from inside the XLIFF document and into the request document for the operation. This was the case for the target language for several Web Services such as Leveraging. In other cases additional fault messages had to be introduced to respond to cases where some essential information was missing from the input XLIFF file.

It isn't clear that the benefits obtained from having a flexible standard outweigh the corresponding increase in the effort required to write functional implementations for it.

3.1.2 Non standard schemas

The problems discussed in the previous section come from the attempt to use XLIFF as input to Web Service operations. XLIFF flexibility has an impact on the input handling code as well as increasing the number of fault messages needed in case the XLIFF file does not contain what the service needs.

The alternative approach is to design a schema that fully and exclusively defines the input data required by the Web Service. A schema such as this is non standard with the resulting disadvantages in areas such as tool interoperability. But it brings a number of advantages of its own:

- reduced network traffic: the schema can be defined in a more concise manner, explicitly excluding all non necessary data that is present in XLIFF. A good example are the `<skeleton>/<internal-file>` elements. They can be mandated for services that need them, such as the Target Generation, but excluded for the rest.
- reduced security risk: only the required information is exchanged for each service thus reducing the security risk as well as the level of trust placed on the service provider to the necessary minimum.

A Web Service schema can use XML Schema's import mechanism to 'pick and choose' from XLIFF the elements deemed necessary. As an example, let's consider such a schema for the Pseudo Translation service and what the main differences are in respect to XLIFF:

- the <file> element does not contain a <header> element
- the <trans-unit> element does not contain any of the following: <seg-source>, <target>, <context-group>, <count-group>, <prop-group>, <note>, <alt-trans>

The Pseudo Translation service is interesting because the required input is a small subset of XLIFF. The opposite applies to the Translation service where only two elements present in XLIFF are clearly superfluous:

- the <skeleton> element is used to recreate the source or localised files. This function, performed by the Target Generation service, is not required during Translation
- the <bin-unit> element is used to place artwork or multimedia content. This content is handled differently than text which is the focus of the Translation service.

Consequently the Translation input schema closely resembles XLIFF.

The reference implementation includes the following Web Service specific schemas (see full listings in Appendix 2): LeveragingXliff.xsd, MachineTranslationXliff.xsd, PseudoTranslationXliff.xsd, QaXliff.xsd, ReviewXliff.xsd, TargetGenerationXliff.xsd, TranslationXliff.xsd.

There are no impediments to Web Services implementing operations both using XLIFF documents and non-standard ones. It can be argued that this approach provides greater flexibility to the API provided by the Web Service. For demonstration purposes this is the approach taken in the implementation. In the listing of operations provided in section 3.2, in the cases where there are alternative operations, the one using XLIFF is appended with 'Xlf'. E.g.:

- opPseudoTranslateXlf: the input to the operation contains XLIFF documents
- opPseudoTranslate: the input to the operation contains documents using schema: <http://tcd.ie/nds/sw110n/ws/schema/pseudoTranslationXliff>

Web Service specific schemas greatly reduce the validation effort required from the Web Service. If the input data is originally XLIFF, the responsibility for validation is transferred to the client. Clients can detect the presence or not of optional XLIFF elements that are necessary for the Web Service and raise exceptions prior to the service being invoked. This shifts the complex-

ity of handling XLIFF flexibility onto the client. The disadvantage of this architecture is increased client complexity, but one advantage is better performance:

- validation errors can be detected locally without waiting for network connections
- reduced load on the Web Service improves its scalability and responsiveness

In the reference implementation, conversions between XLIFF and the Web Service specific schemas are handled using Java and DOM. An interesting possibility that hasn't been explored in this research would be to use XSLT transformations instead.

3.2 Web Service interfaces

The following Web Services interfaces have been designed (see complete WSDL files in Appendix 2):

3.2.1 Archiving

Operation	Description	
opArchive	Archives finised localised content	
<i>Input Message: msgArchiveRequest</i>		
Part	Element	Description
archiveRequest	xliff []	Xliff file(s) to be archived
<i>Output Message: msgArchiveResponse</i>		
Part	Element	Description
archiveResponse	id	Identifier(s) for the archived content
<i>Fault message(s):</i>		
Message	Description	
<i>OperationFailed</i>	Information message	

3.2.2 Leveraging

Operation	Description	
opLeverageXlf	Use the accompanying TMs to pretranslate the content sent	
<i>Input Message: msgLeverageXlfRequest</i>		
Part	Element	Description
leverageXlfRequest	targetLang	Target language
	xliff	Xliff file to be leveraged
	traslationMemory []	Array of TMs. Subelements: xliff []

Output Message: msgLeverageResponse		
Part	Element	Description
leverageResponse	xliff	Xliff file after leveraging
Fault message(s):		
Message		Description
<i>OperationFailed</i>		Information message

Operation	Description	
opLeverage	Use the accompanying TMs to pretranslate the content sent	
Input Message: msgLeverageRequest		
Part	Element	Description
leverageRequest	xliff-lev	Xliff-lev file to be leveraged
	traslationMemory []	Array of TMs. Subelements: xliff []
Output Message: msgLeverageResponse		
Part	Element	Description
leverageResponse	xliff	Xliff file after leveraging
Fault message(s):		
Message		Description
<i>OperationFailed</i>		Information message

Operation	Description	
opSubmitTm	Submit a TM to be used for leveraging at a later stage	
Input Message: msgSubmitTmRequest		
Part	Element	Description
submitTmRequest	transMemoryId	TM identifier
	xliff	TM
Output Message: msgSubmitTmResponse		
Part	Element	Description
submitTmResponse	acknowledgement	Acknowledgement message
Fault message(s):		
Message		Description
<i>OperationFailed</i>		Information message

Operation	Description	
opLeverageWith	Leverage content using the TM with the specified identifier	
Input Message: <i>msgLeverageWithRequest</i>		
Part	Element	Description
leverageWithRequest	targetLang	Target language
	transMemoryId	TM identifier
	xliff	Xliff file to be leveraged
Output Message: <i>msgLeverageWithResponse</i>		
Part	Element	Description
leverageWithResponse	xliff	Xliff file after leveraging
Fault message(s):		
Message		Description
<i>TranslationMemoryNotFound</i>		Information message
<i>OperationFailed</i>		Information message

3.2.3 Machine Translation

Operation	Description	
opMachineTranslateXlf	Machine translates the submitted content	
Input Message: <i>msgMachineTranslateXlfRequest</i>		
Part	Element	Description
MtXlfRequest	targetLang	target language
	xliff []	Xliff file(s) to be machine translated
Output Message: <i>msgMachineTranslateResponse</i>		
Part	Element	Description
MtResponse	xliff []	Xliff file(s) containing the machine translations
Fault message(s):		
Message		Description
<i>OperationFailed</i>		Information message

Operation	Description	
opMachineTranslate	Machine translates the submitted content	
Input Message: <i>msgMachineTranslateRequest</i>		
Part	Element	Description
MtRequest	xliff-mt []	Xliff-mt file(s) to be machine translated
Output Message: <i>msgMachineTranslateResponse</i>		

Part	Element	Description
MtResponse	xliff []	Xliff file(s) containing the machine translations
<i>Fault message(s):</i>		
Message		Description
<i>OperationFailed</i>		Information message

3.2.4 Parsing

Operation	Description	
opGetSupportedDataTypes	Supplies the list of data types (po, winres, etc) the service is able to process	
<i>Output Message: msgGetSupportedDataTypesResponse</i>		
Part	Element	Description
supportedDataTypesResponse	dataType []	Array of strings listing the supported data types
<i>Fault message(s):</i>		
Message		Description
<i>OperationFailed</i>		Information message

Operation	Description	
opParseFiles	Parse the submitted content and return their Xliff representation	
<i>Input Message: msgParseFilesRequest</i>		
Part	Element	Description
parseRequest	return-SingleXliff	Option to indicate whether to include the parsed files into a single or separate Xliff files
	fileInfo []	Array of fileInfo elements. Subelements: dataType: file data type fileLanguage: original language fileName: original file name fileContent: original content
<i>Output Message: msgParseFilesResponse</i>		
Part	Element	Description
parseResponse	xliff	Xliff files containing the submitted files

<i>Fault message(s):</i>	
Message	Description
UnsupportedDataType	Information message
<i>OperationFailed</i>	Information message

3.2.5 Pseudo translating

Operation	Description	
opPseudoTranslateXlf	Generates a pseudo translated version of the localisable content	
<i>Input Message: msgPseudoTranslateXlfRequest</i>		
Part	Element	Description
pseudoTranslateXlfRequest	targetLang	Target language
	usePrefix	Append a string
	prefix	Prefix
	useSuffix	Prepend a string
	suffix	Suffix
	excludeStringsOneWord	Do not change one word strings
	excludeStringsLessThan	Do not change strings less than n chars
	lessThan	Number of chars
	excludeStringsCamelCase	Do not change CaMeL CaSe strings
	excludeStringAllCaps	Do not change all caps strings
	excludeStringsNumeric	Do not change numeric only strings
	excludeTranslated	Use translation if available
	expansionRatio	Ratio to expand the string
xliff []	Xliff file(s) to process	
<i>Output Message: msgPseudoTranslateResponse</i>		
Part	Element	Description
pseudoTranslateResponse	xliff []	Xliff file(s) after pseudo translation
<i>Fault message(s):</i>		
Message	Description	
<i>OperationFailed</i>	Information message	

Operation	Description	
opPseudoTranslate	Generates a pseudo translated version of the localisable content	
Input Message: msgPseudoTranslateRequest		
Part	Element	Description
pseudoTranslateRequest	usePrefix	Append a string
	prefix	Prefix
	useSuffix	Prepend a string
	suffix	Suffix
	excludeStringsOneWord	Do not change one word strings
	excludeStringsLessThan	Do not change strings less than n chars
	lessThan	Number of chars
	excludeStringsCamelCase	Do not change CaMeL CaSe strings
	excludeStringAllCaps	Do not change all caps strings
	excludeStringsNumeric	Do not change numeric only strings
	expansionRatio	Ratio to expand the string
xliff-pseudo []	Xliff-pseudo file(s) to process	
Output Message: msgPseudoTranslateResponse		
Part	Element	Description
pseudoTranslateResponse	xliff []	Xliff file(s) after pseudo translation
Fault message(s):		
Message	Description	
OperationFailed	Information message	

3.2.6 QA checking

Operation	Description	
opQaCheckXliff	Run QA checks on the localised content	
Input Message: msgQaCheckXliffRequest		
Part	Element	Description
qaCheckXliffRequest	xliff []	Xliff file(s) to be processed
Output Message: msgQaCheckResponse		
Part	Element	Description
qaCheckResponse	xliff []	Xliff file(s) after QA processing

<i>Fault message(s):</i>	
Message	Description
<i>OperationFailed</i>	Information message

Operation	Description	
opQaCheck	Run QA checks on the localised content	
<i>Input Message: msgQaCheckRequest</i>		
Part	Element	Description
qaCheckRequest	xliff-qa []	Xliff-qa file(s) to be processed
<i>Output Message: msgQaCheckResponse</i>		
Part	Element	Description
qaCheckResponse	xliff []	Xliff file(s) after QA processing
<i>Fault message(s):</i>		
Message	Description	
<i>OperationFailed</i>	Information message	

3.2.7 Reporting

Operation	Description	
opGetReport	generate statistics on the submitted content	
<i>Input Message: msgReportRequest</i>		
Part	Element	Description
reportRequest	xliff []	Xliff file(s) to be processed
<i>Output Message: msgReportResponse</i>		
Part	Element	Description
reportResponse	reportResponseType	Report on the submitted Xliff files
<i>Fault message(s):</i>		
Message	Description	
<i>OperationFailed</i>	Information message	

3.2.8 Language review

Operation	Description	
opSubmitForReviewXlf	Submit content for language review	
<i>Input Message: msgSubmitForReviewXlfRequest</i>		
Part	Element	Description
submitForReviewXlfRequest	targetLang	Target language
	xliff	Xliff file to be reviewed

Output Message: msgSubmitForReviewResponse		
Part	Element	Description
submitForReviewResponse	id	Review job identifier
Fault message(s):		
Message		Description
<i>OperationFailed</i>		Information message

Operation	Description	
opSubmitForReview	Submit content for language review	
Input Message: msgSubmitForReviewRequest		
Part	Element	Description
submitForReviewRequest	xliff-rev	Xliff-rev file to be reviewed
Output Message: msgSubmitForReviewResponse		
Part	Element	Description
submitForReviewResponse	id	Review job identifier
Fault message(s):		
Message		Description
<i>OperationFailed</i>		Information message

Operation	Description	
opGetReview	Retrieve translated content	
Input Message: msgGetReviewRequest		
Part	Element	Description
getReviewRequest	id	Review job identifier
Output Message: msgGetReviewResponse		
Part	Element	Description
getReviewResponse	xliff	Reviewed Xliff file
Fault message(s):		
Message		Description
<i>OperationFailed</i>		Information message

3.2.9 Target generation

Operation	Description	
opGenerateTargetXlf	Generate a localised version of the original file	
Input Message: msgGenerateTargetXlfRequest		
Part	Element	Description

generateTargetXliffRequest	xliff	Xliff file touse for target generation
Output Message: msgGenerateTargetResponse		
Part	Element	Description
generateTargetResponse	targets []	Array of targets. Subelements: fileName – Name for the localised file fileContents – Localised file contents
Fault message(s):		
Message		Description
UnsupportedDataType		Information message
MissingSkeleton		Information message
OperationFailed		Information message

Operation	Description	
opGenerateTarget	Generate a localised version of the original file	
Input Message: msgGenerateTargetRequest		
Part	Element	Description
generateTargetRequest	xliff-tg	Xliff-tg file touse for target generation
Output Message: msgGenerateTargetResponse		
Part	Element	Description
generateTargetResponse	targets []	Array of targets. Subelements: fileName – Name for the localised file fileContents – Localised file contents
Fault message(s):		
Message		Description
UnsupportedDataType		Information message
OperationFailed		Information message

3.2.10 Testing

Operation	Description	
opTest	Submit a file to be used for testing	
Input Message: msgTestRequest		
Part	Element	Description
testRequest	xliff []	Xliff file(s)
Output Message: msgTestResponse		
Part	Element	Description
acknowledgement	acknowledgement	Confirmation message

<i>Fault message(s):</i>	
Message	Description
<i>OperationFailed</i>	Information message

3.2.11 Translation

Operation	Description	
opSubmitForTranslationXliff	Submit content for translation	
<i>Input Message: msgSubmitForTranslationXliffRequest</i>		
Part	Element	Description
submitForTranslationXliffRequest	targetLang	Target language
	xliff	Xliff file to be translated
<i>Output Message: msgSubmitForTranslationResponse</i>		
Part	Element	Description
submitForTranslationResponse	id	Translation job identifier
<i>Fault message(s):</i>		
Message	Description	
<i>OperationFailed</i>	Information message	

Operation	Description	
opSubmitForTranslation	Submit content for translation	
<i>Input Message: msgSubmitForTranslationRequest</i>		
Part	Element	Description
submitForTranslationRequest	xliff-tr	Xliff-tr file to be translated
<i>Output Message: msgSubmitForTranslationResponse</i>		
Part	Element	Description
submitForTranslationResponse	id	Translation job identifier
<i>Fault message(s):</i>		
Message	Description	
<i>OperationFailed</i>	Information message	

Operation	Description	
opGetTranslation	Retrieve translated content	
<i>Input Message: msgGetTranslationRequest</i>		
Part	Element	Description
getTranslationRequest	id	Translation job identifier
<i>Output Message: msgGetTranslationResponse</i>		

Part	Element	Description
getTranslationResponse	xliff	Translated Xliff file
<i>Fault message(s):</i>		
Message		Description
<i>OperationFailed</i>		Information message

3.2.12 Updating

Operation	Description	
opUpdate	Use the previous localised version of the software to pre-translate	
<i>Input Message: msgUpdateRequest</i>		
Part	Element	Description
updateRequest	targetLang	Target language
	xliff	Xliff file to be updated
	previousVersion	Subelements: xliff : previous localised version.
<i>Output Message: msgUpdateResponse</i>		
Part	Element	Description
updateResponse	xliff	Xliff file after leveraging
<i>Fault message(s):</i>		
Message		Description
<i>OperationFailed</i>		Information message

4. Evaluation

In the introduction we listed some potential benefits of using an SOA design over the model currently used for software localisation:

- Lessened vendor and tool lock-in which allows to pick best of breed solutions for each function.
- Increased interoperability between localisation tools.
- A more flexible and agile architecture. Through the use of services as components ISVs and LSP can react more quickly to new business demands.
- Increased automation: reduction in process management overhead through the use of automated workflows.

The reference implementation described in section 3 has been used to assess how realistic these objectives are.

4.1 Standard support

The foundation for lessened vendor and tool lock-in and increased localisation tool interoperability is the existence of supporting standards. In a ecosystem where ISVs and LSPs interact through Web Services standards enables the reuse of the same tools solution regardless of whom are the providers of those Web Services. The need for standards was raised by several of the industry experts interviewed for this dissertation.

The only standard currently available in the area of localisation Web Services is OASIS Translation-WS [6] reviewed in section 2.2. Translation-WS defines an interface to submit and receive localisation jobs between ISVs and LSPs. The standard targets the scenario where localisation is fully outsourced to an LSP; consequently it doesn't expose any of the localisation functions internal to the LSP and listed in table 3. The visibility offered on the internal processes at the LSP is limited to project status information. This lack of granularity makes the standard unsuitable for the model envisaged in this dissertation as it precludes the flexible composition of localisation services in an application.

There are a number of supporting localisation standards that are relevant to software localisation Web Services. XLIFF is the principal one but, as discussed in section 3.1, there are areas where it isn't clear the standard meets the needs of software localisation Web Services.

4.2 Software localisation workflows

To evaluate the possibilities for composability and increased automation that SOA can bring to software localisation two workflows were implemented using Business Process Execution Language (BPEL) -for the complete BPEL scripts see Appendix 3.

Localisation workflows can be quite complex as they encompass localisation tasks and more general project administration tasks such as budget control, work distribution or billing. A good description of a localisation workflow can be found in [23]. It presents a fairly comprehensive life cycle made up of fourteen steps applicable to content that require high quality localisation. It reflects the most common workflow currently performed in the localisation industry. The sample workflow named 'Translation' is an approximation though, because the focus of this dissertation is on software localisation, it excludes project management steps such as Costing & Approval, Work distribution or Billing & Collecting.

The second sample workflow, named 'Machine Translation', was designed to show how, through composition, it is easy to develop workflows that can be used to process content with different requirements. It is a short workflow used to automatically provide gist translations. It reuses services such as the File Parsing and Target Generation, used in the Translation workflow, and matches them with the Machine Translation service instead. It shows the possibilities to deviate from the workflow found at [23] in innovative ways.

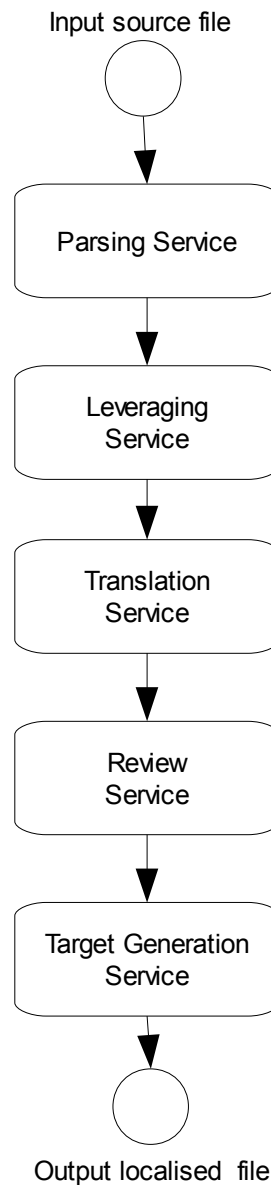
4.2.1 Translation Workflow

This is the type of workflow that can be used for content that require high quality localisation.

The input to this workflow are the files that require localisation. Steps:

- uses the Parsing Service to extract the localisable content as an XLIFF file.
- uses the Leveraging Service for pre-translation
- uses the Translation Service to translate any remaining untranslated content
- uses the Review Service for language review
- finally it uses the Target Generation Service to obtain a localised version of the original files

The output of the workflow are these localised files.

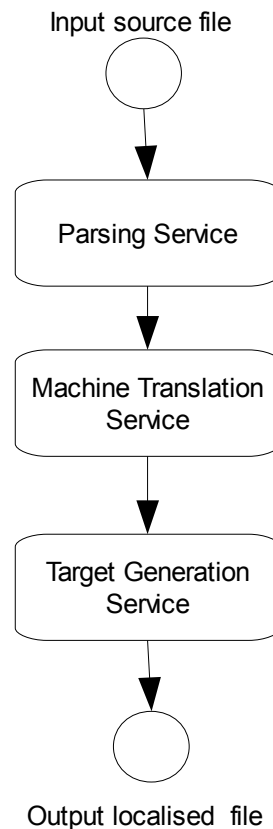


4.2.2 Machine Translation Workflow

This is the type of workflow that could be used for content that does not require high quality localisation. An example could be user generated content such as contributions to technical support forums. For this type of content a gist translation may be all that is required. The input to this workflow are the files (.xml, .html, etc) that need localisation. Steps:

- uses the Parsing Service to extract the localisable content as an XLIFF file.
- uses the Machine Translation Service to obtain gist translations
- finally it uses the Target Generation Service to obtain localised versions of the original files

The output of the workflow are these localised files.



4.3 Performance

Software localisation Web Services introduces a performance penalty compared with today's, desktop type applications. This penalty comes from several sources:

- the Web Services and the client applications may internally use data repositories different from the XML request and response documents specified in the WSDL interface for that service. In that case there is an overhead associated with the transformations to and from the data store (e.g. an RDBS) and the XML document. In the reference implementation, where XLIFF is used extensively, this would be the case for Web Services and clients that do not use XLIFF as their native data store.
- the overhead coming from serializing and de-serializing the XML request and response documents so that they can be communicated through SOAP.
- network latency.

There is a great variety of data stores used by localisation tools, from relational databases such as MySQL to proprietary binary formats such as Alchemy Catalyst TTK. The performance im-

plications of the transformations between them and XLIFF are plentiful and difficult to categorized. This dissertation does not attempt to do so. Exploring network latency in a meaningful way, specially for connections across the Internet, requires a large testing setup, and is not specific to localisation Web Services but to any Web Service. This is also left for future research.

Still this research wants to provide some basic performance figures that, at a minimum, can validate the basic viability of the setup developed for the reference implementation. A number of tests were performed using the implementation for the Leveraging service. Note the following:

- The client as well as the Web Service use XLIFF natively so no transformations to and from XLIFF are performed
- A number of sample XLIFF files were created with word-counts ranging from 1,000 to 500,000. The file sizes are: 1000 – 17KB, 5,000 – 85KB, 10,000 – 171KB, 50,000 – 859KB, 100,000 – 1.72MB, 500,000 – 8.63MB
- A first set of tests run the client and Web Service software on the same machine to try an eliminate network latency. This set is labelled 'Local host' below. A second set of test, labelled 'Remote host', run the Web Service on a server inside Trinity College network and the client on a computer located in a Dublin residence. The connection was made through an Internet Service Provider.
- Five measurements were taken for each test and the average was used. For the full results see Appendix 6.

