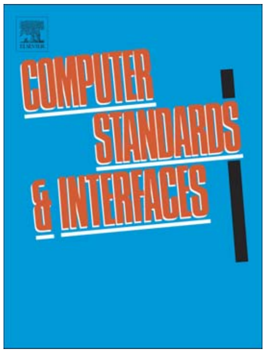

COMPUTER STANDARDS & INTERFACES

An empirical framework for evaluating interoperability of data exchange standards based on their actual usage: A case study on XLIFF

Asanka Wasala, Jim Buckley, Reinhard Schäler, Chris Exton

PII: S0920-5489(15)00068-9
DOI: doi: 10.1016/j.csi.2015.05.006
Reference: CSI 3038

To appear in: *Computer Standards & Interfaces*

Received date: 18 November 2014
Revised date: 27 April 2015
Accepted date: 24 May 2015

An empirical framework for evaluating interoperability of data exchange standards based on their actual usage: A case study on XLIFF

Asanka Wasala*, Jim Buckley, Reinhard Schäler, Chris Exton

Department of Computer Science and Information Systems, University of Limerick, Limerick, Ireland

Abstract

Data exchange formats play a prominent role in facilitating interoperability. Standardization of data exchange formats is therefore extremely important. In this paper, we present two contributions: an empirical framework called XML-DIUE, for evaluating data exchange format standards in terms of their usage and an illustration of this framework, demonstrating its ability to inform on these standards from their usage in practice. This illustration is derived from the localization domain and focuses on identifying interoperability issues associated with the usage of XML Localization Interchange File Format (XLIFF), an open standard data exchange format.

The initial results from this illustrative XLIFF study suggest the utility of the XML-DIUE approach. Specifically they suggest that there is prevalent ambiguity in the standard's usage, and that there are validation errors across 85% of the XLIFF files studied. The study also suggests several features for deprecation/modularization of the standard, in line with the XLIFF Technical Committee's deliberations, and successfully identifies the core features of XLIFF.

Keywords: Empirical framework, Interoperability, Data Exchange Formats, XLIFF, XML, Localization

*Corresponding author at: Localisation Research Centre, Department of Computer Science and Information Systems, University of Limerick, Limerick, Ireland

Email addresses: Asanka.Wasala@ul.ie (Asanka Wasala), Jim.Buckley@ul.ie (Jim Buckley), Reinhard.Schaler@ul.ie (Reinhard Schäler), Chris.Exton@ul.ie (Chris Exton)

1. Introduction

The most widely used definition for interoperability is the definition by the IEEE [1]:

> "Interoperability is the ability of two or more systems or components to exchange information and to use the information that has been exchanged."

Interoperability is becoming increasingly important in heterogeneous environments as it facilitates the integration of different entities such as tools, businesses, technologies and processes. Data exchange formats play a prominent role in facilitating interoperability by providing agreed or standardized notations for storage and exchange of data. Data exchange formats that are based on the Extensible Markup Language (XML) are becoming ever-more pervasive [2, 3] and they can be categorized as either open or proprietary. Examples of popular XML-based open data exchange standards (also known as open file formats) include: XHTML, DOCBOOK, and Office Open XML (OOXML). However, the definition of such standards is an arduous time-consuming process due to the constantly evolving nature of the technologies, businesses, and tools in question [4]. That is, standards need to be constantly reviewed and updated to cope with the changing requirements, typically across multiple organizations.

In this paper, we propose a novel empirical framework that can be used as a tool to evaluate the usage of data-exchange, XML-based standards and thus inform on the development, maintenance and evolution of those standards. The utility of this framework is illustrated by applying it to the XML Localization Interchange File Format (XLIFF), an open standard for the exchange of localization data.

The XLIFF standard has been developed and is being maintained by a Technical Committee (TC) of OASIS and is an important standard for enabling interoperability in the localization domain (see Section 2.1.1 for more details on the XLIFF standard). It aims to enable the loss-less exchange of localization-relevant data and metadata between different tools and technologies across the localization process. XLIFF is gaining increased acceptance within the localization community [5], and is increasingly being used not just as an exchange format, but also as a container for the storage and transport of data [6, 7].

XLIFF version 2 was released on the 5th of August 2014 to provide solutions to various significant issues relating to the previous version of the XLIFF standard (version 1.2). However problems remain with respect to adoption, as confirmed by a study conducted in 2011, which revealed that lack of interoperability could cost language service providers more than 20% of their total translation budget. According to this study, the main cause for lack of interoperability is the "lack of compliance to interchange