Table 5 presents the results of the tests. The tests focus mainly on the overheads associated with serializing and de-serializing XML, this is, the DOM processing time. In the case of the tests labelled 'Remote host' it also includes network latency.

	Wordcount					
	1,000	5,000	10,000	50,000	100,000	500,000
Local host	0.030s	0.078s	0.118s	0.457s	0.906s	4.426s
Remote host	0.167s	0.524s	0.769s	3.166s	6.670s	31.800s

Table 5: Leveraging Web Service performance measurements

4.4 Security considerations

The following points should be considered when designing localisation Web Services.

4.4.1 Confidentiality

Generally, clients of software localisation services do not regard their past translation assets as a competitive advantage. But the translation content for a current localisation project is considered highly sensitive as it gives away information on the features included in the software product being localised. These days most localisation projects are done in parallel with the development of the original product so that the software can be simultaneously shipped to several international markets as well as the home market. Information on the upcoming features can be of interest to competitors.

Localisation metadata such as schedules, project statistics (wordcounts), etc can also be sensitive because it can also be used to draw information on the software product: the launch date for the original and localised versions, the size (and thus the importance) of the software, etc

There is a need to protect the confidentiality of the localisation content as well as the localisation metadata being exchanged between the consumer and provider of localisation Web Services. This issue is compounded by the fact that a Web Service may itself consume the services of other Web Services. This type of composite service is common in Service Oriented Architectures and is the basis for the flexibility afforded by SOA. For example a leveraging Web Service may be consuming a segmentation Web Service to improve the number of matches found. The chain of services being called may not be known to the original customer.

In general the use of transport layer security, such as HTTPS, is discouraged. Instead the use of message level security based on WS-Security is preferred. Message level security provides a finer level of security granularity which makes it better suited to deal with composite services. It can also be used to provide some protection against service provider abuse which may be a consideration when the reputation of a third party service is unknown.

4.4.2 Data integrity

Software localisation Web Services may be used in the context of commercial software as well as in Open Source Software (OSS). OSS, by the nature of its public development process, have no interest in confidentiality but it shares with commercial software a need for data integrity. In both cases the localisation content as well as metadata requires protection from tampering while en route.

The use of WS-Security and XML Signature specifications is expected as well as related ones such as XML Canonicalization.

4.4.3 Authentication

Authentication is a common requirement for localisation Web Services. In many cases there will be costs associated with the use of the services so the identities of the parties need to be ascertained. This is especially true considering that Web Services allow for greater process automation.

According to Kanneganti et al. in [16], best practices for authentication use WS-Security in combination with digital certificates based on a Public Key Infrastructure.

4.4.4 Authorization

The clear consensus among the industry experts interviewed for this dissertation (see section 2.5) was that the different roles and associated access control lists (ACLs) should be left to the parties to agree. The variability in this area is such that any attempt to standardise would become too restrictive. Therefore this area is better left to each party to negotiate based on internal security policies and repositories.

4.4.5 Non repudiation

The services provided by a software localisation Web Service will commonly be offered by commercial LSPs. In the context of a commercial transaction it is desirable for both parties to ensure non-repudiation. The setup required to achieve non-repudiation is not trivial. Generally it requires the use of Public Key Infrastructure certificates but also logging that ensures the validity of the certificates at the time when the transaction took place.

5. Conclusion and future work

Standard support for software localisation Web Services is largely missing. This support is crucial to realize some of the potential benefits of a Service Oriented Architecture, specifically tool interoperability and lessening vendor and tool lock-in. Building this support will require a large consensus across the localisation industry including ISVs, LSPs and LTPs. It will require a public specification of the Web Service interfaces published by an standards body (OASIS, LISA) and its adoption by the different players, specially the tools providers.

But there is no need to see standard support for all possible software localisation Web Services at once. In fact a list of software localisation Web Services is a moving target. Software localisation functions can be broken at different levels of abstraction and, the same way that the interface defined by Translation-WS [6] can be regarded as too high level, there may be a need to increase / decreased their granularity in the future. Interfaces for different Web Services can be added to

the specification gradually. Some interfaces could be prioritised according to a number of criteria:

- Simplicity: Web Services that require less complex interfaces could be standardised first
- Commercial viability: interfaces for Web Services that are already offered on a commercial basis, such as Machine Translation, could be given a higher priority
- In house services: some Web Services are less likely to be involve in interactions between third parties. For example, a Web Service offering archiving of final project content is likely to be hosted in-house. Standards benefit in house services through increased tool interoperability. But services involving exchanges between third parties benefit not only from tool interoperability but also lessened lock-in to any given supplier of the service. These latter services may be given a higher priority.

It should be noted that XLIFF may need to be changed to accommodate the demands from localisation Web Services. The issues raised in section 4 need to be considered, specially the lack of representation guides for a majority of localisation file formats. Without them, compliant but incompatible implementations may arise jeopardizing interoperability which should be the prime goal for a standard.

The lack of standard support, even though important, doesn't preclude all the benefits that may come from a Service Oriented Architecture. Exposing software localisation functions as Web Services makes it easier to build automated workflows. There are considerable cost-savings from the reduction in management overheads associated with manual steps. Also, the platform independent nature of Web Services provides companies with an insurance against changes in the underlying technology. Web Services provide a level of abstraction that makes it easier to align the implementation to changing business and technical conditions.

These advantages still remain for deployment scenarios where ISVs and LSPs interact through ad-hoc interfaces. And they are particularly relevant in the closed access scenario that, from the interviews to industry experts carried out during this research (see section 2.5), seems to be the prevalent one in the localisation industry. In this in-house scenario localisation Web Services are deployed inside a company's intranet and are not accessible externally neither to customers nor suppliers - different departments or teams offer and consume the available Web Services. In this type of deployment:

- Security can leverage the authentication mechanism used for the companies' intranet. Also, the Web Service runs behind the companies' firewall so data confidentiality may

not be required

- Variations in performance are less unpredictable. This applies both to network latency and the Web Service response times. Both the network and server resources are under the companies' control so the option exists of scaling them according to usage.

The higher level of control over closed access Web Services makes them safer and more reliable. So it is normal that the first deployments of localisation Web Services fit this model. As ISVs and LSPs become more confident using Web Services, a hybrid model with closed and open access Web Services, may become more common.

There are some areas that could be explored by future research. One of these involves a more detailed performance analysis of software localisation Web Services. It is clear that transformations to and from XLIFF add overhead not present in today's, desktop type applications. It would be interesting to explore ways in which it could be minimized.

Another area that could be explored is the usability levels of the workflow engines available in the market. In the localisation industry the people normally in charge of schedule and process control are the localisation project managers. SOA offers the ability to adapt workflows to changing business needs through Web Service Orchestration. The natural owners of such adaptations are the localisation project managers. But project managers are not necessarily highly technical; they often defer on the localisation engineer for technical matters. During the development of the reference implementation a number of workflows were produced using BPEL. It would seem that BPEL exposes an interface that is too low level and too technical to be manipulated by localisation project managers. Thus BPEL workflows, though useful, seem more suitable for static workflows that only require occasional changes. It would be interesting to ascertain if other Web Service orchestration engines present the higher level business metaphors project managers are used to. This could enable them to adapt the workflow on a more granular and potentially more effective manner. This research should include BPEL and YAWL and the tools that support them, such as Netbeans, and also the existing commercial offerings from the likes of BIGAZI, Fujitsu, SAP, Microsoft or IBM.

References

- [1] Erl, T, (2008).SOA Design Patterns, Prentice Hall .
- [2] Lewis, D, Curran, S., Jones, D., Moran, J. (2009). An Open Service Framework for Next Generation Localisation.
- [3] Lewis, D., Curran, S., Feeney, K., Etzioni, Z., Keeney, J., Way, A., and Schäler, R. (2009). Web service integration for next generation localisation. In Proceedings of the Workshop on Software Engineering, Testing, and Quality Assurance For Natural Language Processing (Boulder, Colorado, June 05 - 05, 2009). ACL Workshops. Association for Computational Linguistics, Morristown, NJ, 47-55.
- [4] Internationalization Tag Set (ITS) (2007) Version 1.0 W3C Recommendation 03 April 2007, accessed from <http://www.w3.org/TR/its/> on 03 Apr 2010
- [5] Reynolds, P. (2006), Using Web Services for Translation, accessed from <http://www.oasis-open.org/committees/download.php/21054/TWS%20Whitepaper.doc> on 03 Apr 2010
- [6] Reynolds, P. et al (2007) “Translation Web Services - draft committee specification” OASIS Translation Web Services TC, Proposed draft specification 16 May 2007, access from <http://www.oasis-open.org/committees/download.php/24350/trans-ws-spec-1.0.3.html> on 03 Apr 2010
- [7] XLIFF “A white paper on version 1.2 of the XML Localisation Interchange File Format (XLIFF)”, Revision: 1.0, 17 Oct 2007 accessed from <http://xml.coverpages.org/XLIFF-Core-WhitePaper200710-CSv12.pdf> on 03 Apr 2010
- [8] XLIFF Version 1.2, OASIS Standard, 1 February 2008, accessed from <http://docs.oasis-open.org/xliff/xliff-core/xliff-core.html> on 03 Apr 2010
- [9] TMX 1.4b Specification OSCAR Recommendation, Localisation Industry Standards Association, 26 April 2005, accessed from <http://www.lisa.org/fileadmin/standards/tmx1.4/tmx.htm> on 03 Apr 2010
- [10] TermBase eXchange TBX Localisation Industry Standards Association, 29 October 2008, accessed from http://www.lisa.org/fileadmin/standards/TBX_2008_10_29.pdf on 03 Apr 2010
- [11] Segmentation Rules eXchange SRX 2.0 Specification OSCAR Recommendation, Localisation Industry Standards Association, 7 April 2008, accessed from <http://www.lisa.org/fileadmin/standards/srx20.pdf> on 03 Apr 2010

- [12] Erl, T. (2008), SOA Principles of Service Design, Prentice Hall
- [13] Erl, T, Karmarkar, A., Walmsley, P., Haas, H., Yalcinalp, U., Liu, C.K., Orchard, D., Tost, A., Pasley, J. (2009). Web Service Contract Design & Versioning for SOA, Prentice Hall
- [14] Kailn, M. (2009). Java Web Services up and running, O'Reilly, 2009
- [15] Richardson, L., Ruby, S. (2007). RESTful Web Services, O'Reilly
- [16] Kanneganti, R., Chodavarpu, P. (2008). SOA Security, Manning
- [17] Gudgin, M. et al. SOAP Version 1.2, April 2007, W3C Recommendation, accessed from <http://www.w3.org/TR/soap12-part1/> on 03 Apr 2010
- [18] Graham, S. et al. Web Service Base Notification 1.3, 01 October 2006, OASIS standard, accessed from http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf on 0 Apr 2010
- [19] Box, D., Curbera, F. et al. Web Services Addressing (WS-Addressing), August 2004. W3C Member Submission, accessed from <http://www.w3.org/Submission/2004/SUBM-ws-addressing-20040810/> on 03 Apr 2010
- [20] Vedamuthu A. S. et al. Web Services Policy 1.5, 04 September 2007, W3C Recommendation accessed from <http://www.w3.org/TR/ws-policy/> on 03 Apr 2010
- [21] Christensen, E. et al. Web Services Description Language (WSDL) 1.1, 15 March 2001, W3C Note, accessed from <http://www.w3.org/TR/wsdl> on 03 Apr 2010
- [22] OAXAL Reference Model Version 1.0, OASIS Standard, 11 December 2009, accessed from <http://www.oasis-open.org/committees/download.php/35736/OASIS%20Open%20Architecture%20for%20XML%20Authoring%20and%20Localization%20Reference%20Model%20%28OAXAL%29.pdf> on 03 Apr 2010
- [23] Automated Localization Workflow – A Reference Model, accessed from <http://www.i18n.ca/publications/AutoLocalizationModel.pdf> on 05 Sep 2010
- [24] XLIFF 1.2 Representation Guide for Java Resource Bundles, OASIS Committee Draft 02, 30 April 2007, accessed from <http://docs.oasis-open.org/xliff/v1.2/xliff-profile-java/xliff-profile-java-v1.2-cd02.html> on 03 Apr 2010
- [25] XLIFF 1.2 Representation Guide for HTML, OASIS Committee Draft 02, 30 April 2007, accessed from <http://docs.oasis-open.org/xliff/v1.2/xliff-profile-html/xliff-profile-html-1.2-cd02.html> on 03 Apr 2010

- [26] XLIFF 1.2 Representation Guide for Gettext PO, OASIS Committee Draft 02, 30 April 2007, accessed from <http://docs.oasis-open.org/xliff/v1.2/xliff-profile-po/xliff-profile-po-1.2-cd02.html> on 03 Apr 2010
- [27] Hogan, J., Ho-Stuart, C., Pham, B., Key challenges in software internationalisation, (2004), ACSW Frontiers '04: Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation, Australian Computer Society, Inc.
- [28] Frimannsson, Asgeir and Hogan, James M. (2005) Adopting Standards Based XML file formats in Open Source Localisation. Localisation Focus – The International Journal of Localisation, 4. pp. 9-23.

Appendix 1: Questionnaire used for the interviews with Industry Experts

1. Do you use Web Services currently in your organization?
2. If you do, do you only use them internally or do you expose them to third parties?
3. Do you think Service Oriented Architecture (SOA) is a viable architecture for building a software localisation (SW localisation) solution?
4. What business problems if any do you think can be better handled through the use of Web Services for SW localisation? E.g.: scalability, flexibility, etc
5. What benefits if any do you think SOA can bring to the different players in the SW localisation industry? Players being generically grouped as localisation clients, localisation service providers (LSP) and localisation tools vendors.
6. Factors such as the type of content or financial or time constraints make a project demand different localisation work-flows. A potential benefit of Web Services is that they can help handle work-flow variability and still maintain high levels of automation. Is this something that could benefit your organization?
7. Web Services:
 - 7.1. Which of the following SW localisation functions do you think should be exposed as a service:
 - a) File Parsing: parse the supplied files looking for localisable content.
 - b) Leveraging: use a translation memory to pre-translate
 - c) Machine Translating: use machine translation to pre-translate
 - d) Updating: use the previous localised version of the software to pre-translate
 - e) Generate localised file: generate a localised version of the original file
 - f) QA checking: run automatic QA checks on the localised content
 - g) Reporting: generate statistics on the supplied content
 - h) Pseudo translating: generate a pseudo translated version of the localisable content
 - i) Artwork/Multimedia processing: localise artwork/multimedia content
 - j) Translating: translate the localisable content
 - k) Language reviewing: asses and rate the quality of the translation
 - l) Testing: test for defects in the localised product
 - m) Archiving: permanent storage for finished localised content
 - 7.2. Are there other functions you think should be exposed as services
 - 7.3. Could you prioritise them for adoption, or alternatively group them into ones you might be willing to use/pay for in the near, medium or long term.
 - 7.4. How might you use or implement/offer them? e.g. as part of a workflow, or integrated into a CAT tool
 - 7.5. What interchanged XML based format do you think should be used for a framework such as this? XLIFF?
8. Which of the following non functional requirements do you think should also be formalised as part of a Web Services framework for SW localisation:
 - 8.1. Security
 - 8.2. Service Quality (e.g. uptime, response time, etc)
 - 8.3. Constraints on temporal or spatial availability (e.g. service geographic location, etc)
 - 8.4. Methods for charging and payment
 - 8.5. Rights attached to the service

Appendix 2: Interfaces for Software Localisation Web Services

1. Archiving Service

1.1 Archiving.wsdl

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<definitions
  name="Archiving"
  targetNamespace="http://tcd.ie/nds/swl10n/ws/contract/archiving"
  xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/archiving"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ns1="http://tcd.ie/nds/swl10n/ws/schema/archiving">
  <types>
    <xs:schema
      targetNamespace="http://tcd.ie/nds/swl10n/ws/schema/archiving"
      xmlns:tns="http://tcd.ie/nds/swl10n/ws/schema/archiving"
      xmlns:xf="urn:oasis:names:tc:xliff:document:1.2"
      xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:import namespace="urn:oasis:names:tc:xliff:document:1.2"
        schemaLocation="xliff-core-1.2-strict.xsd"/>
      <xs:element name="archiveRequest" type="tns:archiveRequestType"/>
      <xs:element name="archiveResponse" type="tns:archiveResponseType"/>
      <xs:element name="OperationFailed" type="xs:string"/>
      <xs:complexType name="archiveRequestType">
        <xs:sequence>
          <xs:element ref="xf:xliff" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="archiveResponseType">
        <xs:sequence>
          <xs:element name="id" type="xs:string" minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
  </types>
  <message name="msgArchiveRequest">
    <part name="Request" element="ns1:archiveRequest"/>
  </message>
  <message name="msgArchiveResponse">
    <part name="Response" element="ns1:archiveResponse"/>
  </message>
  <message name="OperationFailed">
    <part name="fault" element="ns1:OperationFailed"/>
  </message>
  <portType name="ptArchiving">
    <operation name="opArchive">
      <input message="tns:msgArchiveRequest"/>
      <output message="tns:msgArchiveResponse"/>
      <fault message="tns:OperationFailed" name="OperationFailed"/>
    </operation>
  </portType>
  <binding name="Binding1" type="tns:ptArchiving">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <operation name="opArchive">
      <soap:operation soapAction="">
        <input>
          <soap:body use="literal"/>
        </input>
        <output>
```

```

        <soap:body use="literal"/>
    </output>
    <fault name="OperationFailed">
        <soap:fault name="OperationFailed" use="literal"/>
    </fault>
</operation>
</binding>
<service name="svArchiving">
    <port name="archiving" binding="tns:Binding1">
        <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
    </port>
</service>
</definitions>

```

2. Leveraging Service

2.1 Leveraging.wsdl

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<definitions
    name="Leveraging"
    targetNamespace="http://tcd.ie/nds/swl10n/ws/contract/leveraging"
    xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/leveraging"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:ns1="http://tcd.ie/nds/swl10n/ws/schema/leveraging">
    <types>
        <xs:schema
            targetNamespace="http://tcd.ie/nds/swl10n/ws/schema/leveraging"
            xmlns:tns="http://tcd.ie/nds/swl10n/ws/schema/leveraging"
            xmlns:xf="urn:oasis:names:tc:xliff:document:1.2"
            xmlns:lex="http://tcd.ie/nds/swl10n/ws/schema/leveragingXliff"
            xmlns:xs="http://www.w3.org/2001/XMLSchema">
            <xs:import namespace="urn:oasis:names:tc:xliff:document:1.2"
                schemaLocation="xliff-core-1.2-strict.xsd"/>
            <xs:import namespace="http://tcd.ie/nds/swl10n/ws/schema/leveragingXliff"
                schemaLocation="LeveragingXliff.xsd"/>
            <xs:element name="leverageXlfRequest" type="tns:leverageXlfRequestType"/>
            <xs:element name="leverageRequest" type="tns:leverageRequestType"/>
            <xs:element name="leverageResponse" type="tns:leverageResponseType"/>
            <xs:element name="submitTmRequest" type="tns:submitTmRequestType"/>
            <xs:element name="submitTmResponse" type="tns:submitTmResponseType"/>
            <xs:element name="leverageWithRequest" type="tns:leverageWithRequestType"/>
            <xs:element name="leverageWithResponse" type="tns:leverageWithResponseType"/>
            <xs:element name="TranslationMemoryNotFound" type="xs:string"/>
            <xs:element name="OperationFailed" type="xs:string"/>
            <xs:complexType name="leverageXlfRequestType">
                <xs:sequence>
                    <xs:element name="targetLang" type="xs:string" minOccurs="0"/>
                    <xs:element ref="xf:xliff" minOccurs="0"/>
                    <xs:element name="translationMemory" type="tns:translationMemoryType"
                        minOccurs="0"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="leverageRequestType">
                <xs:sequence>
                    <xs:element ref="lex:xliff-lev" minOccurs="0"/>
                    <xs:element name="translationMemory" type="tns:translationMemoryType"
                        minOccurs="0"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="translationMemoryType">
                <xs:sequence>
                    <xs:element ref="xf:xliff" minOccurs="0" maxOccurs="unbounded"/>
                </xs:sequence>
            </xs:complexType>
        </xs:schema>
    </types>

```

```

</xs:complexType>
<xs:complexType name="leverageResponseType">
  <xs:sequence>
    <xs:element ref="xlf:xliff" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="submitTmRequestType">
  <xs:sequence>
    <xs:element name="transMemoryId" type="xs:string" minOccurs="0"/>
    <xs:element ref="xlf:xliff" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="submitTmResponseType">
  <xs:sequence>
    <xs:element name="acknowledgement" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="leverageWithRequestType">
  <xs:sequence>
    <xs:element name="targetLang" type="xs:string" minOccurs="0"/>
    <xs:element name="transMemoryId" type="xs:string" minOccurs="0"/>
    <xs:element ref="xlf:xliff" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="leverageWithResponseType">
  <xs:sequence>
    <xs:element ref="xlf:xliff" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>
</types>
<message name="msgLeverageXlfRequest">
  <part name="Request" element="ns1:leverageXlfRequest"/>
</message>
<message name="msgLeverageRequest">
  <part name="Request" element="ns1:leverageRequest"/>
</message>
<message name="msgLeverageResponse">
  <part name="Response" element="ns1:leverageResponse"/>
</message>
<message name="msgSubmitTmRequest">
  <part name="Request" element="ns1:submitTmRequest"/>
</message>
<message name="msgSubmitTmResponse">
  <part name="Response" element="ns1:submitTmResponse"/>
</message>
<message name="msgLeverageWithRequest">
  <part name="Request" element="ns1:leverageWithRequest"/>
</message>
<message name="msgLeverageWithResponse">
  <part name="Response" element="ns1:leverageWithResponse"/>
</message>
<message name="TranslationMemoryNotFound">
  <part name="fault" element="ns1:TranslationMemoryNotFound"/>
</message>
<message name="OperationFailed">
  <part name="fault" element="ns1:OperationFailed"/>
</message>
<portType name="ptLeveraging">
  <operation name="opLeverageXlf">
    <input message="tns:msgLeverageXlfRequest"/>
    <output message="tns:msgLeverageResponse"/>
    <fault message="tns:OperationFailed" name="OperationFailed"/>
  </operation>
  <operation name="opLeverage">
    <input message="tns:msgLeverageRequest"/>
    <output message="tns:msgLeverageResponse"/>
    <fault message="tns:OperationFailed" name="OperationFailed"/>
  </operation>
</portType>

```

```

</operation>
<operation name="opSubmitTm">
  <input message="tns:msgSubmitTmRequest"/>
  <output message="tns:msgSubmitTmResponse"/>
  <fault message="tns:OperationFailed" name="OperationFailed"/>
</operation>
<operation name="opLeverageWith">
  <input message="tns:msgLeverageWithRequest"/>
  <output message="tns:msgLeverageWithResponse"/>
  <fault message="tns:TranslationMemoryNotFound"
name="TranslationMemoryNotFound"/>
  <fault message="tns:OperationFailed" name="OperationFailed"/>
</operation>
</portType>
<binding name="Binding1" type="tns:ptLeveraging">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
  <operation name="opLeverageXlf">
    <soap:operation soapAction=""/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
    <fault name="OperationFailed">
      <soap:fault name="OperationFailed" use="literal"/>
    </fault>
  </operation>
  <operation name="opLeverage">
    <soap:operation soapAction=""/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
    <fault name="OperationFailed">
      <soap:fault name="OperationFailed" use="literal"/>
    </fault>
  </operation>
  <operation name="opSubmitTm">
    <soap:operation soapAction=""/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
    <fault name="OperationFailed">
      <soap:fault name="OperationFailed" use="literal"/>
    </fault>
  </operation>
  <operation name="opLeverageWith">
    <soap:operation soapAction=""/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
    <fault name="TranslationMemoryNotFound">
      <soap:fault name="TranslationMemoryNotFound" use="literal"/>
    </fault>
    <fault name="OperationFailed">
      <soap:fault name="OperationFailed" use="literal"/>
    </fault>
  </operation>
</binding>

```

```

<service name="svLeveraging">
  <port name="leveraging" binding="tns:Binding1">
    <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
  </port>
</service>
</definitions>

```

2.2 LeveragingXliff.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://tcd.ie/nds/swl10n/ws/schema/leveragingXliff"
  xmlns:tns="http://tcd.ie/nds/swl10n/ws/schema/leveragingXliff"
  elementFormDefault="qualified"
  xmlns:xliff="urn:oasis:names:tc:xliff:document:1.2"
  xmlns:xml="http://www.w3.org/XML/1998/namespace">
  <xsd:import namespace="urn:oasis:names:tc:xliff:document:1.2" schemaLocation="xliff-
core-1.2-strict.xsd"/>
  <xsd:import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="http://www.w3.org/2001/xml.xsd"/>
  <xsd:element name="xliff-lev">
    <xsd:complexType>
      <xsd:sequence maxOccurs="unbounded">
        <xsd:element ref="tns:file-lev"/>
      </xsd:sequence>
      <xsd:attribute name="version" type="xliff:AttrType_Version" use="required"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="file-lev">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="tns:body-lev"/>
      </xsd:sequence>
      <xsd:attribute name="original" type="xsd:string" use="required"/>
      <xsd:attribute name="source-language" type="xsd:language" use="required"/>
      <xsd:attribute name="datatype" type="xliff:AttrType_datatype" use="required"/>
      <xsd:attribute name="tool-id" type="xsd:string" use="optional"/>
      <xsd:attribute name="date" type="xsd:dateTime" use="optional"/>
      <xsd:attribute ref="xml:space" use="optional"/>
      <xsd:attribute name="category" type="xsd:string" use="optional"/>
      <xsd:attribute name="target-language" type="xsd:language" use="required"/>
      <xsd:attribute name="product-name" type="xsd:string" use="optional"/>
      <xsd:attribute name="product-version" type="xsd:string" use="optional"/>
      <xsd:attribute name="build-num" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:unique name="U_group_id">
      <xsd:selector xpath="//xliff:group"/>
      <xsd:field xpath="@id"/>
    </xsd:unique>
    <xsd:key name="K_unit_id">
      <xsd:selector xpath="//xliff:trans-unit|//xliff:bin-unit"/>
      <xsd:field xpath="@id"/>
    </xsd:key>
    <xsd:keyref name="KR_unit_id" refer="xliff:K_unit_id">
      <xsd:selector xpath="//bpt|//ept|//it|//ph|//g|//x|//bx|//ex|//sub"/>
      <xsd:field xpath="@xid"/>
    </xsd:keyref>
    <xsd:keyref name="KR_alt-trans_tool-id" refer="xliff:K_tool-id">
      <xsd:selector xpath="//xliff:trans-unit/xliff:alt-trans"/>
      <xsd:field xpath="@tool-id"/>
    </xsd:keyref>
    <xsd:unique name="U_context-group_name">
      <xsd:selector xpath="//xliff:context-group"/>
      <xsd:field xpath="@name"/>
    </xsd:unique>
  </xsd:element>
  <xsd:element name="body-lev">
    <xsd:complexType>

```

```

    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:group-lev"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:trans-unit-lev"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
<xsd:element name="group-lev">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:sequence>
        <xsd:element maxOccurs="unbounded" minOccurs="0" ref="xlf:context-group"/>
      </xsd:sequence>
      <xsd:choice maxOccurs="unbounded">
        <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:group-lev"/>
        <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:trans-unit-lev"/>
      </xsd:choice>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:string" use="optional"/>
    <xsd:attribute name="datatype" type="xlf:AttrType_datatype" use="optional"/>
    <xsd:attribute default="default" ref="xml:space" use="optional"/>
    <xsd:attribute name="restype" type="xlf:AttrType_restype" use="optional"/>
    <xsd:attribute name="resname" type="xsd:string" use="optional"/>
    <xsd:attribute name="extradata" type="xsd:string" use="optional"/>
    <xsd:attribute name="extype" type="xsd:string" use="optional"/>
    <xsd:attribute name="help-id" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="menu" type="xsd:string" use="optional"/>
    <xsd:attribute name="menu-option" type="xsd:string" use="optional"/>
    <xsd:attribute name="menu-name" type="xsd:string" use="optional"/>
    <xsd:attribute name="coord" type="xlf:AttrType_Coordinates" use="optional"/>
    <xsd:attribute name="font" type="xsd:string" use="optional"/>
    <xsd:attribute name="css-style" type="xsd:string" use="optional"/>
    <xsd:attribute name="style" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="exstyle" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute default="yes" name="translate" type="xlf:AttrType_YesNo"
use="optional"/>
    <xsd:attribute default="yes" name="reformat" type="xlf:AttrType_reformat"
use="optional"/>
    <xsd:attribute default="pixel" name="size-unit" type="xlf:AttrType_size-unit"
use="optional"/>
    <xsd:attribute name="maxwidth" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="minwidth" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="maxheight" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="minheight" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="maxbytes" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="minbytes" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="charclass" type="xsd:string" use="optional"/>
    <xsd:attribute default="no" name="merged-trans" type="xlf:AttrType_YesNo"
use="optional"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="trans-unit-lev">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xlf:source"/>
      <xsd:element minOccurs="0" ref="xlf:seg-source"/>
      <xsd:element minOccurs="0" ref="xlf:target"/>
      <xsd:choice maxOccurs="unbounded" minOccurs="0">
        <xsd:element ref="xlf:context-group"/>
        <xsd:element ref="xlf:alt-trans"/>
      </xsd:choice>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:string" use="required"/>
    <xsd:attribute name="approved" type="xlf:AttrType_YesNo" use="optional"/>
    <xsd:attribute default="yes" name="translate" type="xlf:AttrType_YesNo"
use="optional"/>
    <xsd:attribute default="yes" name="reformat" type="xlf:AttrType_reformat"
use="optional"/>
    <xsd:attribute default="default" ref="xml:space" use="optional"/>

```

```

<xsd:attribute name="datatype" type="xlf:AttrType_datatype" use="optional"/>
<xsd:attribute name="phase-name" type="xsd:string" use="optional"/>
<xsd:attribute name="restype" type="xlf:AttrType_restype" use="optional"/>
<xsd:attribute name="resname" type="xsd:string" use="optional"/>
<xsd:attribute name="extradata" type="xsd:string" use="optional"/>
<xsd:attribute name="extype" type="xsd:string" use="optional"/>
<xsd:attribute name="help-id" type="xsd:NMTOKEN" use="optional"/>
<xsd:attribute name="menu" type="xsd:string" use="optional"/>
<xsd:attribute name="menu-option" type="xsd:string" use="optional"/>
<xsd:attribute name="menu-name" type="xsd:string" use="optional"/>
<xsd:attribute name="coord" type="xlf:AttrType_Coordinates" use="optional"/>
<xsd:attribute name="font" type="xsd:string" use="optional"/>
<xsd:attribute name="css-style" type="xsd:string" use="optional"/>
<xsd:attribute name="style" type="xsd:NMTOKEN" use="optional"/>
<xsd:attribute name="exstyle" type="xsd:NMTOKEN" use="optional"/>
<xsd:attribute default="pixel" name="size-unit" type="xlf:AttrType_size-unit"
use="optional"/>
<xsd:attribute name="maxwidth" type="xsd:NMTOKEN" use="optional"/>
<xsd:attribute name="minwidth" type="xsd:NMTOKEN" use="optional"/>
<xsd:attribute name="maxheight" type="xsd:NMTOKEN" use="optional"/>
<xsd:attribute name="minheight" type="xsd:NMTOKEN" use="optional"/>
<xsd:attribute name="maxbytes" type="xsd:NMTOKEN" use="optional"/>
<xsd:attribute name="minbytes" type="xsd:NMTOKEN" use="optional"/>
<xsd:attribute name="charclass" type="xsd:string" use="optional"/>
<xsd:attribute default="yes" name="merged-trans" type="xlf:AttrType_YesNo"
use="optional"/>
</xsd:complexType>
<xsd:unique name="U_tu_segsrc_mid">
  <xsd:selector xpath="./xlf:seg-source/xlf:mrk"/>
  <xsd:field xpath="@mid"/>
</xsd:unique>
<xsd:keyref name="KR_tu_segsrc_mid" refer="xlf:U_tu_segsrc_mid">
  <xsd:selector xpath="./xlf:target/xlf:mrk|./xlf:alt-trans"/>
  <xsd:field xpath="@mid"/>
</xsd:keyref>
</xsd:element>
</xsd:schema>

```

3. Machine Translation Service

3.1 MachineTranslation.wsdl

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<definitions
  name="MachineTranslation"
  targetNamespace="http://tcd.ie/nds/swl10n/ws/contract/machinetranslation"
  xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/machinetranslation"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ns1="http://tcd.ie/nds/swl10n/ws/schema/machinetranslation">
  <types>
    <xs:schema
      targetNamespace="http://tcd.ie/nds/swl10n/ws/schema/machinetranslation"
      xmlns:tns="http://tcd.ie/nds/swl10n/ws/schema/machinetranslation"
      xmlns:xlf="urn:oasis:names:tc:xliff:document:1.2"
      xmlns:mtx="http://tcd.ie/nds/swl10n/ws/schema/machinetranslationXliff"
      xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:import namespace="urn:oasis:names:tc:xliff:document:1.2"
        schemaLocation="xliff-core-1.2-strict.xsd"/>
      <xs:import
        namespace="http://tcd.ie/nds/swl10n/ws/schema/machinetranslationXliff"
        schemaLocation="MachineTranslationXliff.xsd"/>
      <xs:element name="mtXlfRequest" type="tns:mtXlfRequestType"/>
      <xs:element name="mtRequest" type="tns:mtRequestType"/>
      <xs:element name="mtResponse" type="tns:mtResponseType"/>
    </xs:schema>
  </types>

```

```

<xs:element name="OperationFailed" type="xs:string"/>
<xs:complexType name="mtXlfRequestType">
  <xs:sequence>
    <xs:element name="targetLang" type="xs:string" minOccurs="0"/>
    <xs:element ref="xlf:xliff" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="mtRequestType">
  <xs:sequence>
    <xs:element ref="mtx:xliff-mt" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="mtResponseType">
  <xs:sequence>
    <xs:element ref="xlf:xliff" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>
</types>
<message name="msgMachineTranslateXlfRequest">
  <part name="Request" element="ns1:mtXlfRequest"/>
</message>
<message name="msgMachineTranslateRequest">
  <part name="Request" element="ns1:mtRequest"/>
</message>
<message name="msgMachineTranslateResponse">
  <part name="Response" element="ns1:mtResponse"/>
</message>
<message name="OperationFailed">
  <part name="fault" element="ns1:OperationFailed"/>
</message>
<portType name="ptMachineTranslation">
  <operation name="opMachineTranslateXlf">
    <input message="tns:msgMachineTranslateXlfRequest"/>
    <output message="tns:msgMachineTranslateResponse"/>
    <fault message="tns:OperationFailed" name="OperationFailed"/>
  </operation>
  <operation name="opMachineTranslate">
    <input message="tns:msgMachineTranslateRequest"/>
    <output message="tns:msgMachineTranslateResponse"/>
    <fault message="tns:OperationFailed" name="OperationFailed"/>
  </operation>
</portType>
<binding name="Binding1" type="tns:ptMachineTranslation">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
  <operation name="opMachineTranslateXlf">
    <soap:operation soapAction=""/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
    <fault name="OperationFailed">
      <soap:fault name="OperationFailed" use="literal"/>
    </fault>
  </operation>
  <operation name="opMachineTranslate">
    <soap:operation soapAction=""/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
    <fault name="OperationFailed">
      <soap:fault name="OperationFailed" use="literal"/>
    </fault>
  </operation>

```



```

    </operation>
  </binding>
  <service name="svMachineTranslation">
    <port name="machineTranslation" binding="tns:Binding1">
      <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
    </port>
  </service>
</definitions>

```

3.2 Machine TranslationXliff.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://tcd.ie/nds/swl10n/ws/schema/machinetranslationXliff"
  xmlns:tns="http://tcd.ie/nds/swl10n/ws/schema/machinetranslationXliff"
  elementFormDefault="qualified"
  xmlns:xf="urn:oasis:names:tc:xliff:document:1.2"
  xmlns:xml="http://www.w3.org/XML/1998/namespace">
  <xsd:import namespace="urn:oasis:names:tc:xliff:document:1.2" schemaLocation="xliff-
core-1.2-strict.xsd"/>
  <xsd:import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="http://www.w3.org/2001/xml.xsd"/>
  <xsd:element name="xliff-mt">
    <xsd:complexType>
      <xsd:sequence maxOccurs="unbounded">
        <xsd:element ref="tns:file-mt"/>
      </xsd:sequence>
      <xsd:attribute name="version" type="xf:AttrType_Version" use="required"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="file-mt">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="tns:body-mt"/>
      </xsd:sequence>
      <xsd:attribute name="original" type="xsd:string" use="required"/>
      <xsd:attribute name="source-language" type="xsd:language" use="required"/>
      <xsd:attribute name="datatype" type="xf:AttrType_datatype" use="required"/>
      <xsd:attribute name="tool-id" type="xsd:string" use="optional"/>
      <xsd:attribute name="date" type="xsd:dateTime" use="optional"/>
      <xsd:attribute ref="xml:space" use="optional"/>
      <xsd:attribute name="category" type="xsd:string" use="optional"/>
      <xsd:attribute name="target-language" type="xsd:language" use="required"/>
      <xsd:attribute name="product-name" type="xsd:string" use="optional"/>
      <xsd:attribute name="product-version" type="xsd:string" use="optional"/>
      <xsd:attribute name="build-num" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:unique name="U_group_id">
      <xsd:selector xpath="//xf:group"/>
      <xsd:field xpath="@id"/>
    </xsd:unique>
    <xsd:key name="K_unit_id">
      <xsd:selector xpath="//xf:trans-unit|//xf:bin-unit"/>
      <xsd:field xpath="@id"/>
    </xsd:key>
    <xsd:keyref name="KR_unit_id" refer="xf:K_unit_id">
      <xsd:selector xpath="//bpt|//ept|//it|//ph|//g|//x|//bx|//ex|//sub"/>
      <xsd:field xpath="@xid"/>
    </xsd:keyref>
    <xsd:keyref name="KR_alt-trans_tool-id" refer="xf:K_tool-id">
      <xsd:selector xpath="//xf:trans-unit/xf:alt-trans"/>
      <xsd:field xpath="@tool-id"/>
    </xsd:keyref>
    <xsd:unique name="U_context-group_name">
      <xsd:selector xpath="//xf:context-group"/>
      <xsd:field xpath="@name"/>
    </xsd:unique>
  </xsd:element>

```

```

<xsd:element name="body-mt">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:group-mt"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:trans-unit-mt"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
<xsd:element name="group-mt">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:sequence>
        <xsd:element maxOccurs="unbounded" minOccurs="0" ref="xlf:context-group"/>
      </xsd:sequence>
      <xsd:choice maxOccurs="unbounded">
        <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:group-mt"/>
        <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:trans-unit-mt"/>
      </xsd:choice>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:string" use="optional"/>
    <xsd:attribute name="datatype" type="xlf:AttrType_datatype" use="optional"/>
    <xsd:attribute default="default" ref="xml:space" use="optional"/>
    <xsd:attribute name="restype" type="xlf:AttrType_restype" use="optional"/>
    <xsd:attribute name="resname" type="xsd:string" use="optional"/>
    <xsd:attribute name="extradata" type="xsd:string" use="optional"/>
    <xsd:attribute name="extype" type="xsd:string" use="optional"/>
    <xsd:attribute name="help-id" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="menu" type="xsd:string" use="optional"/>
    <xsd:attribute name="menu-option" type="xsd:string" use="optional"/>
    <xsd:attribute name="menu-name" type="xsd:string" use="optional"/>
    <xsd:attribute name="coord" type="xlf:AttrType_Coordinates" use="optional"/>
    <xsd:attribute name="font" type="xsd:string" use="optional"/>
    <xsd:attribute name="css-style" type="xsd:string" use="optional"/>
    <xsd:attribute name="style" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="exstyle" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute default="yes" name="translate" type="xlf:AttrType_YesNo"
use="optional"/>
    <xsd:attribute default="yes" name="reformat" type="xlf:AttrType_reformat"
use="optional"/>
    <xsd:attribute default="pixel" name="size-unit" type="xlf:AttrType_size-unit"
use="optional"/>
    <xsd:attribute name="maxwidth" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="minwidth" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="maxheight" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="minheight" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="maxbytes" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="minbytes" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="charclass" type="xsd:string" use="optional"/>
    <xsd:attribute default="no" name="merged-trans" type="xlf:AttrType_YesNo"
use="optional"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="trans-unit-mt">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xlf:source"/>
      <xsd:element minOccurs="0" ref="xlf:seg-source"/>
      <xsd:element minOccurs="0" ref="xlf:target"/>
      <xsd:choice maxOccurs="unbounded" minOccurs="0">
        <xsd:element ref="xlf:context-group"/>
        <xsd:element ref="xlf:alt-trans"/>
      </xsd:choice>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:string" use="required"/>
    <xsd:attribute name="approved" type="xlf:AttrType_YesNo" use="optional"/>
    <xsd:attribute default="yes" name="translate" type="xlf:AttrType_YesNo"
use="optional"/>
    <xsd:attribute default="yes" name="reformat" type="xlf:AttrType_reformat"

```

```

use="optional"/>
  <xsd:attribute default="default" ref="xml:space" use="optional"/>
  <xsd:attribute name="datatype" type="xlf:AttrType_datatype" use="optional"/>
  <xsd:attribute name="phase-name" type="xsd:string" use="optional"/>
  <xsd:attribute name="restype" type="xlf:AttrType_restype" use="optional"/>
  <xsd:attribute name="resname" type="xsd:string" use="optional"/>
  <xsd:attribute name="extradata" type="xsd:string" use="optional"/>
  <xsd:attribute name="extype" type="xsd:string" use="optional"/>
  <xsd:attribute name="help-id" type="xsd:NMTOKEN" use="optional"/>
  <xsd:attribute name="menu" type="xsd:string" use="optional"/>
  <xsd:attribute name="menu-option" type="xsd:string" use="optional"/>
  <xsd:attribute name="menu-name" type="xsd:string" use="optional"/>
  <xsd:attribute name="coord" type="xlf:AttrType_Coordinates" use="optional"/>
  <xsd:attribute name="font" type="xsd:string" use="optional"/>
  <xsd:attribute name="css-style" type="xsd:string" use="optional"/>
  <xsd:attribute name="style" type="xsd:NMTOKEN" use="optional"/>
  <xsd:attribute name="exstyle" type="xsd:NMTOKEN" use="optional"/>
  <xsd:attribute default="pixel" name="size-unit" type="xlf:AttrType_size-unit"
use="optional"/>
  <xsd:attribute name="maxwidth" type="xsd:NMTOKEN" use="optional"/>
  <xsd:attribute name="minwidth" type="xsd:NMTOKEN" use="optional"/>
  <xsd:attribute name="maxheight" type="xsd:NMTOKEN" use="optional"/>
  <xsd:attribute name="minheight" type="xsd:NMTOKEN" use="optional"/>
  <xsd:attribute name="maxbytes" type="xsd:NMTOKEN" use="optional"/>
  <xsd:attribute name="minbytes" type="xsd:NMTOKEN" use="optional"/>
  <xsd:attribute name="charclass" type="xsd:string" use="optional"/>
  <xsd:attribute default="yes" name="merged-trans" type="xlf:AttrType_YesNo"
use="optional"/>
</xsd:complexType>
<xsd:unique name="U_tu_segsrc_mid">
  <xsd:selector xpath="./xlf:seg-source/xlf:mrk"/>
  <xsd:field xpath="@mid"/>
</xsd:unique>
<xsd:keyref name="KR_tu_segsrc_mid" refer="xlf:U_tu_segsrc_mid">
  <xsd:selector xpath="./xlf:target/xlf:mrk|xlf:alt-trans"/>
  <xsd:field xpath="@mid"/>
</xsd:keyref>
</xsd:element>
</xsd:schema>

```

4. Parsing Service

4.1 Parsing.wSDL

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<definitions
  name="Parsing"
  targetNamespace="http://tcd.ie/nds/swl10n/ws/contract/parsing"
  xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/parsing"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ns1="http://tcd.ie/nds/swl10n/ws/schema/parsing">
  <types>
    <xsd:schema
      targetNamespace="http://tcd.ie/nds/swl10n/ws/schema/parsing"
      xmlns:tns="http://tcd.ie/nds/swl10n/ws/schema/parsing"
      xmlns:xlf="urn:oasis:names:tc:xliff:document:1.2"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <xsd:import namespace="urn:oasis:names:tc:xliff:document:1.2"
schemaLocation="xlfiff-core-1.2-strict.xsd"/>
      <xsd:element name="supportedDataTypesRequest"
type="tns:supportedDataTypesRequestType"/>
      <xsd:element name="supportedDataTypesResponse"
type="tns:supportedDataTypesResponseType"/>
      <xsd:element name="parseRequest" type="tns:parseRequestType"/>

```

```

<xsd:element name="parseResponse" type="tns:parseResponseType"/>
<xsd:element name="UnsupportedDataType" type="xsd:string"/>
<xsd:element name="OperationFailed" type="xsd:string"/>
<xsd:complexType name="supportedDataTypesRequestType">
  <xsd:sequence/>
</xsd:complexType>
<xsd:complexType name="supportedDataTypesResponseType">
  <xsd:sequence>
    <xsd:element name="dataType" type="xsd:string" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="parseRequestType">
  <xsd:sequence>
    <xsd:element name="returnSingleXliff" type="xsd:boolean"/>
    <xsd:element name="fileInfo" type="tns:fileInfoType" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="fileInfoType">
  <xsd:sequence>
    <xsd:element name="dataType" type="xsd:string" minOccurs="0"/>
    <xsd:element name="fileLanguage" type="xsd:string" minOccurs="0"/>
    <xsd:element name="fileName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="fileContent" type="xsd:base64Binary" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="parseResponseType">
  <xsd:sequence>
    <xsd:element ref="xlf:xliff" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>
</types>
<message name="msgGetSupportedDataTypesRequest">
  <part name="Request" element="ns1:supportedDataTypesRequest"/>
</message>
<message name="msgGetSupportedDataTypesResponse">
  <part name="Response" element="ns1:supportedDataTypesResponse"/>
</message>
<message name="msgParseFilesRequest">
  <part name="Request" element="ns1:parseRequest"/>
</message>
<message name="msgParseFilesResponse">
  <part name="Response" element="ns1:parseResponse"/>
</message>
<message name="UnsupportedDataType">
  <part name="fault" element="ns1:UnsupportedDataType"/>
</message>
<message name="OperationFailed">
  <part name="fault" element="ns1:OperationFailed"/>
</message>
<portType name="ptParsing">
  <operation name="opGetSupportedDataTypes">
    <input message="tns:msgGetSupportedDataTypesRequest"/>
    <output message="tns:msgGetSupportedDataTypesResponse"/>
    <fault message="tns:OperationFailed" name="OperationFailed"/>
  </operation>
  <operation name="opParseFiles">
    <input message="tns:msgParseFilesRequest"/>
    <output message="tns:msgParseFilesResponse"/>
    <fault message="tns:UnsupportedDataType" name="UnsupportedDataType"/>
    <fault message="tns:OperationFailed" name="OperationFailed"/>
  </operation>
</portType>
<binding name="Binding1" type="tns:ptParsing">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
  <operation name="opGetSupportedDataTypes">

```

```

    <soap:operation soapAction=""/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
    <fault name="OperationFailed">
      <soap:fault name="OperationFailed" use="literal"/>
    </fault>
  </operation>
</operation>
<operation name="opParseFiles">
  <soap:operation soapAction=""/>
  <input>
    <soap:body use="literal"/>
  </input>
  <output>
    <soap:body use="literal"/>
  </output>
  <fault name="UnsupportedDataType">
    <soap:fault name="UnsupportedDataType" use="literal"/>
  </fault>
  <fault name="OperationFailed">
    <soap:fault name="OperationFailed" use="literal"/>
  </fault>
</operation>
</binding>
<service name="svParsing">
  <port name="parsing" binding="tns:Binding1">
    <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
  </port>
</service>
</definitions>

```

5. Pseudo Translation Service

5.1 PseudoTranslation.wsdl

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<definitions
  name="PseudoTranslation"
  targetNamespace="http://tcd.ie/nds/swl10n/ws/contract/pseudoTranslation"
  xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/pseudoTranslation"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ns1="http://tcd.ie/nds/swl10n/ws/schema/pseudoTranslation">
  <types>
    <xs:schema
      targetNamespace="http://tcd.ie/nds/swl10n/ws/schema/pseudoTranslation"
      xmlns:tns="http://tcd.ie/nds/swl10n/ws/schema/pseudoTranslation"
      xmlns:xlf="urn:oasis:names:tc:xliff:document:1.2"
      xmlns:ptx="http://tcd.ie/nds/swl10n/ws/schema/pseudoTranslationXliff"
      xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:import namespace="urn:oasis:names:tc:xliff:document:1.2"
        schemaLocation="xliff-core-1.2-strict.xsd"/>
      <xs:import namespace="http://tcd.ie/nds/swl10n/ws/schema/pseudoTranslationXliff"
        schemaLocation="PseudoTranslationXliff.xsd"/>
      <xs:element name="pseudoTranslateXlfRequest"
        type="tns:pseudoTranslateXlfRequestType"/>
      <xs:element name="pseudoTranslateRequest"
        type="tns:pseudoTranslateRequestType"/>
      <xs:element name="pseudoTranslateResponse"
        type="tns:pseudoTranslateResponseType"/>
      <xs:element name="OperationFailed" type="xs:string"/>
      <xs:complexType name="pseudoTranslateXlfRequestType">

```

```

<xs:sequence>
  <xs:element name="targetLang" type="xs:string" minOccurs="0"/>
  <xs:element name="usePrefix" type="xs:boolean"/>
  <xs:element name="prefix" type="xs:string" minOccurs="0"/>
  <xs:element name="useSuffix" type="xs:boolean"/>
  <xs:element name="suffix" type="xs:string" minOccurs="0"/>
  <xs:element name="excludeStringsOneWord" type="xs:boolean"/>
  <xs:element name="excludeStringsLessThan" type="xs:boolean"/>
  <xs:element name="excludeStringsCamelCase" type="xs:boolean"/>
  <xs:element name="excludeStringAllCaps" type="xs:boolean"/>
  <xs:element name="excludeStringsNumeric" type="xs:boolean"/>
  <xs:element name="excludeTranslated" type="xs:boolean"/>
  <xs:element name="expansionRatio" type="xs:int"/>
  <xs:element name="lessThan" type="xs:int"/>
  <xs:element ref="xlf:xliff" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="pseudoTranslateRequestType">
  <xs:sequence>
    <xs:element name="targetLang" type="xs:string" minOccurs="0"/>
    <xs:element name="usePrefix" type="xs:boolean"/>
    <xs:element name="prefix" type="xs:string" minOccurs="0"/>
    <xs:element name="useSuffix" type="xs:boolean"/>
    <xs:element name="suffix" type="xs:string" minOccurs="0"/>
    <xs:element name="excludeStringsOneWord" type="xs:boolean"/>
    <xs:element name="excludeStringsLessThan" type="xs:boolean"/>
    <xs:element name="excludeStringsCamelCase" type="xs:boolean"/>
    <xs:element name="excludeStringAllCaps" type="xs:boolean"/>
    <xs:element name="excludeStringsNumeric" type="xs:boolean"/>
    <xs:element name="expansionRatio" type="xs:int"/>
    <xs:element name="lessThan" type="xs:int"/>
    <xs:element ref="ptx:xliff-pseudo" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="pseudoTranslateResponseType">
  <xs:sequence>
    <xs:element ref="xlf:xliff" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>
</types>
<message name="msgPseudoTranslateXlfRequest">
  <part name="Request" element="ns1:pseudoTranslateXlfRequest"/>
</message>
<message name="msgPseudoTranslateRequest">
  <part name="Request" element="ns1:pseudoTranslateRequest"/>
</message>
<message name="msgPseudoTranslateResponse">
  <part name="Response" element="ns1:pseudoTranslateResponse"/>
</message>
<message name="OperationFailed">
  <part name="fault" element="ns1:OperationFailed"/>
</message>
<portType name="ptPseudoTranslation">
  <operation name="opPseudoTranslateXlf">
    <input message="tns:msgPseudoTranslateXlfRequest"/>
    <output message="tns:msgPseudoTranslateResponse"/>
    <fault message="tns:OperationFailed" name="OperationFailed"/>
  </operation>
  <operation name="opPseudoTranslate">
    <input message="tns:msgPseudoTranslateRequest"/>
    <output message="tns:msgPseudoTranslateResponse"/>
    <fault message="tns:OperationFailed" name="OperationFailed"/>
  </operation>
</portType>
<binding name="Binding1" type="tns:ptPseudoTranslation">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
  <operation name="opPseudoTranslateXlf">

```

```

    <soap:operation soapAction=""/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
    <fault name="OperationFailed">
      <soap:fault name="OperationFailed" use="literal"/>
    </fault>
  </operation>
</operation name="opPseudoTranslate">
  <soap:operation soapAction=""/>
  <input>
    <soap:body use="literal"/>
  </input>
  <output>
    <soap:body use="literal"/>
  </output>
  <fault name="OperationFailed">
    <soap:fault name="OperationFailed" use="literal"/>
  </fault>
</operation>
</binding>
<service name="svPseudoTranslation">
  <port name="pseudoTranslation" binding="tns:Binding1">
    <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
  </port>
</service>
</definitions>

```

5.2 PseudoTranslation.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://tcd.ie/nds/swl10n/ws/schema/pseudoTranslationXliff"
  xmlns:tns="http://tcd.ie/nds/swl10n/ws/schema/pseudoTranslationXliff"
  elementFormDefault="qualified"
  xmlns:xf="urn:oasis:names:tc:xliff:document:1.2"
  xmlns:xml="http://www.w3.org/XML/1998/namespace">
  <xsd:import namespace="urn:oasis:names:tc:xliff:document:1.2" schemaLocation="xliff-
core-1.2-strict.xsd"/>
  <xsd:import namespace="http://www.w3.org/XML/1998/namespace"
  schemaLocation="http://www.w3.org/2001/xml.xsd"/>
  <xsd:element name="xliff-pseudo">
    <xsd:complexType>
      <xsd:sequence maxOccurs="unbounded">
        <xsd:element ref="tns:file-pseudo"/>
      </xsd:sequence>
      <xsd:attribute name="version" type="xf:AttrType_Version" use="required"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="file-pseudo">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="tns:body-pseudo"/>
      </xsd:sequence>
      <xsd:attribute name="original" type="xsd:string" use="required"/>
      <xsd:attribute name="source-language" type="xsd:language" use="required"/>
      <xsd:attribute name="datatype" type="xf:AttrType_datatype" use="required"/>
      <xsd:attribute ref="xml:space" use="optional"/>
      <xsd:attribute name="category" type="xsd:string" use="optional"/>
      <xsd:attribute name="target-language" type="xsd:language" use="optional"/>
      <xsd:attribute name="product-name" type="xsd:string" use="optional"/>
      <xsd:attribute name="product-version" type="xsd:string" use="optional"/>
      <xsd:attribute name="build-num" type="xsd:string" use="optional"/>
    </xsd:complexType>
  <xsd:unique name="U_group_id">

```

```

        <xsd:selector xpath="//xlf:group"/>
        <xsd:field xpath="@id"/>
    </xsd:unique>
    <xsd:key name="K_unit_id">
        <xsd:selector xpath="//xlf:trans-unit|//xlf:bin-unit"/>
        <xsd:field xpath="@id"/>
    </xsd:key>
    <xsd:keyref name="KR_unit_id" refer="xlf:K_unit_id">
        <xsd:selector xpath="//bpt|//ept|//it|//ph|//g|//x|//bx|//ex|//sub"/>
        <xsd:field xpath="@xid"/>
    </xsd:keyref>
</xsd:element>
<xsd:element name="body-pseudo">
    <xsd:complexType>
        <xsd:choice maxOccurs="unbounded" minOccurs="0">
            <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:group-pseudo"/>
            <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:trans-unit-pseudo"/>
        </xsd:choice>
    </xsd:complexType>
</xsd:element>
<xsd:element name="group-pseudo">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:choice maxOccurs="unbounded">
                <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:group-pseudo"/>
                <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:trans-unit-
pseudo"/>
            </xsd:choice>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:string" use="optional"/>
        <xsd:attribute name="datatype" type="xlf:AttrType_datatype" use="optional"/>
        <xsd:attribute default="default" ref="xml:space" use="optional"/>
        <xsd:attribute name="restype" type="xlf:AttrType_restype" use="optional"/>
        <xsd:attribute name="resname" type="xsd:string" use="optional"/>
        <xsd:attribute name="extradata" type="xsd:string" use="optional"/>
        <xsd:attribute name="extype" type="xsd:string" use="optional"/>
        <xsd:attribute name="help-id" type="xsd:NMTOKEN" use="optional"/>
        <xsd:attribute name="menu" type="xsd:string" use="optional"/>
        <xsd:attribute name="menu-option" type="xsd:string" use="optional"/>
        <xsd:attribute name="menu-name" type="xsd:string" use="optional"/>
        <xsd:attribute name="coord" type="xlf:AttrType_Coordinates" use="optional"/>
        <xsd:attribute name="font" type="xsd:string" use="optional"/>
        <xsd:attribute name="css-style" type="xsd:string" use="optional"/>
        <xsd:attribute name="style" type="xsd:NMTOKEN" use="optional"/>
        <xsd:attribute name="exstyle" type="xsd:NMTOKEN" use="optional"/>
        <xsd:attribute default="yes" name="translate" type="xlf:AttrType_YesNo"
use="optional"/>
        <xsd:attribute default="yes" name="reformat" type="xlf:AttrType_reformat"
use="optional"/>
        <xsd:attribute default="pixel" name="size-unit" type="xlf:AttrType_size-unit"
use="optional"/>
        <xsd:attribute name="maxwidth" type="xsd:NMTOKEN" use="optional"/>
        <xsd:attribute name="minwidth" type="xsd:NMTOKEN" use="optional"/>
        <xsd:attribute name="maxheight" type="xsd:NMTOKEN" use="optional"/>
        <xsd:attribute name="minheight" type="xsd:NMTOKEN" use="optional"/>
        <xsd:attribute name="maxbytes" type="xsd:NMTOKEN" use="optional"/>
        <xsd:attribute name="minbytes" type="xsd:NMTOKEN" use="optional"/>
        <xsd:attribute name="charclass" type="xsd:string" use="optional"/>
        <xsd:attribute default="no" name="merged-trans" type="xlf:AttrType_YesNo"
use="optional"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="trans-unit-pseudo">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="xlf:source"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:string" use="required"/>

```



```

    <xsd:attribute name="approved" type="xlf:AttrType_YesNo" use="optional"/>
    <xsd:attribute default="yes" name="translate" type="xlf:AttrType_YesNo"
use="optional"/>
    <xsd:attribute default="yes" name="reformat" type="xlf:AttrType_reformat"
use="optional"/>
    <xsd:attribute default="default" ref="xml:space" use="optional"/>
    <xsd:attribute name="datatype" type="xlf:AttrType_datatype" use="optional"/>
    <xsd:attribute name="phase-name" type="xsd:string" use="optional"/>
    <xsd:attribute name="restype" type="xlf:AttrType_restype" use="optional"/>
    <xsd:attribute name="resname" type="xsd:string" use="optional"/>
    <xsd:attribute name="extradata" type="xsd:string" use="optional"/>
    <xsd:attribute name="extype" type="xsd:string" use="optional"/>
    <xsd:attribute name="help-id" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="menu" type="xsd:string" use="optional"/>
    <xsd:attribute name="menu-option" type="xsd:string" use="optional"/>
    <xsd:attribute name="menu-name" type="xsd:string" use="optional"/>
    <xsd:attribute name="coord" type="xlf:AttrType_Coordinates" use="optional"/>
    <xsd:attribute name="font" type="xsd:string" use="optional"/>
    <xsd:attribute name="css-style" type="xsd:string" use="optional"/>
    <xsd:attribute name="style" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="exstyle" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute default="pixel" name="size-unit" type="xlf:AttrType_size-unit"
use="optional"/>
    <xsd:attribute name="maxwidth" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="minwidth" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="maxheight" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="minheight" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="maxbytes" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="minbytes" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="charclass" type="xsd:string" use="optional"/>
    <xsd:attribute default="yes" name="merged-trans" type="xlf:AttrType_YesNo"
use="optional"/>
  </xsd:complexType>
</xsd:element>
</xsd:schema>

```

6. QaCheck Service

6.1 QaCheck.wsdl

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<definitions
  name="QaCheck"
  targetNamespace="http://tcd.ie/nds/swl10n/ws/contract/qacheck"
  xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/qacheck"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ns1="http://tcd.ie/nds/swl10n/ws/schema/qacheck">
  <types>
    <xs:schema
      targetNamespace="http://tcd.ie/nds/swl10n/ws/schema/qacheck"
      xmlns:tns="http://tcd.ie/nds/swl10n/ws/schema/qacheck"
      xmlns:xlf="urn:oasis:names:tc:xliff:document:1.2"
      xmlns:qax="http://tcd.ie/nds/swl10n/ws/schema/qacheckXliff"
      xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:import namespace="urn:oasis:names:tc:xliff:document:1.2"
schemaLocation="xliff-core-1.2-strict.xsd"/>
      <xs:import namespace="http://tcd.ie/nds/swl10n/ws/schema/qacheckXliff"
schemaLocation="QaXliff.xsd"/>
      <xs:element name="qaCheckXlfRequest" type="tns:qaCheckXlfRequestType"/>
      <xs:element name="qaCheckRequest" type="tns:qaCheckRequestType"/>
      <xs:element name="qaCheckResponse" type="tns:qaCheckResponseType"/>
      <xs:element name="qaIssue" type="tns:qaIssueType"/>
      <xs:element name="OperationFailed" type="xs:string"/>
      <xs:complexType name="qaCheckXlfRequestType">

```

```

        <xs:sequence>
            <xs:element ref="xlf:xliff" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="qaCheckRequestType">
        <xs:sequence>
            <xs:element ref="qax:xliff-qa" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="qaCheckResponseType">
        <xs:sequence>
            <xs:element ref="xlf:xliff" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="qaIssueType">
        <xs:attribute name="type" type="xs:string" use="required"/>
        <xs:attribute name="resolved" type="xlf:AttrType_YesNo" use="required"/>
        <xs:attribute name="severity" type="xs:string" use="optional"/>
        <xs:attribute name="defect-id" type="xs:string" use="optional"/>
    </xs:complexType>
</xs:schema>
</types>
<message name="msgQaCheckXlfRequest">
    <part name="Request" element="ns1:qaCheckXlfRequest"/>
</message>
<message name="msgQaCheckRequest">
    <part name="Request" element="ns1:qaCheckRequest"/>
</message>
<message name="msgQaCheckResponse">
    <part name="Response" element="ns1:qaCheckResponse"/>
</message>
<message name="OperationFailed">
    <part name="fault" element="ns1:OperationFailed"/>
</message>
<portType name="ptQaCheck">
    <operation name="opQaCheckXlf">
        <input message="tns:msgQaCheckXlfRequest"/>
        <output message="tns:msgQaCheckResponse"/>
        <fault message="tns:OperationFailed" name="OperationFailed"/>
    </operation>
    <operation name="opQaCheck">
        <input message="tns:msgQaCheckRequest"/>
        <output message="tns:msgQaCheckResponse"/>
        <fault message="tns:OperationFailed" name="OperationFailed"/>
    </operation>
</portType>
<binding name="Binding1" type="tns:ptQaCheck">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <operation name="opQaCheckXlf">
        <soap:operation soapAction=""/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
        <fault name="OperationFailed">
            <soap:fault name="OperationFailed" use="literal"/>
        </fault>
    </operation>
    <operation name="opQaCheck">
        <soap:operation soapAction=""/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
    </operation>

```

```

        <fault name="OperationFailed">
            <soap:fault name="OperationFailed" use="literal"/>
        </fault>
    </operation>
</binding>
<service name="svQaCheck">
    <port name="qacheck" binding="tns:Binding1">
        <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
    </port>
</service>
</definitions>

```

6.2 QaXliff.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://tcd.ie/nds/swl10n/ws/schema/qacheckXliff"
    xmlns:tns="http://tcd.ie/nds/swl10n/ws/schema/qacheckXliff"
    elementFormDefault="qualified"
    xmlns:xf="urn:oasis:names:tc:xliff:document:1.2"
    xmlns:xml="http://www.w3.org/XML/1998/namespace"
>
    <xsd:import namespace="urn:oasis:names:tc:xliff:document:1.2" schemaLocation="xliff-
core-1.2-strict.xsd"/>
    <xsd:import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="http://www.w3.org/2001/xml.xsd"/>
    <xsd:element name="xliff-qa">
        <xsd:complexType>
            <xsd:sequence maxOccurs="unbounded">
                <xsd:element ref="tns:file-qa"/>
            </xsd:sequence>
            <xsd:attribute name="version" type="xf:AttrType_Version" use="required"/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="file-qa">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="tns:body-qa"/>
            </xsd:sequence>
            <xsd:attribute name="original" type="xsd:string" use="required"/>
            <xsd:attribute name="source-language" type="xsd:language" use="required"/>
            <xsd:attribute name="datatype" type="xf:AttrType_datatype" use="required"/>
            <xsd:attribute ref="xml:space" use="optional"/>
            <xsd:attribute name="category" type="xsd:string" use="optional"/>
            <xsd:attribute name="target-language" type="xsd:language" use="optional"/>
            <xsd:attribute name="product-name" type="xsd:string" use="optional"/>
            <xsd:attribute name="product-version" type="xsd:string" use="optional"/>
            <xsd:attribute name="build-num" type="xsd:string" use="optional"/>
        </xsd:complexType>
        <xsd:unique name="U_group_id">
            <xsd:selector xpath="//xf:group"/>
            <xsd:field xpath="@id"/>
        </xsd:unique>
        <xsd:key name="K_unit_id">
            <xsd:selector xpath="//xf:trans-unit|//xf:bin-unit"/>
            <xsd:field xpath="@id"/>
        </xsd:key>
        <xsd:keyref name="KR_unit_id" refer="xf:K_unit_id">
            <xsd:selector xpath="//bpt|//ept|//it|//ph|//g|//x|//bx|//ex|//sub"/>
            <xsd:field xpath="@xid"/>
        </xsd:keyref>
    </xsd:element>
    <xsd:element name="body-qa">
        <xsd:complexType>
            <xsd:choice maxOccurs="unbounded" minOccurs="0">
                <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:group-qa"/>
                <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:trans-unit-qa"/>
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>

```

```

</xsd:complexType>
</xsd:element>
<xsd:element name="group-qa">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:choice maxOccurs="unbounded">
        <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:group-qa"/>
        <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:trans-unit-qa"/>
      </xsd:choice>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:string" use="optional"/>
    <xsd:attribute name="datatype" type="xlf:AttrType_datatype" use="optional"/>
    <xsd:attribute default="default" ref="xml:space" use="optional"/>
    <xsd:attribute name="restype" type="xlf:AttrType_restype" use="optional"/>
    <xsd:attribute name="resname" type="xsd:string" use="optional"/>
    <xsd:attribute name="extradata" type="xsd:string" use="optional"/>
    <xsd:attribute name="extype" type="xsd:string" use="optional"/>
    <xsd:attribute name="help-id" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="menu" type="xsd:string" use="optional"/>
    <xsd:attribute name="menu-option" type="xsd:string" use="optional"/>
    <xsd:attribute name="menu-name" type="xsd:string" use="optional"/>
    <xsd:attribute name="coord" type="xlf:AttrType_Coordinates" use="optional"/>
    <xsd:attribute name="font" type="xsd:string" use="optional"/>
    <xsd:attribute name="css-style" type="xsd:string" use="optional"/>
    <xsd:attribute name="style" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="exstyle" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute default="yes" name="translate" type="xlf:AttrType_YesNo"
use="optional"/>
    <xsd:attribute default="yes" name="reformat" type="xlf:AttrType_reformat"
use="optional"/>
    <xsd:attribute default="pixel" name="size-unit" type="xlf:AttrType_size-unit"
use="optional"/>
    <xsd:attribute name="maxwidth" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="minwidth" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="maxheight" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="minheight" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="maxbytes" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="minbytes" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="charclass" type="xsd:string" use="optional"/>
    <xsd:attribute default="no" name="merged-trans" type="xlf:AttrType_YesNo"
use="optional"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="trans-unit-qa">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xlf:source"/>
      <xsd:element minOccurs="0" ref="xlf:target"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:string" use="required"/>
    <xsd:attribute name="approved" type="xlf:AttrType_YesNo" use="optional"/>
    <xsd:attribute default="yes" name="translate" type="xlf:AttrType_YesNo"
use="optional"/>
    <xsd:attribute default="yes" name="reformat" type="xlf:AttrType_reformat"
use="optional"/>
    <xsd:attribute default="default" ref="xml:space" use="optional"/>
    <xsd:attribute name="datatype" type="xlf:AttrType_datatype" use="optional"/>
    <xsd:attribute name="phase-name" type="xsd:string" use="optional"/>
    <xsd:attribute name="restype" type="xlf:AttrType_restype" use="optional"/>
    <xsd:attribute name="resname" type="xsd:string" use="optional"/>
    <xsd:attribute name="extradata" type="xsd:string" use="optional"/>
    <xsd:attribute name="extype" type="xsd:string" use="optional"/>
    <xsd:attribute name="help-id" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="menu" type="xsd:string" use="optional"/>
    <xsd:attribute name="menu-option" type="xsd:string" use="optional"/>
    <xsd:attribute name="menu-name" type="xsd:string" use="optional"/>
    <xsd:attribute name="coord" type="xlf:AttrType_Coordinates" use="optional"/>
    <xsd:attribute name="font" type="xsd:string" use="optional"/>

```

```

    <xsd:attribute name="css-style" type="xsd:string" use="optional"/>
    <xsd:attribute name="style" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="exstyle" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute default="pixel" name="size-unit" type="xlf:AttrType_size-unit"
use="optional"/>
    <xsd:attribute name="maxwidth" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="minwidth" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="maxheight" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="minheight" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="maxbytes" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="minbytes" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="charclass" type="xsd:string" use="optional"/>
    <xsd:attribute default="yes" name="merged-trans" type="xlf:AttrType_YesNo"
use="optional"/>
  </xsd:complexType>
</xsd:element>
</xsd:schema>

```

7. Reporting Service

7.1 Reporting.wSDL

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<definitions
  name="Reporting"
  targetNamespace="http://tcd.ie/nds/swl10n/ws/contract/reporting"
  xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/reporting"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ns1="http://tcd.ie/nds/swl10n/ws/schema/reporting">
  <types>
    <xs:schema
      targetNamespace="http://tcd.ie/nds/swl10n/ws/schema/reporting"
      xmlns:tns="http://tcd.ie/nds/swl10n/ws/schema/reporting"
      xmlns:xlf="urn:oasis:names:tc:xliff:document:1.2"
      xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:import namespace="urn:oasis:names:tc:xliff:document:1.2"
schemaLocation="xliff-core-1.2-strict.xsd"/>
      <xs:element name="reportRequest" type="tns:reportRequestType"/>
      <xs:element name="reportResponse" type="tns:reportResponseType"/>
      <xs:element name="OperationFailed" type="xs:string"/>
      <xs:complexType name="reportRequestType">
        <xs:sequence>
          <xs:element ref="xlf:xliff" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="reportResponseType">
        <xs:sequence>
          <xs:element name="totalWordcount" type="xs:int"/>
          <xs:element name="totalTranslated" type="xs:int"/>
          <xs:element name="totalUntranslated" type="xs:int"/>
          <xs:element name="numberOfFiles" type="xs:int"/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
  </types>
  <message name="msgReportRequest">
    <part name="Request" element="ns1:reportRequest"/>
  </message>
  <message name="msgReportResponse">
    <part name="Response" element="ns1:reportResponse"/>
  </message>
  <message name="OperationFailed">
    <part name="fault" element="ns1:OperationFailed"/>
  </message>

```

```

<portType name="ptReporting">
  <operation name="opGetReport">
    <input message="tns:msgReportRequest"/>
    <output message="tns:msgReportResponse"/>
    <fault message="tns:OperationFailed" name="OperationFailed"/>
  </operation>
</portType>
<binding name="Binding1" type="tns:ptReporting">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
  <operation name="opGetReport">
    <soap:operation soapAction=""/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
    <fault name="OperationFailed">
      <soap:fault name="OperationFailed" use="literal"/>
    </fault>
  </operation>
</binding>
<service name="svReporting">
  <port name="reporting" binding="tns:Binding1">
    <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
  </port>
</service>
</definitions>

```

8. Review Service

8.1 Review.wsdl

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<definitions
  name="Review"
  targetNamespace="http://tcd.ie/nds/swl10n/ws/contract/review"
  xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/review"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ns1="http://tcd.ie/nds/swl10n/ws/schema/review">
  <types>
    <xs:schema
      targetNamespace="http://tcd.ie/nds/swl10n/ws/schema/review"
      xmlns:tns="http://tcd.ie/nds/swl10n/ws/schema/review"
      xmlns:xf="urn:oasis:names:tc:xliff:document:1.2"
      xmlns:rex="http://tcd.ie/nds/swl10n/ws/schema/reviewXliff"
      xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:import namespace="urn:oasis:names:tc:xliff:document:1.2"
        schemaLocation="xliff-core-1.2-strict.xsd"/>
      <xs:import namespace="http://tcd.ie/nds/swl10n/ws/schema/reviewXliff"
        schemaLocation="ReviewXliff.xsd"/>
      <xs:element name="submitForReviewXlfRequest"
        type="tns:submitForReviewXlfRequestType"/>
      <xs:element name="submitForReviewRequest"
        type="tns:submitForReviewRequestType"/>
      <xs:element name="submitForReviewResponse"
        type="tns:submitForReviewResponseType"/>
      <xs:element name="getReviewRequest" type="tns:getReviewRequestType"/>
      <xs:element name="getReviewResponse" type="tns:getReviewResponseType"/>
      <xs:element name="OperationFailed" type="xs:string"/>
      <xs:complexType name="submitForReviewXlfRequestType">
        <xs:sequence>
          <xs:element name="targetLang" type="xs:string" minOccurs="0"/>
          <xs:element ref="xf:xliff" minOccurs="0"/>

```

```

        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="submitForReviewRequestType">
        <xs:sequence>
            <xs:element ref="rex:xliff-rev" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="submitForReviewResponseType">
        <xs:sequence>
            <xs:element name="id" type="xs:string" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="getReviewRequestType">
        <xs:sequence>
            <xs:element name="id" type="xs:string" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="getReviewResponseType">
        <xs:sequence>
            <xs:element ref="xlf:xliff" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
</types>
<message name="msgSubmitForReviewXlfRequest">
    <part name="Request" element="ns1:submitForReviewXlfRequest"/>
</message>
<message name="msgSubmitForReviewRequest">
    <part name="Request" element="ns1:submitForReviewRequest"/>
</message>
<message name="msgSubmitForReviewResponse">
    <part name="Response" element="ns1:submitForReviewResponse"/>
</message>
<message name="msgGetReviewRequest">
    <part name="Request" element="ns1:getReviewRequest"/>
</message>
<message name="msgGetReviewResponse">
    <part name="Response" element="ns1:getReviewResponse"/>
</message>
<message name="OperationFailed">
    <part name="fault" element="ns1:OperationFailed"/>
</message>
<portType name="ptReview">
    <operation name="opSubmitForReviewXlf">
        <input message="tns:msgSubmitForReviewXlfRequest"/>
        <output message="tns:msgSubmitForReviewResponse"/>
        <fault message="tns:OperationFailed" name="OperationFailed"/>
    </operation>
    <operation name="opSubmitForReview">
        <input message="tns:msgSubmitForReviewRequest"/>
        <output message="tns:msgSubmitForReviewResponse"/>
        <fault message="tns:OperationFailed" name="OperationFailed"/>
    </operation>
    <operation name="opGetReview">
        <input message="tns:msgGetReviewRequest"/>
        <output message="tns:msgGetReviewResponse"/>
        <fault message="tns:OperationFailed" name="OperationFailed"/>
    </operation>
</portType>
<binding name="Binding1" type="tns:ptReview">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <operation name="opSubmitForReviewXlf">
        <soap:operation soapAction=""/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
    </operation>

```

```

        </output>
        <fault name="OperationFailed">
            <soap:fault name="OperationFailed" use="literal"/>
        </fault>
    </operation>
</operation>
<operation name="opSubmitForReview">
    <soap:operation soapAction=""/>
    <input>
        <soap:body use="literal"/>
    </input>
    <output>
        <soap:body use="literal"/>
    </output>
    <fault name="OperationFailed">
        <soap:fault name="OperationFailed" use="literal"/>
    </fault>
</operation>
<operation name="opGetReview">
    <soap:operation soapAction=""/>
    <input>
        <soap:body use="literal"/>
    </input>
    <output>
        <soap:body use="literal"/>
    </output>
    <fault name="OperationFailed">
        <soap:fault name="OperationFailed" use="literal"/>
    </fault>
</operation>
</binding>
<service name="svReview">
    <port name="review" binding="tns:Binding1">
        <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
    </port>
</service>
</definitions>

```

8.2 ReviewXliff.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://tcd.ie/nds/swl10n/ws/schema/reviewXliff"
    xmlns:tns="http://tcd.ie/nds/swl10n/ws/schema/reviewXliff"
    elementFormDefault="qualified"
    xmlns:xf="urn:oasis:names:tc:xliff:document:1.2"
    xmlns:xml="http://www.w3.org/XML/1998/namespace">
    <xsd:import namespace="urn:oasis:names:tc:xliff:document:1.2" schemaLocation="xliff-
core-1.2-strict.xsd"/>
    <xsd:import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="http://www.w3.org/2001/xml.xsd"/>
    <xsd:element name="xliff-rev">
        <xsd:complexType>
            <xsd:sequence maxOccurs="unbounded">
                <xsd:element ref="tns:file-rev"/>
            </xsd:sequence>
            <xsd:attribute name="version" type="xf:AttrType_Version" use="required"/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="file-rev">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element minOccurs="0" ref="tns:header-rev"/>
                <xsd:element ref="tns:body-rev"/>
            </xsd:sequence>
            <xsd:attribute name="original" type="xsd:string" use="required"/>
            <xsd:attribute name="source-language" type="xsd:language" use="required"/>
            <xsd:attribute name="datatype" type="xf:AttrType_datatype" use="required"/>
            <xsd:attribute name="tool-id" type="xsd:string" use="optional"/>
        </xsd:complexType>
    </xsd:element>

```



```

<xsd:attribute name="date" type="xsd:dateTime" use="optional"/>
<xsd:attribute ref="xml:space" use="optional"/>
<xsd:attribute name="category" type="xsd:string" use="optional"/>
<xsd:attribute name="target-language" type="xsd:language" use="required"/>
<xsd:attribute name="product-name" type="xsd:string" use="optional"/>
<xsd:attribute name="product-version" type="xsd:string" use="optional"/>
<xsd:attribute name="build-num" type="xsd:string" use="optional"/>
</xsd:complexType>
<xsd:unique name="U_group_id">
  <xsd:selector xpath="//xlf:group"/>
  <xsd:field xpath="@id"/>
</xsd:unique>
<xsd:key name="K_unit_id">
  <xsd:selector xpath="//xlf:trans-unit|//xlf:bin-unit"/>
  <xsd:field xpath="@id"/>
</xsd:key>
<xsd:keyref name="KR_unit_id" refer="xlf:K_unit_id">
  <xsd:selector xpath="//bpt|//ept|//it|//ph|//g|//x|//bx|//ex|//sub"/>
  <xsd:field xpath="@xid"/>
</xsd:keyref>
<xsd:key name="K_tool-id">
  <xsd:selector xpath="xlf:header/xlf:tool"/>
  <xsd:field xpath="@tool-id"/>
</xsd:key>
<xsd:keyref name="KR_file_tool-id" refer="xlf:K_tool-id">
  <xsd:selector xpath="."/>
  <xsd:field xpath="@tool-id"/>
</xsd:keyref>
<xsd:keyref name="KR_phase_tool-id" refer="xlf:K_tool-id">
  <xsd:selector xpath="xlf:header/xlf:phase-group/xlf:phase"/>
  <xsd:field xpath="@tool-id"/>
</xsd:keyref>
<xsd:keyref name="KR_alt-trans_tool-id" refer="xlf:K_tool-id">
  <xsd:selector xpath="//xlf:trans-unit/xlf:alt-trans"/>
  <xsd:field xpath="@tool-id"/>
</xsd:keyref>
<xsd:key name="K_count-group_name">
  <xsd:selector xpath="//xlf:count-group"/>
  <xsd:field xpath="@name"/>
</xsd:key>
<xsd:unique name="U_context-group_name">
  <xsd:selector xpath="//xlf:context-group"/>
  <xsd:field xpath="@name"/>
</xsd:unique>
<xsd:key name="K_phase-name">
  <xsd:selector xpath="xlf:header/xlf:phase-group/xlf:phase"/>
  <xsd:field xpath="@phase-name"/>
</xsd:key>
<xsd:keyref name="KR_phase-name" refer="xlf:K_phase-name">
  <xsd:selector xpath="//xlf:count|//xlf:trans-unit|//xlf:target|//bin-
unit|//bin-target"/>
  <xsd:field xpath="@phase-name"/>
</xsd:keyref>
<xsd:unique name="U_uid">
  <xsd:selector xpath="//xlf:external-file"/>
  <xsd:field xpath="@uid"/>
</xsd:unique>
</xsd:element>
<xsd:element name="header-rev">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element minOccurs="0" ref="xlf:phase-group"/>
      <xsd:choice maxOccurs="unbounded" minOccurs="0">
        <xsd:element name="glossary" type="xlf:ElemType_ExternalReference"/>
        <xsd:element name="reference" type="xlf:ElemType_ExternalReference"/>
        <xsd:element ref="xlf:count-group"/>
        <xsd:element ref="xlf:note"/>
        <xsd:element ref="xlf:tool"/>
      </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

```

```

        </xsd:choice>
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="body-rev">
    <xsd:complexType>
        <xsd:choice maxOccurs="unbounded" minOccurs="0">
            <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:group-rev"/>
            <xsd:element maxOccurs="unbounded" minOccurs="0" ref="xlf:trans-unit"/>
        </xsd:choice>
    </xsd:complexType>
</xsd:element>
<xsd:element name="group-rev">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:choice maxOccurs="unbounded">
                <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:group-rev"/>
                <xsd:element maxOccurs="unbounded" minOccurs="0" ref="xlf:trans-unit"/>
            </xsd:choice>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:string" use="optional"/>
        <xsd:attribute name="datatype" type="xlf:AttrType_datatype" use="optional"/>
        <xsd:attribute default="default" ref="xml:space" use="optional"/>
        <xsd:attribute name="restype" type="xlf:AttrType_restype" use="optional"/>
        <xsd:attribute name="resname" type="xsd:string" use="optional"/>
        <xsd:attribute name="extradata" type="xsd:string" use="optional"/>
        <xsd:attribute name="extype" type="xsd:string" use="optional"/>
        <xsd:attribute name="help-id" type="xsd:NMTOKEN" use="optional"/>
        <xsd:attribute name="menu" type="xsd:string" use="optional"/>
        <xsd:attribute name="menu-option" type="xsd:string" use="optional"/>
        <xsd:attribute name="menu-name" type="xsd:string" use="optional"/>
        <xsd:attribute name="coord" type="xlf:AttrType_Coordinates" use="optional"/>
        <xsd:attribute name="font" type="xsd:string" use="optional"/>
        <xsd:attribute name="css-style" type="xsd:string" use="optional"/>
        <xsd:attribute name="style" type="xsd:NMTOKEN" use="optional"/>
        <xsd:attribute name="exstyle" type="xsd:NMTOKEN" use="optional"/>
        <xsd:attribute default="yes" name="translate" type="xlf:AttrType_YesNo"
use="optional"/>
        <xsd:attribute default="yes" name="reformat" type="xlf:AttrType_reformat"
use="optional"/>
        <xsd:attribute default="pixel" name="size-unit" type="xlf:AttrType_size-unit"
use="optional"/>
        <xsd:attribute name="maxwidth" type="xsd:NMTOKEN" use="optional"/>
        <xsd:attribute name="minwidth" type="xsd:NMTOKEN" use="optional"/>
        <xsd:attribute name="maxheight" type="xsd:NMTOKEN" use="optional"/>
        <xsd:attribute name="minheight" type="xsd:NMTOKEN" use="optional"/>
        <xsd:attribute name="maxbytes" type="xsd:NMTOKEN" use="optional"/>
        <xsd:attribute name="minbytes" type="xsd:NMTOKEN" use="optional"/>
        <xsd:attribute name="charclass" type="xsd:string" use="optional"/>
        <xsd:attribute default="no" name="merged-trans" type="xlf:AttrType_YesNo"
use="optional"/>
    </xsd:complexType>
</xsd:element>
</xsd:schema>

```

9. Target Generation Service

9.1 TargetGeneration.wsdl

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<definitions
    name="TragetGeneration"
    targetNamespace="http://tcd.ie/nds/swl10n/ws/contract/targetgeneration"
    xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/targetgeneration"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns="http://schemas.xmlsoap.org/wsdl/"

```

```

xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ns1="http://tcd.ie/nds/swl10n/ws/schema/targetgeneration">
<types>
  <xs:schema
    targetNamespace="http://tcd.ie/nds/swl10n/ws/schema/targetgeneration"
    xmlns:tns="http://tcd.ie/nds/swl10n/ws/schema/targetgeneration"
    xmlns:xf="urn:oasis:names:tc:xliff:document:1.2"
    xmlns:tgx="http://tcd.ie/nds/swl10n/ws/schema/targetgenerationXliff"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:import namespace="urn:oasis:names:tc:xliff:document:1.2"
schemaLocation="xliff-core-1.2-strict.xsd"/>
    <xs:import namespace="http://tcd.ie/nds/swl10n/ws/schema/targetgenerationXliff"
schemaLocation="TargetGenerationXliff.xsd"/>
    <xs:element name="generateTargetXlfRequest"
type="tns:generateTargetXlfRequestType"/>
    <xs:element name="generateTargetRequest" type="tns:generateTargetRequestType"/>
    <xs:element name="generateTargetResponse"
type="tns:generateTargetResponseType"/>
    <xs:element name="UnsupportedDataType" type="xs:string"/>
    <xs:element name="MissingSkeleton" type="xs:string"/>
    <xs:element name="OperationFailed" type="xs:string"/>
    <xs:complexType name="generateTargetXlfRequestType">
      <xs:sequence>
        <xs:element ref="xf:xliff" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
    <xs:complexType name="generateTargetRequestType">
      <xs:sequence>
        <xs:element ref="tgx:xliff-tg" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
    <xs:complexType name="generateTargetResponseType">
      <xs:sequence>
        <xs:element name="targets" type="tns:targetFile" nillable="true"
minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
    <xs:complexType name="targetFile">
      <xs:sequence>
        <xs:element name="fileName" type="xs:string" minOccurs="0"/>
        <xs:element name="fileContent" type="xs:base64Binary" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:schema>
</types>
<message name="msgGenerateTargetXlfRequest">
  <part name="Request" element="ns1:generateTargetXlfRequest"/>
</message>
<message name="msgGenerateTargetRequest">
  <part name="Request" element="ns1:generateTargetRequest"/>
</message>
<message name="msgGenerateTargetResponse">
  <part name="Response" element="ns1:generateTargetResponse"/>
</message>
<message name="UnsupportedDataType">
  <part name="fault" element="ns1:UnsupportedDataType"/>
</message>
<message name="MissingSkeleton">
  <part name="fault" element="ns1:MissingSkeleton"/>
</message>
<message name="OperationFailed">
  <part name="fault" element="ns1:OperationFailed"/>
</message>
<portType name="ptTargetGeneration">
  <operation name="opGenerateTargetXlf">
    <input message="tns:msgGenerateTargetXlfRequest"/>
    <output message="tns:msgGenerateTargetResponse"/>
    <fault message="tns:UnsupportedDataType" name="UnsupportedDataType"/>
  </operation>
</portType>

```

```

        <fault message="tns:MissingSkeleton" name="MissingSkeleton"/>
        <fault message="tns:OperationFailed" name="OperationFailed"/>
    </operation>
    <operation name="opGenerateTarget">
        <input message="tns:msgGenerateTargetRequest"/>
        <output message="tns:msgGenerateTargetResponse"/>
        <fault message="tns:UnsupportedDataType" name="UnsupportedDataType"/>
        <fault message="tns:OperationFailed" name="OperationFailed"/>
    </operation>
</portType>
<binding name="Binding1" type="tns:ptTargetGeneration">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <operation name="opGenerateTargetXliff">
        <soap:operation soapAction=""/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
        <fault name="UnsupportedDataType">
            <soap:fault name="UnsupportedDataType" use="literal"/>
        </fault>
        <fault name="MissingSkeleton">
            <soap:fault name="MissingSkeleton" use="literal"/>
        </fault>
        <fault name="OperationFailed">
            <soap:fault name="OperationFailed" use="literal"/>
        </fault>
    </operation>
    <operation name="opGenerateTarget">
        <soap:operation soapAction=""/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
        <fault name="UnsupportedDataType">
            <soap:fault name="UnsupportedDataType" use="literal"/>
        </fault>
        <fault name="OperationFailed">
            <soap:fault name="OperationFailed" use="literal"/>
        </fault>
    </operation>
</binding>
<service name="svTargetGeneration">
    <port name="targetgeneration" binding="tns:Binding1">
        <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
    </port>
</service>
</definitions>

```

9.2 TargetGenerationXliff.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://tcd.ie/nds/swl10n/ws/schema/targetgenerationXliff"
    xmlns:tns="http://tcd.ie/nds/swl10n/ws/schema/targetgenerationXliff"
    elementFormDefault="qualified"
    xmlns:xliff="urn:oasis:names:tc:xliff:document:1.2"
    xmlns:xml="http://www.w3.org/XML/1998/namespace">
    <xsd:import namespace="urn:oasis:names:tc:xliff:document:1.2" schemaLocation="xliff-
core-1.2-strict.xsd"/>
    <xsd:import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="http://www.w3.org/2001/xml.xsd"/>
    <xsd:element name="xliff-tg">
        <xsd:complexType>

```

```

        <xsd:sequence maxOccurs="unbounded">
            <xsd:element ref="tns:file-tg"/>
        </xsd:sequence>
        <xsd:attribute name="version" type="xlf:AttrType_Version" use="required"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="file-tg">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element minOccurs="1" ref="tns:header-tg"/>
            <xsd:element ref="tns:body-tg"/>
        </xsd:sequence>
        <xsd:attribute name="original" type="xsd:string" use="required"/>
        <xsd:attribute name="source-language" type="xsd:language" use="required"/>
        <xsd:attribute name="datatype" type="xlf:AttrType_datatype" use="required"/>
        <xsd:attribute ref="xml:space" use="optional"/>
        <xsd:attribute name="category" type="xsd:string" use="optional"/>
        <xsd:attribute name="target-language" type="xsd:language" use="optional"/>
        <xsd:attribute name="product-name" type="xsd:string" use="optional"/>
        <xsd:attribute name="product-version" type="xsd:string" use="optional"/>
        <xsd:attribute name="build-num" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:unique name="U_group_id">
        <xsd:selector xpath="//xlf:group"/>
        <xsd:field xpath="@id"/>
    </xsd:unique>
    <xsd:key name="K_unit_id">
        <xsd:selector xpath="//xlf:trans-unit|//xlf:bin-unit"/>
        <xsd:field xpath="@id"/>
    </xsd:key>
    <xsd:keyref name="KR_unit_id" refer="xlf:K_unit_id">
        <xsd:selector xpath="//bpt|//ept|//it|//ph|//g|//x|//bx|//ex|//sub"/>
        <xsd:field xpath="@xid"/>
    </xsd:keyref>
</xsd:element>
<xsd:element name="header-tg">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element minOccurs="1" name="sk1" type="xlf:ElemType_ExternalReference"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="body-tg">
    <xsd:complexType>
        <xsd:choice maxOccurs="unbounded" minOccurs="0">
            <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:group-tg"/>
            <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:trans-unit-tg"/>
            <xsd:element maxOccurs="unbounded" minOccurs="0" ref="xlf:bin-unit"/>
        </xsd:choice>
    </xsd:complexType>
</xsd:element>
<xsd:element name="group-tg">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:choice maxOccurs="unbounded">
                <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:group-tg"/>
                <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:trans-unit-tg"/>
                <xsd:element maxOccurs="unbounded" minOccurs="0" ref="xlf:bin-unit"/>
            </xsd:choice>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:string" use="optional"/>
        <xsd:attribute name="datatype" type="xlf:AttrType_datatype" use="optional"/>
        <xsd:attribute default="default" ref="xml:space" use="optional"/>
        <xsd:attribute name="restype" type="xlf:AttrType_restype" use="optional"/>
        <xsd:attribute name="resname" type="xsd:string" use="optional"/>
        <xsd:attribute name="extradata" type="xsd:string" use="optional"/>
        <xsd:attribute name="extype" type="xsd:string" use="optional"/>
        <xsd:attribute name="help-id" type="xsd:NMTOKEN" use="optional"/>
    </xsd:complexType>
</xsd:element>

```

```

<xsd:attribute name="menu" type="xsd:string" use="optional"/>
<xsd:attribute name="menu-option" type="xsd:string" use="optional"/>
<xsd:attribute name="menu-name" type="xsd:string" use="optional"/>
<xsd:attribute name="coord" type="xlf:AttrType_Coordinates" use="optional"/>
<xsd:attribute name="font" type="xsd:string" use="optional"/>
<xsd:attribute name="css-style" type="xsd:string" use="optional"/>
<xsd:attribute name="style" type="xsd:NMTOKEN" use="optional"/>
<xsd:attribute name="exstyle" type="xsd:NMTOKEN" use="optional"/>
<xsd:attribute default="yes" name="translate" type="xlf:AttrType_YesNo"
use="optional"/>
  <xsd:attribute default="yes" name="reformat" type="xlf:AttrType_reformat"
use="optional"/>
  <xsd:attribute default="pixel" name="size-unit" type="xlf:AttrType_size-unit"
use="optional"/>
  <xsd:attribute name="maxwidth" type="xsd:NMTOKEN" use="optional"/>
  <xsd:attribute name="minwidth" type="xsd:NMTOKEN" use="optional"/>
  <xsd:attribute name="maxheight" type="xsd:NMTOKEN" use="optional"/>
  <xsd:attribute name="minheight" type="xsd:NMTOKEN" use="optional"/>
  <xsd:attribute name="maxbytes" type="xsd:NMTOKEN" use="optional"/>
  <xsd:attribute name="minbytes" type="xsd:NMTOKEN" use="optional"/>
  <xsd:attribute name="charclass" type="xsd:string" use="optional"/>
  <xsd:attribute default="no" name="merged-trans" type="xlf:AttrType_YesNo"
use="optional"/>
</xsd:complexType>
</xsd:element>
<xsd:element name="trans-unit-tg">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xlf:source"/>
      <xsd:element minOccurs="0" ref="xlf:target"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:string" use="required"/>
    <xsd:attribute name="approved" type="xlf:AttrType_YesNo" use="optional"/>
    <xsd:attribute default="yes" name="translate" type="xlf:AttrType_YesNo"
use="optional"/>
    <xsd:attribute default="yes" name="reformat" type="xlf:AttrType_reformat"
use="optional"/>
    <xsd:attribute default="default" ref="xml:space" use="optional"/>
    <xsd:attribute name="datatype" type="xlf:AttrType_datatype" use="optional"/>
    <xsd:attribute name="phase-name" type="xsd:string" use="optional"/>
    <xsd:attribute name="restype" type="xlf:AttrType_restype" use="optional"/>
    <xsd:attribute name="resname" type="xsd:string" use="optional"/>
    <xsd:attribute name="extradata" type="xsd:string" use="optional"/>
    <xsd:attribute name="extype" type="xsd:string" use="optional"/>
    <xsd:attribute name="help-id" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="menu" type="xsd:string" use="optional"/>
    <xsd:attribute name="menu-option" type="xsd:string" use="optional"/>
    <xsd:attribute name="menu-name" type="xsd:string" use="optional"/>
    <xsd:attribute name="coord" type="xlf:AttrType_Coordinates" use="optional"/>
    <xsd:attribute name="font" type="xsd:string" use="optional"/>
    <xsd:attribute name="css-style" type="xsd:string" use="optional"/>
    <xsd:attribute name="style" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="exstyle" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute default="pixel" name="size-unit" type="xlf:AttrType_size-unit"
use="optional"/>
    <xsd:attribute name="maxwidth" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="minwidth" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="maxheight" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="minheight" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="maxbytes" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="minbytes" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="charclass" type="xsd:string" use="optional"/>
    <xsd:attribute default="yes" name="merged-trans" type="xlf:AttrType_YesNo"
use="optional"/>
  </xsd:complexType>
</xsd:element>
</xsd:schema>

```

10. Testing Service

10.1 Testing.wsdl

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<definitions
  name="Testing"
  targetNamespace="http://tcd.ie/nds/swl10n/ws/contract/testing"
  xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/testing"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ns1="http://tcd.ie/nds/swl10n/ws/schema/testing">
  <types>
    <xs:schema
      targetNamespace="http://tcd.ie/nds/swl10n/ws/schema/testing"
      xmlns:tns="http://tcd.ie/nds/swl10n/ws/schema/testing"
      xmlns:xf="urn:oasis:names:tc:xliff:document:1.2"
      xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:import namespace="urn:oasis:names:tc:xliff:document:1.2"
        schemaLocation="xliff-core-1.2-strict.xsd"/>
      <xs:element name="testRequest" type="tns:testRequestType"/>
      <xs:element name="acknowledgement" type="xs:string"/>
      <xs:element name="OperationFailed" type="xs:string"/>
      <xs:complexType name="testRequestType">
        <xs:sequence>
          <xs:element ref="xf:xliff" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
  </types>
  <message name="msgTestRequest">
    <part name="Request" element="ns1:testRequest"/>
  </message>
  <message name="msgTestResponse">
    <part name="Response" element="ns1:acknowledgement"/>
  </message>
  <message name="OperationFailed">
    <part name="fault" element="ns1:OperationFailed"/>
  </message>
  <portType name="ptTesting">
    <operation name="opTest">
      <input message="tns:msgTestRequest" />
      <output message="tns:msgTestResponse"/>
      <fault message="tns:OperationFailed" name="OperationFailed"/>
    </operation>
  </portType>
  <binding name="TestingPortBinding" type="tns:ptTesting">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <operation name="opTest">
      <soap:operation soapAction="http://tcd.ie/nds/swl10n/ws/test/request"/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
      <fault name="OperationFailed">
        <soap:fault name="OperationFailed" use="literal"/>
      </fault>
    </operation>
  </binding>
  <service name="svTesting">
    <port name="testing" binding="tns:TestingPortBinding">
      <soap:address location="http://localhost:8080/Prueba4"/>
    </port>
  </service>
</definitions>
```

```
</definitions>
```

11. Translation Service

11.1 Translation.wSDL

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<definitions
  name="Translation"
  targetNamespace="http://tcd.ie/nds/swl10n/ws/contract/translation"
  xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/translation"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ns1="http://tcd.ie/nds/swl10n/ws/schema/translation">
  <types>
    <xs:schema
      targetNamespace="http://tcd.ie/nds/swl10n/ws/schema/translation"
      xmlns:tns="http://tcd.ie/nds/swl10n/ws/schema/translation"
      xmlns:xf="urn:oasis:names:tc:xliff:document:1.2"
      xmlns:trx="http://tcd.ie/nds/swl10n/ws/schema/translationXliff"
      xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:import namespace="urn:oasis:names:tc:xliff:document:1.2"
        schemaLocation="xliff-core-1.2-strict.xsd"/>
      <xs:import namespace="http://tcd.ie/nds/swl10n/ws/schema/translationXliff"
        schemaLocation="TranslationXliff.xsd"/>
      <xs:element name="submitForTranslationXlfRequest"
        type="tns:submitForTranslationXlfRequestType"/>
      <xs:element name="submitForTranslationRequest"
        type="tns:submitForTranslationRequestType"/>
      <xs:element name="submitForTranslationResponse"
        type="tns:submitForTranslationResponseType"/>
      <xs:element name="getTranslationRequest" type="tns:getTranslationRequestType"/>
      <xs:element name="getTranslationResponse"
        type="tns:getTranslationResponseType"/>
      <xs:element name="OperationFailed" type="xs:string"/>
      <xs:complexType name="submitForTranslationXlfRequestType">
        <xs:sequence>
          <xs:element name="targetLang" type="xs:string" minOccurs="0"/>
          <xs:element ref="xf:xliff" minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="submitForTranslationRequestType">
        <xs:sequence>
          <xs:element ref="trx:xliff-tr" minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="submitForTranslationResponseType">
        <xs:sequence>
          <xs:element name="id" type="xs:string" minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="getTranslationRequestType">
        <xs:sequence>
          <xs:element name="id" type="xs:string" minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="getTranslationResponseType">
        <xs:sequence>
          <xs:element ref="xf:xliff" minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
  </types>
  <message name="msgSubmitForTranslationXlfRequest">
    <part name="Request" element="ns1:submitForTranslationXlfRequest"/>
  </message>
</definitions>
```



```

</message>
<message name="msgSubmitForTranslationRequest">
  <part name="Request" element="ns1:submitForTranslationRequest"/>
</message>
<message name="msgSubmitForTranslationResponse">
  <part name="Response" element="ns1:submitForTranslationResponse"/>
</message>
<message name="msgGetTranslationRequest">
  <part name="Request" element="ns1:getTranslationRequest"/>
</message>
<message name="msgGetTranslationResponse">
  <part name="Response" element="ns1:getTranslationResponse"/>
</message>
<message name="OperationFailed">
  <part name="fault" element="ns1:OperationFailed"/>
</message>
<portType name="ptTranslation">
  <operation name="opSubmitForTranslationXlf">
    <input message="tns:msgSubmitForTranslationXlfRequest"/>
    <output message="tns:msgSubmitForTranslationResponse"/>
    <fault message="tns:OperationFailed" name="OperationFailed"/>
  </operation>
  <operation name="opSubmitForTranslation">
    <input message="tns:msgSubmitForTranslationRequest"/>
    <output message="tns:msgSubmitForTranslationResponse"/>
    <fault message="tns:OperationFailed" name="OperationFailed"/>
  </operation>
  <operation name="opGetTranslation">
    <input message="tns:msgGetTranslationRequest"/>
    <output message="tns:msgGetTranslationResponse"/>
    <fault message="tns:OperationFailed" name="OperationFailed"/>
  </operation>
</portType>
<binding name="Binding1" type="tns:ptTranslation">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
  <operation name="opSubmitForTranslationXlf">
    <soap:operation soapAction=""/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
    <fault name="OperationFailed">
      <soap:fault name="OperationFailed" use="literal"/>
    </fault>
  </operation>
  <operation name="opSubmitForTranslation">
    <soap:operation soapAction=""/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
    <fault name="OperationFailed">
      <soap:fault name="OperationFailed" use="literal"/>
    </fault>
  </operation>
  <operation name="opGetTranslation">
    <soap:operation soapAction=""/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
    <fault name="OperationFailed">

```

```

        <soap:fault name="OperationFailed" use="literal"/>
    </fault>
</operation>
</binding>
<service name="svTranslation">
    <port name="translation" binding="tns:Binding1">
        <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
    </port>
</service>
</definitions>

```

11.2 TranslationXliff.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://tcd.ie/nds/swl10n/ws/schema/translationXliff"
    xmlns:tns="http://tcd.ie/nds/swl10n/ws/schema/translationXliff"
    elementFormDefault="qualified"
    xmlns:xlif="urn:oasis:names:tc:xliff:document:1.2"
    xmlns:xml="http://www.w3.org/XML/1998/namespace"
>
    <xsd:import namespace="urn:oasis:names:tc:xliff:document:1.2" schemaLocation="xliff-
core-1.2-strict.xsd"/>
    <xsd:import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="http://www.w3.org/2001/xml.xsd"/>
    <xsd:element name="xliff-tr">
        <xsd:complexType>
            <xsd:sequence maxOccurs="unbounded">
                <xsd:element ref="tns:file-tr"/>
            </xsd:sequence>
            <xsd:attribute name="version" type="xlif:AttrType_Version" use="required"/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="file-tr">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element minOccurs="0" ref="tns:header-tr"/>
                <xsd:element ref="tns:body-tr"/>
            </xsd:sequence>
            <xsd:attribute name="original" type="xsd:string" use="required"/>
            <xsd:attribute name="source-language" type="xsd:language" use="required"/>
            <xsd:attribute name="datatype" type="xlif:AttrType_datatype" use="required"/>
            <xsd:attribute name="tool-id" type="xsd:string" use="optional"/>
            <xsd:attribute name="date" type="xsd:dateTime" use="optional"/>
            <xsd:attribute ref="xml:space" use="optional"/>
            <xsd:attribute name="category" type="xsd:string" use="optional"/>
            <xsd:attribute name="target-language" type="xsd:language" use="required"/>
            <xsd:attribute name="product-name" type="xsd:string" use="optional"/>
            <xsd:attribute name="product-version" type="xsd:string" use="optional"/>
            <xsd:attribute name="build-num" type="xsd:string" use="optional"/>
        </xsd:complexType>
        <xsd:unique name="U_group_id">
            <xsd:selector xpath="//xlif:group"/>
            <xsd:field xpath="@id"/>
        </xsd:unique>
        <xsd:key name="K_unit_id">
            <xsd:selector xpath="//xlif:trans-unit|//xlif:bin-unit"/>
            <xsd:field xpath="@id"/>
        </xsd:key>
        <xsd:keyref name="KR_unit_id" refer="xlif:K_unit_id">
            <xsd:selector xpath="//bpt|//ept|//it|//ph|//g|//x|//bx|//ex|//sub"/>
            <xsd:field xpath="@xid"/>
        </xsd:keyref>
        <xsd:key name="K_tool-id">
            <xsd:selector xpath="xlif:header/xlif:tool"/>
            <xsd:field xpath="@tool-id"/>
        </xsd:key>
        <xsd:keyref name="KR_file_tool-id" refer="xlif:K_tool-id">

```

```

    <xsd:selector xpath="."/>
    <xsd:field xpath="@tool-id"/>
  </xsd:keyref>
  <xsd:keyref name="KR_phase_tool-id" refer="xlf:K_tool-id">
    <xsd:selector xpath="xlf:header/xlf:phase-group/xlf:phase"/>
    <xsd:field xpath="@tool-id"/>
  </xsd:keyref>
  <xsd:keyref name="KR_alt-trans_tool-id" refer="xlf:K_tool-id">
    <xsd:selector xpath="//xlf:trans-unit/xlf:alt-trans"/>
    <xsd:field xpath="@tool-id"/>
  </xsd:keyref>
  <xsd:key name="K_count-group_name">
    <xsd:selector xpath="//xlf:count-group"/>
    <xsd:field xpath="@name"/>
  </xsd:key>
  <xsd:unique name="U_context-group_name">
    <xsd:selector xpath="//xlf:context-group"/>
    <xsd:field xpath="@name"/>
  </xsd:unique>
  <xsd:key name="K_phase-name">
    <xsd:selector xpath="xlf:header/xlf:phase-group/xlf:phase"/>
    <xsd:field xpath="@phase-name"/>
  </xsd:key>
  <xsd:keyref name="KR_phase-name" refer="xlf:K_phase-name">
    <xsd:selector xpath="//xlf:count|//xlf:trans-unit|//xlf:target|//bin-
unit|//bin-target"/>
    <xsd:field xpath="@phase-name"/>
  </xsd:keyref>
  <xsd:unique name="U_uid">
    <xsd:selector xpath="//xlf:external-file"/>
    <xsd:field xpath="@uid"/>
  </xsd:unique>
</xsd:element>
<xsd:element name="header-tr">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element minOccurs="0" ref="xlf:phase-group"/>
      <xsd:choice maxOccurs="unbounded" minOccurs="0">
        <xsd:element name="glossary" type="xlf:ElemType_ExternalReference"/>
        <xsd:element name="reference" type="xlf:ElemType_ExternalReference"/>
        <xsd:element ref="xlf:count-group"/>
        <xsd:element ref="xlf:note"/>
        <xsd:element ref="xlf:tool"/>
      </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="body-tr">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:group-tr"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" ref="xlf:trans-unit"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
<xsd:element name="group-tr">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:choice maxOccurs="unbounded">
        <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:group-tr"/>
        <xsd:element maxOccurs="unbounded" minOccurs="0" ref="xlf:trans-unit"/>
      </xsd:choice>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:string" use="optional"/>
    <xsd:attribute name="datatype" type="xlf:AttrType_datatype" use="optional"/>
    <xsd:attribute default="default" ref="xml:space" use="optional"/>
    <xsd:attribute name="restype" type="xlf:AttrType_restype" use="optional"/>
    <xsd:attribute name="resname" type="xsd:string" use="optional"/>
  </xsd:complexType>

```

```

<xsd:attribute name="extradata" type="xsd:string" use="optional"/>
<xsd:attribute name="extype" type="xsd:string" use="optional"/>
<xsd:attribute name="help-id" type="xsd:NMTOKEN" use="optional"/>
<xsd:attribute name="menu" type="xsd:string" use="optional"/>
<xsd:attribute name="menu-option" type="xsd:string" use="optional"/>
<xsd:attribute name="menu-name" type="xsd:string" use="optional"/>
<xsd:attribute name="coord" type="xlf:AttrType_Coordinates" use="optional"/>
<xsd:attribute name="font" type="xsd:string" use="optional"/>
<xsd:attribute name="css-style" type="xsd:string" use="optional"/>
<xsd:attribute name="style" type="xsd:NMTOKEN" use="optional"/>
<xsd:attribute name="exstyle" type="xsd:NMTOKEN" use="optional"/>
<xsd:attribute default="yes" name="translate" type="xlf:AttrType_YesNo"
use="optional"/>
<xsd:attribute default="yes" name="reformat" type="xlf:AttrType_reformat"
use="optional"/>
<xsd:attribute default="pixel" name="size-unit" type="xlf:AttrType_size-unit"
use="optional"/>
<xsd:attribute name="maxwidth" type="xsd:NMTOKEN" use="optional"/>
<xsd:attribute name="minwidth" type="xsd:NMTOKEN" use="optional"/>
<xsd:attribute name="maxheight" type="xsd:NMTOKEN" use="optional"/>
<xsd:attribute name="minheight" type="xsd:NMTOKEN" use="optional"/>
<xsd:attribute name="maxbytes" type="xsd:NMTOKEN" use="optional"/>
<xsd:attribute name="minbytes" type="xsd:NMTOKEN" use="optional"/>
<xsd:attribute name="charclass" type="xsd:string" use="optional"/>
<xsd:attribute default="no" name="merged-trans" type="xlf:AttrType_YesNo"
use="optional"/>
</xsd:complexType>
</xsd:element>
</xsd:schema>

```

12. Update Service

12.1 Update.wsdl

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<definitions
  name="Update"
  targetNamespace="http://tcd.ie/nds/swl10n/ws/contract/update"
  xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/update"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ns1="http://tcd.ie/nds/swl10n/ws/schema/update">
  <types>
    <xs:schema
      targetNamespace="http://tcd.ie/nds/swl10n/ws/schema/update"
      xmlns:tns="http://tcd.ie/nds/swl10n/ws/schema/update"
      xmlns:xlf="urn:oasis:names:tc:xliff:document:1.2"
      xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:import namespace="urn:oasis:names:tc:xliff:document:1.2"
        schemaLocation="xliff-core-1.2-strict.xsd"/>
      <xs:element name="updateRequest" type="tns:updateRequestType"/>
      <xs:element name="updateResponse" type="tns:updateResponseType"/>
      <xs:element name="OperationFailed" type="xs:string"/>
      <xs:complexType name="updateRequestType">
        <xs:sequence>
          <xs:element name="targetLang" type="xs:string" minOccurs="0"/>
          <xs:element ref="xlf:xliff" minOccurs="0"/>
          <xs:element name="previousVersion" type="tns:previousVersionType"
            minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="previousVersionType">
        <xs:sequence>
          <xs:element ref="xlf:xliff" minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
  </types>

```

```

    </xs:complexType>
    <xs:complexType name="updateResponseType">
      <xs:sequence>
        <xs:element ref="xlf:xliff" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:schema>
</types>
<message name="msgUpdateRequest">
  <part name="Request" element="ns1:updateRequest"/>
</message>
<message name="msgUpdateResponse">
  <part name="Response" element="ns1:updateResponse"/>
</message>
<message name="OperationFailed">
  <part name="fault" element="ns1:OperationFailed"/>
</message>
<portType name="ptUpdate">
  <operation name="opUpdate">
    <input message="tns:msgUpdateRequest"/>
    <output message="tns:msgUpdateResponse"/>
    <fault message="tns:OperationFailed" name="OperationFailed"/>
  </operation>
</portType>
<binding name="Binding1" type="tns:ptUpdate">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
  <operation name="opUpdate">
    <soap:operation soapAction=""/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
    <fault name="OperationFailed">
      <soap:fault name="OperationFailed" use="literal"/>
    </fault>
  </operation>
</binding>
<service name="svUpdate">
  <port name="update" binding="tns:Binding1">
    <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
  </port>
</service>
</definitions>

```

Appendix 3: BPEL scripts for Software Localisation Workflows

1 MtWorkflow.bpel

```
<?xml version="1.0" encoding="UTF-8"?>
<process
  name="MTWorkflow"
  targetNamespace="http://enterprise.netbeans.org/bpel/Localisation/MTWorkflow"
  xmlns="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:bpws="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
  xmlns:sxt="http://www.sun.com/wsbpel/2.0/process/executable/SUNExtension/Trace"
  xmlns:sxed="http://www.sun.com/wsbpel/2.0/process/executable/SUNExtension/Editor"
  xmlns:sxat="http://www.sun.com/wsbpel/2.0/process/executable/SUNExtension/Attachment"
  xmlns:sxeh="http://www.sun.com/wsbpel/2.0/process/executable/SUNExtension/ErrorHandling"
  xmlns:wSDLNS="http://enterprise.netbeans.org/bpel/Localisation/MTWorkflow"
  xmlns:ns1="http://tcd.ie/nds/swl10n/bpel/localisation"
  xmlns:ns2="http://tcd.ie/nds/swl10n/bpel/schema/localisation"
  xmlns:sxed2="http://www.sun.com/wsbpel/2.0/process/executable/SUNExtension/Editor2"
  xmlns:xlif="urn:oasis:names:tc:xliff:document:1.2"
  xmlns:ns4="http://tcd.ie/nds/swl10n/ws/schema/parsing">
  <documentation>
    The MTWorkflow BPEL process illustrates a localisation flow that uses a Machine Translation service. The process receives an input message and sends it back synchronously. A client starts the synchronous process by invoking a request-response operation. This is a synchronous process; the client is blocked until the process finishes and returns the result.
  </documentation>
  <import namespace="http://tcd.ie/nds/swl10n/bpel/localisation"
    location="./Localisation.wsdl"
    importType="http://schemas.xmlsoap.org/wsdl/">
  <import namespace="http://enterprise.netbeans.org/bpel/svParsingWrapper"
    location="svParsingWrapper.wsdl" importType="http://schemas.xmlsoap.org/wsdl/">
  <import namespace="http://tcd.ie/nds/swl10n/ws/contract/parsing" location="localhost_8080/Sw_Localisation_Web_Services/svParsing.wsdl" importType="http://schemas.xmlsoap.org/wsdl/">
  <import namespace="http://enterprise.netbeans.org/bpel/svMachineTranslationWrapper"
    location="svMachineTranslationWrapper.wsdl"
    importType="http://schemas.xmlsoap.org/wsdl/">
  <import namespace="http://tcd.ie/nds/swl10n/ws/contract/machinetranslation" location="localhost_8080/Sw_Localisation_Web_Services/svMachineTranslation.wsdl" importType="http://schemas.xmlsoap.org/wsdl/">
  <import namespace="http://enterprise.netbeans.org/bpel/svTargetGenerationWrapper"
    location="svTargetGenerationWrapper.wsdl"
    importType="http://schemas.xmlsoap.org/wsdl/">
  <import namespace="http://tcd.ie/nds/swl10n/ws/contract/targetgeneration"
    location="localhost_8080/Sw_Localisation_Web_Services/svTargetGeneration.wsdl" importType="http://schemas.xmlsoap.org/wsdl/">
  <partnerLinks>
    <partnerLink name="Parsing" xmlns:tns="http://enterprise.netbeans.org/bpel/svParsingWrapper" partnerLinkType="tns:ptParsingLinkType" partnerRole="ptParsingRole"/>
    <partnerLink name="MachineTranslation"
      xmlns:tns="http://enterprise.netbeans.org/bpel/svMachineTranslationWrapper" partnerLinkType="tns:ptMachineTranslationLinkType" partnerRole="ptMachineTranslationRole"/>
    <partnerLink name="TargetGeneration"
      xmlns:tns="http://enterprise.netbeans.org/bpel/svTargetGenerationWrapper" partnerLinkType="tns:ptTargetGenerationLinkType" partnerRole="ptTargetGenerationRole"/>
    <partnerLink
      name="Localisation"
      partnerLinkType="ns1:partnerlinktype1"
      myRole="partnerlinktyperole1"/>
  </partnerLinks>
</process>
```

```

</partnerLinks>
<variables>
  <variable name="OpGenerateTargetOut" xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/targetgeneration" messageType="tns:msgGenerateTargetResponse"/>
  <variable name="OpGenerateTargetIn" xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/targetgeneration" messageType="tns:msgGenerateTargetXlfRequest"/>
  <variable name="OpMachineTranslateOut" xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/machinetranslation" messageType="tns:msgMachineTranslateResponse"/>
  <variable name="OpMachineTranslateIn" xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/machinetranslation" messageType="tns:msgMachineTranslateXlfRequest"/>
  <variable name="OpParseFilesOut"
xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/parsing" messageType="tns:msgParseFilesResponse"/>
  <variable name="OpParseFilesIn"
xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/parsing" messageType="tns:msgParseFilesRequest"/>
  <variable name="outputVar" messageType="ns1:msgWfMachineTranslateResponse">
    <documentation>Output variable.</documentation>
  </variable>
  <variable name="inputVar" messageType="ns1:msgWfMachineTranslateRequest">
    <documentation>Input variable.</documentation>
  </variable>
</variables>
<sequence>
  <receive
    name="start"
    partnerLink="Localisation"
    operation="opWfMachineTranslate"
    portType="ns1:ptMachineTranslation"
    variable="inputVar"
    createInstance="yes">
  </receive>
  <assign name="Assign1">
    <copy>
      <from>$inputVar.Request/ns2:fileInfo/ns2:dataType</from>
      <to>$OpParseFilesIn.Request/fileInfo/dataType</to>
    </copy>
    <copy>
      <from>$inputVar.Request/ns2:fileInfo/ns2:fileLanguage</from>
      <to>$OpParseFilesIn.Request/fileInfo/fileLanguage</to>
    </copy>
    <copy>
      <from>$inputVar.Request/ns2:fileInfo/ns2:fileName</from>
      <to>$OpParseFilesIn.Request/fileInfo/fileName</to>
    </copy>
    <copy>
      <from>$inputVar.Request/ns2:fileInfo/ns2:fileContent</from>
      <to>$OpParseFilesIn.Request/fileInfo/fileContent</to>
    </copy>
  </assign>
  <invoke name="InvokeParsing" partnerLink="Parsing" operation="opParseFiles"
xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/parsing" portType="tns:ptParsing" inputVariable="OpParseFilesIn" outputVariable="OpParseFilesOut"/>
  <assign name="Assign2">
    <copy>
      <from>$OpParseFilesOut.Response/xlf:xliff</from>
      <to>$OpMachineTranslateIn.Request/xlf:xliff</to>
    </copy>
    <copy>
      <from>$inputVar.Request/ns2:targetLanguage</from>
      <to>$OpMachineTranslateIn.Request/targetLang</to>
    </copy>
  </assign>
  <invoke name="InvokeMachineTranslation" partnerLink="MachineTranslation" operation="opMachineTranslateXlf" xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/machinetranslation" portType="tns:ptMachineTranslation" inputVariable="OpMachineTranslateIn" outputVariable="OpMachineTranslateOut"/>
  <assign name="Assign3">

```

```

    <copy>
      <from>$OpMachineTranslateOut.Response/xf:xliff</from>
      <to>$OpGenerateTargetIn.Request/xf:xliff</to>
    </copy>
  </assign>
  <invoke name="InvokeTargetGeneration" partnerLink="TargetGeneration"
operation="opGenerateTargetXlf" xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/tar-
getgeneration" portType="tns:ptTargetGeneration" inputVariable="OpGenerateTargetIn"
outputVariable="OpGenerateTargetOut"/>
  <assign name="Assign4">
    <copy>
      <from>$OpGenerateTargetOut.Response/targets/fileName</from>
      <to>$outputVar.Response/ns2:targets/ns2:fileName</to>
    </copy>
    <copy>
      <from>$OpGenerateTargetOut.Response/targets/fileContent</from>
      <to>$outputVar.Response/ns2:targets/ns2:fileContent</to>
    </copy>
  </assign>
</reply
name="end"
partnerLink="Localisation"
operation="opWfMachineTranslate"
portType="ns1:ptMachineTranslation"
variable="outputVar">
</reply>
</sequence>
</process>

```

2 TranslationWorkflow.bpel

```

<?xml version="1.0" encoding="UTF-8"?>
<process
name="TranslationWorkflow"
targetNamespace="http://enterprise.netbeans.org/bpel/Localisation/TranslationWork-
flow"
xmlns="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:bpws="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
xmlns:sxt="http://www.sun.com/wsbpel/2.0/process/executable/SUNExtension/Trace"
xmlns:sxed="http://www.sun.com/wsbpel/2.0/process/executable/SUNExtension/Editor"
xmlns:sxat="http://www.sun.com/wsbpel/2.0/process/executable/SUNExtension/Attach-
ment"
xmlns:sxeh="http://www.sun.com/wsbpel/2.0/process/executable/SUNExtension/ErrorHand-
ling"
xmlns:tns="http://enterprise.netbeans.org/bpel/Localisation/TranslationWorkflow"
xmlns:ns1="http://tcd.ie/nds/swl10n/bpel/localisation"
xmlns:ns2="http://tcd.ie/nds/swl10n/bpel/schema/localisation"
xmlns:xf="urn:oasis:names:tc:xliff:document:1.2"
xmlns:ns4="http://tcd.ie/nds/swl10n/ws/schema/parsing" xmlns:sxed2="http://www.sun.-
com/wsbpel/2.0/process/executable/SUNExtension/Editor2">
  <documentation>
    The TranslationWorkflow BPEL process illustrates a simple localisation flow.
  </documentation>
  <import namespace="http://tcd.ie/nds/swl10n/bpel/localisation"
location="./Localisation.wsdl"
importType="http://schemas.xmlsoap.org/wsdl/">
  <import namespace="http://enterprise.netbeans.org/bpel/svParsingWrapper"
location="svParsingWrapper.wsdl" importType="http://schemas.xmlsoap.org/wsdl/">
  <import namespace="http://tcd.ie/nds/swl10n/ws/contract/parsing" location="local-
host_8080/Sw_Localisation_Web_Services/svParsing.wsdl" importType="http://schem-
as.xmlsoap.org/wsdl/">
  <import namespace="http://enterprise.netbeans.org/bpel/svMachineTranslationWrapper"
location="svMachineTranslationWrapper.wsdl"
importType="http://schemas.xmlsoap.org/wsdl/">
  <import namespace="http://tcd.ie/nds/swl10n/ws/contract/machinetranslation" loca-

```



```

tion="localhost_8080/Sw_Localisation_Web_Services/svMachineTranslation.wsdl" import-
Type="http://schemas.xmlsoap.org/wsdl/">
  <import namespace="http://enterprise.netbeans.org/bpel/svTargetGenerationWrapper"
location="svTargetGenerationWrapper.wsdl"
importType="http://schemas.xmlsoap.org/wsdl/">
  <import namespace="http://tcd.ie/nds/swl10n/ws/contract/targetgeneration"
location="localhost_8080/Sw_Localisation_Web_Services/svTargetGeneration.wsdl" import-
Type="http://schemas.xmlsoap.org/wsdl/">
  <import namespace="http://enterprise.netbeans.org/bpel/svLeveragingWrapper" loca-
tion="svLeveragingWrapper.wsdl" importType="http://schemas.xmlsoap.org/wsdl/">
  <import namespace="http://tcd.ie/nds/swl10n/ws/contract/leveraging" location="local-
host_8080/Sw_Localisation_Web_Services/svLeveraging.wsdl" importType="http://schem-
as.xmlsoap.org/wsdl/">
  <import namespace="http://enterprise.netbeans.org/bpel/svTranslationWrapper" loca-
tion="svTranslationWrapper.wsdl" importType="http://schemas.xmlsoap.org/wsdl/">
  <import namespace="http://tcd.ie/nds/swl10n/ws/contract/translation" location="loc-
alhost_8080/Sw_Localisation_Web_Services/svTranslation.wsdl" importType="http://schem-
as.xmlsoap.org/wsdl/">
  <import namespace="http://enterprise.netbeans.org/bpel/svReviewWrapper"
location="svReviewWrapper.wsdl" importType="http://schemas.xmlsoap.org/wsdl/">
  <import namespace="http://tcd.ie/nds/swl10n/ws/contract/review"
location="localhost_8080/Sw_Localisation_Web_Services/svReview.wsdl"
importType="http://schemas.xmlsoap.org/wsdl/">
  <partnerLinks>
    <partnerLink name="Parsing" xmlns:tns="http://enterprise.netbeans.org/bpel/svPars-
ingWrapper" partnerLinkType="tns:ptParsingLinkType" partnerRole="ptParsingRole"/>
    <partnerLink name="Leveraging" xmlns:tns="http://enterprise.netbeans.org/bpel/sv-
LeveragingWrapper" partnerLinkType="tns:ptLeveragingLinkType" partnerRole="ptLever-
agingRole"/>
    <partnerLink name="Translation"
xmlns:tns="http://enterprise.netbeans.org/bpel/svTranslationWrapper"
partnerLinkType="tns:ptTranslationLinkType" partnerRole="ptTranslationRole"/>
    <partnerLink name="Review" xmlns:tns="http://enterprise.netbeans.org/bpel/svRe-
viewWrapper" partnerLinkType="tns:ptReviewLinkType" partnerRole="ptReviewRole"/>
    <partnerLink name="TargetGeneration"
xmlns:tns="http://enterprise.netbeans.org/bpel/svTargetGenerationWrapper" partnerLink-
Type="tns:ptTargetGenerationLinkType" partnerRole="ptTargetGenerationRole"/>
    <partnerLink
      name="Localisation"
      partnerLinkType="ns1:partnerlinktype1"
      myRole="serviceRequestor"/>
  </partnerLinks>
  <variables>
    <variable name="OpGetReviewOut"
xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/review" messageType="tns:msgGetRe-
viewResponse"/>
    <variable name="OpGetReviewIn" xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/re-
view" messageType="tns:msgGetReviewRequest"/>
    <variable name="OpSubmitForReviewOut" xmlns:tns="http://tcd.ie/nds/swl10n/ws/con-
tract/review" messageType="tns:msgSubmitForReviewResponse"/>
    <variable name="OpSubmitForReviewIn" xmlns:tns="http://tcd.ie/nds/swl10n/ws/con-
tract/review" messageType="tns:msgSubmitForReviewXlfRequest"/>
    <variable name="OpGetTranslationOut" xmlns:tns="http://tcd.ie/nds/swl10n/ws/con-
tract/translation" messageType="tns:msgGetTranslationResponse"/>
    <variable name="OpGetTranslationIn" xmlns:tns="http://tcd.ie/nds/swl10n/ws/con-
tract/translation" messageType="tns:msgGetTranslationRequest"/>
    <variable name="OpSubmitForTranslationOut"
xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/translation" messageType="tns:msgSub-
mitForTranslationResponse"/>
    <variable name="OpSubmitForTranslationIn"
xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/translation" messageType="tns:msgSub-
mitForTranslationXlfRequest"/>
    <variable name="OpLeverageOut"
xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/leveraging" messageType="tns:msgLever-
ageResponse"/>
    <variable name="OpLeverageIn"
xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/leveraging" messageType="tns:msgLever-
ageXlfRequest"/>

```

```

    <variable name="OpGenerateTargetOut" xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/targetgeneration" messageType="tns:msgGenerateTargetResponse"/>
    <variable name="OpGenerateTargetIn" xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/targetgeneration" messageType="tns:msgGenerateTargetXlfRequest"/>
    <variable name="OpParseFilesOut"
xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/parsing" messageType="tns:msgParseFilesResponse"/>
    <variable name="OpParseFilesIn"
xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/parsing" messageType="tns:msgParseFilesRequest"/>
    <variable name="outputVar" messageType="ns1:msgWfTranslateResponse"/>
    <variable name="inputVar" messageType="ns1:msgWfTranslateRequest"/>
</variables>
<sequence>
  <receive
    name="start"
    partnerLink="Localisation"
    operation="opWfTranslate"
    portType="ns1:ptTranslation"
    variable="inputVar"
    createInstance="yes"/>
  <assign name="Assign1">
    <copy>
      <from>$inputVar.Request/ns2:fileInfo/ns2:dataType</from>
      <to>$OpParseFilesIn.Request/fileInfo/dataType</to>
    </copy>
    <copy>
      <from>$inputVar.Request/ns2:fileInfo/ns2:fileLanguage</from>
      <to>$OpParseFilesIn.Request/fileInfo/fileLanguage</to>
    </copy>
    <copy>
      <from>$inputVar.Request/ns2:fileInfo/ns2:fileName</from>
      <to>$OpParseFilesIn.Request/fileInfo/fileName</to>
    </copy>
    <copy>
      <from>$inputVar.Request/ns2:fileInfo/ns2:fileContent</from>
      <to>$OpParseFilesIn.Request/fileInfo/fileContent</to>
    </copy>
  </assign>
  <invoke name="InvokeParsing" partnerLink="Parsing" operation="opParseFiles"
xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/parsing" portType="tns:ptParsing" inputVariable="OpParseFilesIn" outputVariable="OpParseFilesOut"/>
  <assign name="Assign2">
    <copy>
      <from>$OpParseFilesOut.Response/xlf:xliff</from>
      <to>$OpLeverageIn.Request/xlf:xliff</to>
    </copy>
    <copy>
      <from>$inputVar.Request/ns2:targetLanguage</from>
      <to>$OpLeverageIn.Request/targetLang</to>
    </copy>
  </assign>
  <invoke name="InvokeLeveraging" partnerLink="Leveraging" operation="opLeverageXlf"
xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/leveraging" portType="tns:ptLeveraging" inputVariable="OpLeverageIn" outputVariable="OpLeverageOut"/>
  <assign name="Assign5">
    <copy>
      <from>$OpLeverageOut.Response/xlf:xliff</from>
      <to>$OpSubmitForTranslationIn.Request/xlf:xliff</to>
    </copy>
    <copy>
      <from>$inputVar.Request/ns2:targetLanguage</from>
      <to>$OpSubmitForTranslationIn.Request/targetLang</to>
    </copy>
  </assign>
  <invoke name="InvokeSubmitForTranslation" partnerLink="Translation"
operation="opSubmitForTranslationXlf"
xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/translation" portType="tns:ptTransla-

```

```

tion" inputVariable="OpSubmitForTranslationIn" outputVariable="OpSubmitForTranslation-
Out"/>
  <assign name="Assign6">
    <copy>
      <from>$OpSubmitForTranslationOut.Response/id</from>
      <to>$OpGetTranslationIn.Request/id</to>
    </copy>
  </assign>
  <invoke name="InvokeGetTranslation" partnerLink="Translation" operation="op-
GetTranslation" xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/translation"
portType="tns:ptTranslation" inputVariable="OpGetTranslationIn" outputVariable="Op-
GetTranslationOut"/>
  <assign name="Assign7">
    <copy>
      <from>$OpGetTranslationOut.Response/xf:xliff</from>
      <to>$OpSubmitForReviewIn.Request/xf:xliff</to>
    </copy>
    <copy>
      <from>$inputVar.Request/ns2:targetLanguage</from>
      <to>$OpSubmitForReviewIn.Request/targetLang</to>
    </copy>
  </assign>
  <invoke name="InvokeSubmitForReview" partnerLink="Review" operation="opSubmitFor-
ReviewXlf" xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/review" portType="tns:ptRe-
view" inputVariable="OpSubmitForReviewIn" outputVariable="OpSubmitForReviewOut"/>
  <assign name="Assign8">
    <copy>
      <from>$OpSubmitForReviewOut.Response/id</from>
      <to>$OpGetReviewIn.Request/id</to>
    </copy>
  </assign>
  <invoke name="InvokeGetReview" partnerLink="Review" operation="opGetReview"
xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/review" portType="tns:ptReview" in-
putVariable="OpGetReviewIn" outputVariable="OpGetReviewOut"/>
  <assign name="Assign3">
    <copy>
      <from>$OpGetReviewOut.Response/xf:xliff</from>
      <to>$OpGenerateTargetIn.Request/xf:xliff</to>
    </copy>
  </assign>
  <invoke name="InvokeTargetGeneration" partnerLink="TargetGeneration"
operation="opGenerateTargetXlf" xmlns:tns="http://tcd.ie/nds/swl10n/ws/contract/tar-
getgeneration" portType="tns:ptTargetGeneration" inputVariable="OpGenerateTargetIn"
outputVariable="OpGenerateTargetOut"/>
  <assign name="Assign4">
    <copy>
      <from>$OpGenerateTargetOut.Response/targets/fileName</from>
      <to>$outputVar.Response/ns2:targets/ns2:fileName</to>
    </copy>
    <copy>
      <from>$OpGenerateTargetOut.Response/targets/fileContent</from>
      <to>$outputVar.Response/ns2:targets/ns2:fileContent</to>
    </copy>
  </assign>
  <reply
name="end"
partnerLink="Localisation"
operation="opWfTranslate"
portType="ns1:ptTranslation"
variable="outputVar">
  </reply>
</sequence>
</process>

```

Appendix 4: Is XLIFF in the right place?

1. Introduction

XML Localisation Interchange File Format (XLIFF) [1] was first standardized by OASIS in 2002. Since its inception there have been debates in the localisation community on how to position XLIFF to maximize the benefits of having a standard file format. This is not a trivial question in so far as the answer influences the goals laid out for the standard.

2. Positioning XLIFF

2.1. Between development and localisation

One option is to position XLIFF as a bridge between development groups and the localisation function (internal or external). In a software localisation workflow (see Figure 1) the input files are normally supplied by development teams through some type of content management system (CMS) usually a version control system. These files can be in any kind of file format (standard or proprietary) though generally are localisation friendly formats such as .properties, .po, .dll, etc. The initial step in the workflow involves extracting the localisation relevant information such as translatable content, sizing information, etc. This requires specialised parsers for each of the input file formats. All software localisation tools come with a number of parsers for common file formats and most support a plug-in architecture to incorporate custom built parsers catering for proprietary formats. Once the localisable content has been parsed, the next steps in the localisation workflow can happen. At a very high level this normally includes a pre-translation (segmentation, leveraging, machine translation) and a translation phase, plus a test / fix cycle where localisation defects are detected and corrected. Finally the translated versions of the input files are generated and added to the development group's CMS (target file generation may also happen during the test/fix cycle, where localised versions of the running software are produced to be used by the test team). Target file generation requires, like parsing, software components to handle each supported file format.

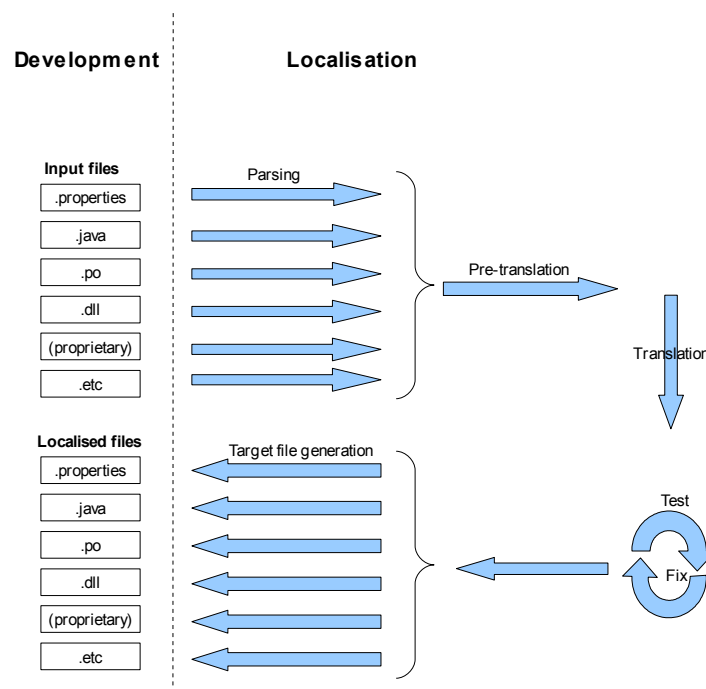


Figure 1

In this context XLIFF can be used as a funnel between the different file formats used by development groups and the software localisation tools (see Figure 2). This option introduces additional overhead as two extra transformations are now needed: between the input files and XLIFF and, at the end of the workflow, between XLIFF and the localised versions of the input files. But it also brings a number of advantages. One of them is to ensure a basic level of internationalization support. This shows in at least two areas:

- Producing an XLIFF file ensures that the localisable content has been identified and separated from the developer's source code.
- It provides localisation with a consistent and standard character encoding such as Unicode (XLIFF uses UTF-8 or UTF-16). Any encoding conversions are handled by the transformation to/from XLIFF.

This basic level of i18n support is already achieved by localisation friendly formats such as .properties or .po files. But it is not a given for proprietary or application specific file formats (e.g. the ones used by some Installer technologies).

Another advantage is that having a single consistent input (XLIFF) to the localisation workflow can open opportunities for increased automation. This advantage is more relevant for companies with a diverse set of development groups using different programming frameworks. This is normally the case for large Independent Software Vendors (ISV) where there may over a hundred file formats that require localisations.

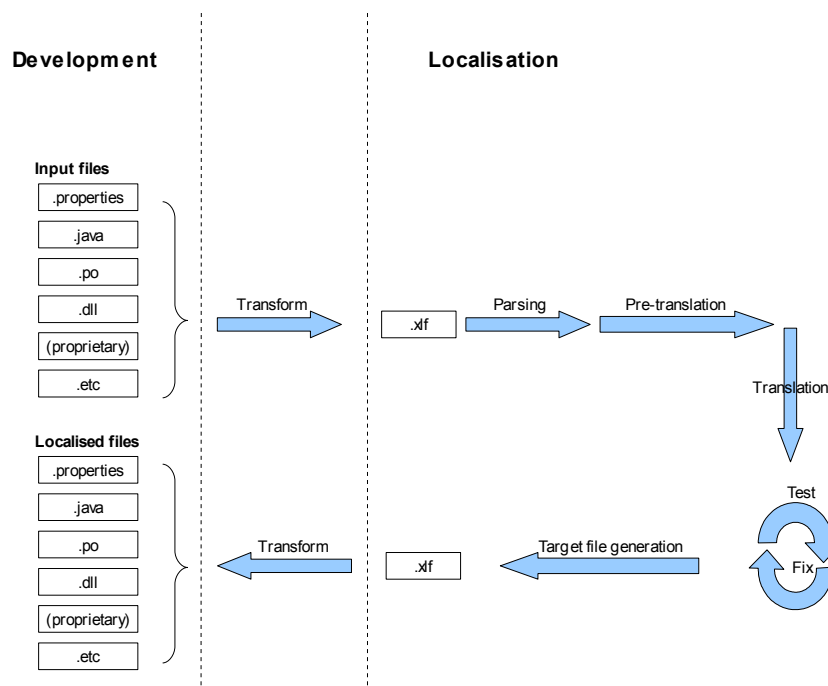


Figure 2

Using XLIFF as a bridge between development and localisation also brings some opportunities. Specifically it allows localisation departments to push the burden of the extra transformations on the development groups. This is how some large ISV are using XLIFF. The benefits of this approach, beyond the obvious one to the localisation departments of offloading some tasks to development, is a) to make clear that development is responsible for i18n and as such it needs to ensure that the product is localisable in a efficient manner, and b) a more streamlined localisation process. In exchange it gives development groups great flexibility when choosing development

frameworks or tools: they are no longer limited to the list of 'supported by localisation' file formats. So long as they can supply and accept XLIFF files they are free to pick any emerging technology or non-standard file formats.

There are some drawbacks to using XLIFF positioned in this way. No matter who owns the transformations to/from XLIFF these transforms add an extra overhead to the workflow. This translates in extra complexity as well as added processing, and the organization as a whole, regardless of which specific function or team, will have to contend with it.

Parsers for most of the file formats used in localisation are readily available and have been for a long time. Many of these file formats have been designed for localisation and, even though they may lack the sophistication of XLIFF, have widespread and mature parser support. In fact XLIFF tries to solve many of the same problems that were tackled by .po, .resx or .properties files. That's why, ideally, if XLIFF is positioned between development and localisation it should be a substitute for those file formats and not an intermediary. But the localisation file formats used for Java (.properties), C# (.resx), etc. are tightly tied to their programming language specifications and it is very unlikely they'll change. The only attempt of doing so that I'm aware of involves a project that aimed to substitute .po files with XLIFF(see [7]), but so far it hasn't gained significant traction.

2.2. Between localisation tools

A different way to position XLIFF is as a bridge between software localisation tools (Figure 3). In this way an XLIFF file would act as an interchange file format not between the files from development (.properties, .po, etc.) and, effectively, the software localisation tools, but between the tools themselves.

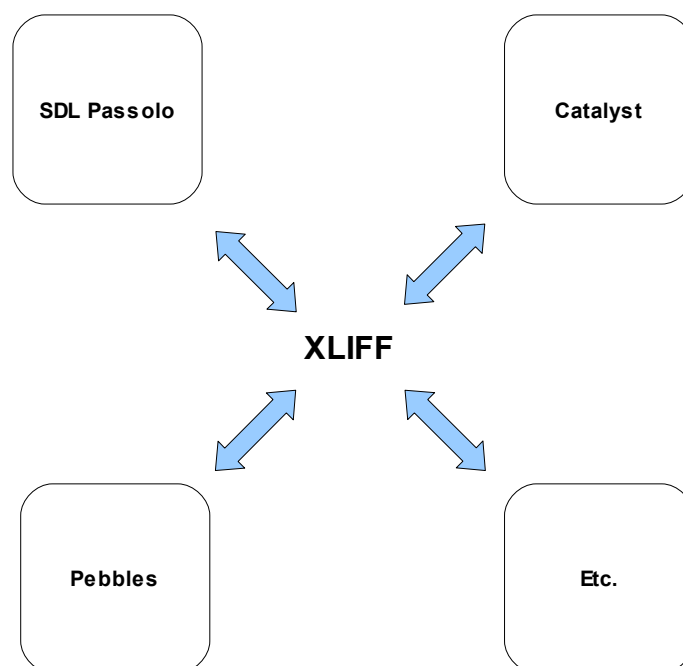


Figure 3

Each software localisation tool uses its own kind of file repository to store the contents of a localisation project. A project repository, such as a .tk file created with Catalyst, includes the localisable information extracted from the input files, the translations and metadata such as transla-

tion status for each string, annotations, skeleton files, etc. These project repositories conform to proprietary file formats owned by the localisation tools vendors. In this context XLIFF can be used as the interchange file format that allows to transfer the contents of one proprietary project repository to another. To support this positioning a software localisation tool needs to provide export / import functionality between their native project repositories and XLIFF.

Used in this way XLIFF becomes the basis for tool interoperability. The advantage of this positioning is lessened tool lock-in with the corresponding flexibility to choose the right tool depending on factors such as the project characteristics, training needs, etc.

A related advantage is that the different players involved in a software localisation project can select software localisation tools independently. Generally Localisation Service Providers (LSP) have to use the localisation tool chosen by the client. Tool interoperability facilitates scenarios where the LSP and the client companies can use different tool sets and still be able to effectively transfer project information.

Using XLIFF as a bridge between software localisation tools does not preclude using it also as a bridge between development and localisation, but it doesn't require it. In fact this use of XLIFF is orthogonal to the localisation workflow posing no restrictions on it.

2.3. Between localisation functions

Most software localisation tools available in the market today are desktop applications designed as one-stop solutions. But widespread increases in network bandwidth as well as the spread of use of cloud computing is creating the basis for new ways to deliver localisation services. In this new model localisation functions such as parsing, leveraging, segmentation, machine translation, QA, etc. can be provided over the network (whether an intranet or the internet).

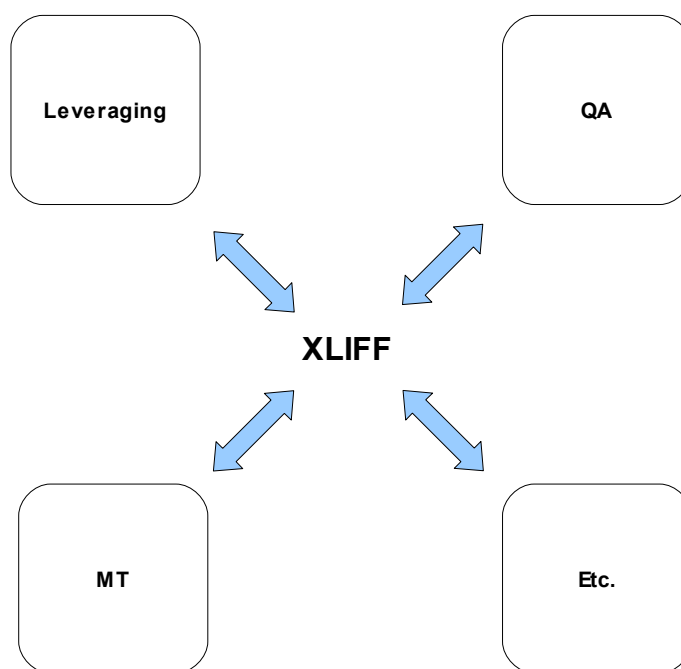


Figure 4

This scenario is closely related to the one presented in Figure 3. In fact, it can be seen as an evolution, where software localisation tools are replaced by software localisation functions delivered as Web Services. This emerging framework promises a more flexible architecture, increased

automation and reductions in process management overhead through the use of automated workflows. But it will require reimplementing existing functionality and new, cloud based, tools. Web Services, either SOAP or REST, exchange information encoded as XML documents. In this scenario XLIFF becomes the transport payload used by software localisation Web Services.

2.4. As a localisation project repository

The mention of this positioning is somehow anecdotal. Though some localisation tools (see [8]) use .xlf files as their native repository, trying to use XLIFF in this way showcases the limitations of XML as a database file format: lack of indexes, lack of efficient storage, lack of transactions, etc. When using XLIFF in this way performance constraints impose limits to the size of projects a tool can handle efficiently.

The main software localisations tools, commercial or in-house (Catalyst, SDL Passolo, NLWorkbench, etc.) all use binary formats for their project serializations. These may be proprietary formats or database repositories (MySQL, SQL Server, etc.) but they provide better scalability and are all optimized for the types of operation expected of a modern integrated localisation environment.

3. A look at the standard

The latest version of the XLIFF specification is 1.2, released in February 2008 [1]. The XLIFF Technical Committee has also released three related documents as committee drafts. These are the representation guides for HTML [2], Java Resource Bundles [3] and PO [4] files.

The different ways, discussed before, in which XLIFF can be positioned make different demands on the standard. This is, the data representation needs are different depending on how and what for XLIFF is used.

If XLIFF is used as a bridge between development and localisation (as described in section 2.1) the data that needs to be represented is the localisable content included in the input files. There are many such input file formats with a great deal of variability on what localisable data they include, how the data is represented and the associated metadata. Some, such as .properties files, can only describe string-tables made up of simple pairs of identifier-string; others, such as .rc or .resx, can describe string-tables and other resource types such as dialogs/forms, menus, etc. Thus they not only include strings but also other localisable information such as sizing, position information etc.

These file formats not only differ on what they include but also on how they represent localisable data. For example, it's common for software strings to include variables that are replaced at run-time with live data. These variables become non-editable codes inside the strings and are represented differently depending on the file format (e.g. {0}, %s, etc.). The same apply to other non-textual elements such as format specifiers, formatting codes, etc.

Finally, the associated metadata incorporated in these file formats is also different. While some lack a standard way to supply it, others such as PO files, have specified ways of conveying information such as translator's comments, flags, etc. all of which have localisation value.

To support this variability XLIFF needs to provide both a comprehensive array of options and also the flexibility to pick and choose among them. This is reflected throughout the standard but specially in the <trans-unit> element. This element has 29 attributes of which 28 are optional. It is also reflected in the list of inline elements that can be included in the <source> and <target> elements. There are 8 types of inline elements catering for codes with/without begin and end tags, codes that cross or not <trans-unit> boundaries, etc.

This flexibility allows the specification to support a large number of file formats. But it also creates the potential for interoperability problems. Filters extracting content from the input files into an XLIFF document can represent the same extracted data in different ways (all of them compliant). This is the problem that the representation guides for HTML, Java Resource Bundles and PO files are trying to solve. They specify a unique way to translate the localisable data inside those file formats into an .xlf file. But the need for representation guides points to a bigger problem. There are many file formats that lack a representation guide and there will be new formats coming as new programming languages appear. One option can be to expand the number of representation guides and to keep maintaining the existing ones. But this poses a considerable maintenance burden. Another option is to use 'vanilla' XLIFF; this is, for filters to dismiss any localisable data beyond the source strings and just use the minimum mandatory elements and attributes required to produce an XLIFF document. This is certainly an option for file formats that only describe string-tables. But doing it with file formats capable of describing other resource types would limit the functionality that localisation tools can offer. For example, dismissing the form information contained in a .resx file (position, size, etc.) would prevent a tool from providing a form visual editor.

So XLIFF's flexibility has a double edge; it expands the range of formats supported by the standard but it increases the scope for interoperability issues.

If XLIFF is used as a bridge between software localisation tools (as described in section 2.2) the data that needs to be represented in XLIFF are the contents of a localisation project. This includes the information extracted from the originals files and also, potentially, the results of the pre-translation, translation and the test-fix cycle. XLIFF has a rich and flexible feature set, but because of its flexibility it is difficult to see how interoperability can be achieved without a representation guide. A localisation project repository will always contain information that is tool specific. Tools differentiate themselves through their feature set and this is one of the basis for their competitive advantage. XLIFF does not need to relay all these uses but should reflect the consensus on the core information that all tools need when importing a localisation project. A representation guide for a localisation project would provide a clear target for the development of the import / export functionality needed to use XLIFF as a intermediary between software localisation tools.

It is interesting to note that there are two different scenarios where XLIFF could be used to transfer a software localisation project between different tools: intra-project (during the life of a project) or inter-project (in between the end of the localisation cycle for a version of a product and the start of the localisation of the next version). The demands on XLIFF are different in each case. There is metadata, such as history logs, available fuzzy matches, etc. that, while relevant during a localisation project, can be and usually is discarded when the project data is archived at the project's end. Generally, the information archived at the end of a project is a subset of the overall information used during a project.

4. Conclusion

Support for standards is generally regarded as a valuable feature. Accordingly many software localisation tools include support for XLIFF (see Table 1). In some cases the support extends to accepting XLIFF files as Translation Memories (for pre-translation), but generally the support consists of including XLIFF as one of the supported input files (Table 2). This is the support required to use XLIFF as a bridge between development and localisation and, in practical terms, involves providing a parser as well as generating target XLIFF files.

Name	Standard support
Alchemy Catalyst www.alchemysoftware.ie	TMX, XLIFF
SDL Passolo www.sdl.com	TMX, XLIFF
RC-Wintrans www.schaudin.com	TMX
Multilizer www.multilizer.com	TMX, XLIFF
Sisulizer www.sisulizer.com	TMX, XLIFF
Visual Localizer www.visloc.com	TMX
Lingobit Localizer www.lingobit.com	No
Heartsome www.heartsome.net/EN/xlfedit.html	XLIFF, TMX, TBX, SRX
VB Language Manager Pro www.whippleware.com/vblm.htm	No

Table 1

But none of the tools listed in Table 2 offer the type of export/import functionality to/from XLIFF that can be used for round-trip conversions between their project repository and an XLIFF document. That is, no software localisation tool attempts to use XLIFF as the basis for tool interoperability. There are a number of possible reasons for this; from a business perspective it may be safer to support XLIFF only as an input file; also there are questions about the maturity of the standard. In any case, the end result is that XLIFF usage is limited to that of intermediary between localisation and development. But the benefits of using XLIFF this way are limited which may explain the corresponding limited uptake of XLIFF across the industry. Issues such as increased overhead or lack of representation guides make XLIFF a less than obvious choice except, maybe, for niche file formats without existing parser support.

Name	XLIFF Input file	XLIFF Import/Export
Alchemy Catalyst www.alchemysoftware.ie	Yes	No
SDL Passolo www.sdl.com	Yes	No
RC-Wintrans www.schaudin.com	No	No
Multilizer www.multilizer.com	Yes	No
Sisulizer www.sisulizer.com	Yes	No
Visual Localizer www.visloc.com	No	No
Lingobit Localizer www.lingobit.com	No	No
Heartsome www.heartsome.net/EN/xlfedit.html	Yes	No
VB Language Manager Pro www.whippleware.com/vblm.htm	No	No

Table 2

A move towards positioning XLIFF as the interoperability standard for software localisation tools would open some new and interesting possibilities. In the short term it would lessened tool lock-in by reducing the cost of transitioning between tool sets. Eventually it would allow the different agents that participate in the localisation workflow (translators, l10n engineers, l10n project managers, etc.) to choose the tool that better suit their different needs, independently of each other. It could also become the basis for delivering software localisation web services.

But for this to happen the standard needs to provide an unambiguous target that minimizes the scope for incompatibilities. As it is XLIFF is too flexible to provide such a target. This flexibility is reflected in the structure, the extensibility mechanisms and the user defined values. In fact the standard seems aware of the issues that this flexibility may generate: “It is strongly recommended that content within the <file> element be uniformly bilingual” [1], “It is strongly recommended to use XLIFF capabilities whenever possible, rather than to create non-standard user-defined elements or attributes” [1].

Defining a clear target for localisation tools vendors can be achieved in several ways. Maybe the standard can be changed; or maybe the standard can remain flexible but offer a representation guide. Whichever the way this would allow XLIFF to increase its value to the localisation industry.

References

- [1] XLIFF Version 1.2, OASIS Standard, 1 February 2008, accessed from <http://docs.oasis-open.org/xliff/xliff-core/xliff-core.html> on 03 Apr 2010
- [2] XLIFF 1.2 Representation Guide for HTML, OASIS Committee Draft, 16 October 2006, accessed from <http://docs.oasis-open.org/xliff/v1.2/xliff-profile-html/xliff-profile-html-1.2-cd02.html> on 03 Apr 2010
- [3] XLIFF 1.2 Representation Guide for Java Resource Bundles, OASIS Committee Draft, 30 April 2007, accessed from <http://docs.oasis-open.org/xliff/v1.2/xliff-profile-java/xliff-profile-java-v1.2-cd02.html> on 03 Apr 2010
- [4] XLIFF 1.2 Representation Guide for Gettext PO, OASIS Committee Draft, 16 October 2006, accessed from http://docs.oasis-open.org/xliff/v1.2/xliff-profile-po/xliff-profile-po-1.2-cd02.html#s.general_considerations.potextraction on 03 Apr 2010
- [5] TMX 1.4b Specification OSCAR Recommendation, Localisation Industry Standards Association, 26 April 2005, accessed from <http://www.lisa.org/fileadmin/standards/tmx1.4/tmx.htm> on 03 Apr 2010
- [6] XLIFF “A white paper on version 1.2 of the XML Localisation Interchange File Format (XLIFF)”, Revision: 1.0, 17 Oct 2007 accessed from <http://xml.coverpages.org/XLIFF-Core-WhitePaper200710-CSv12.pdf> on 03 Apr 2010
- [7] XLIFF Tools, accessed from <http://xliff-tools.freedesktop.org/wiki/> on 03 Apr 2010
- [8] Transolutions, accessed from <http://transolution.python-hosting.com/> on 03 Apr 2010

Appendix 5: Extract from XLIFF v1.2

Value	Description
asp	Indicates Active Server Page data.
c	Indicates C source file data.
cdf	Indicates Channel Definition Format (CDF) data.
cfm	Indicates ColdFusion data.
cpp	Indicates C++ source file data.
csharp	Indicates C-Sharp data.
cstring	Indicates strings from C, ASM, and driver files data.
csv	Indicates comma-separated values data.
database	Indicates database data.
documentfooter	Indicates portions of document that follows data and contains metadata.
documentheader	Indicates portions of document that precedes data and contains metadata.
filedialog	Indicates data from standard UI file operations dialogs (e.g., Open, Save, Save As, Export, Import).
form	Indicates standard user input screen data.
html	Indicates HyperText Markup Language (HTML) data - document instance.
htmlbody	Indicates content within an HTML document's <body> element.
ini	Indicates Windows INI file data.
interleaf	Indicates Interleaf data.
javaclass	Indicates Java source file data (extension '.java').
javapropertyresourcebundle	Indicates Java property resource bundle data.
javalistresourcebundle	Indicates Java list resource bundle data.
javascript	Indicates JavaScript source file data.
jscrip	Indicates JScript source file data.
layout	Indicates information relating to formatting.
lisp	Indicates LISP source file data.
margin	Indicates information relating to margin formats.
menufile	Indicates a file containing menu.
messagefile	Indicates numerically identified string table.
mif	Indicates Maker Interchange Format (MIF) data.
mimetype	Indicates that the datatype attribute value is a MIME Type value and is defined in the mime-type attribute.
mo	Indicates GNU Machine Object data.

msglib	Indicates Message Librarian strings created by Novell's Message Librarian Tool.
pagefooter	Indicates information to be displayed at the bottom of each page of a document.
pageheader	Indicates information to be displayed at the top of each page of a document.
parameters	Indicates a list of property values (e.g., settings within INI files or preferences dialog).
pascal	Indicates Pascal source file data.
php	Indicates Hypertext Preprocessor data.
plaintext	Indicates plain text file (no formatting other than, possibly, wrapping).
po	Indicates GNU Portable Object file.
report	Indicates dynamically generated user defined document. e.g. Oracle Report, Crystal Report, etc.
resources	Indicates Windows .NET binary resources.
resx	Indicates Windows .NET Resources.
rtf	Indicates Rich Text Format (RTF) data.
sgml	Indicates Standard Generalized Markup Language (SGML) data - document instance.
sgmltdt	Indicates Standard Generalized Markup Language (SGML) data - Document Type Definition (DTD).
svg	Indicates Scalable Vector Graphic (SVG) data.
vbscript	Indicates VisualBasic Script source file.
warning	Indicates warning message.
winres	Indicates Windows (Win32) resources (i.e. resources extracted from an RC script, a message file, or a compiled file).
xhtml	Indicates Extensible HyperText Markup Language (XHTML) data - document instance.
xml	Indicates Extensible Markup Language (XML) data - document instance.
xmltdt	Indicates Extensible Markup Language (XML) data - Document Type Definition (DTD).
xsl	Indicates Extensible Stylesheet Language (XSL) data.
xul	Indicates XUL elements.

Appendix 6: Performance measurements

Local host measurements:

Wordcounts	1,000	5,000	10,000	50,000	100,000	500,000
Test 1	0.061	0.198	0.244	0.571	1.379	4.574
Test 2	0.030	0.051	0.087	0.494	0.725	4.708
Test 3	0.023	0.046	0.084	0.383	0.857	3.982
Test 4	0.018	0.046	0.085	0.489	0.682	4.808
Test 5	0.017	0.047	0.088	0.346	0.889	4.060
Average	0.030	0.078	0.118	0.457	0.906	4.426

Tests measured in seconds

Remote host measurements:

Wordcounts	1,000	5,000	10,000	50,000	100,000	500,000
Test 1	0.145	0.509	0.908	3.262	6.729	32.051
Test 2	0.151	0.460	0.711	3.238	6.840	32.357
Test 3	0.216	0.452	0.717	3.150	6.898	32.316
Test 4	0.152	0.394	0.762	3.031	6.779	30.732
Test 5	0.171	0.806	0.749	3.148	6.103	31.546
Average	0.167	0.524	0.769	3.166	6.670	31.800

Tests measured in seconds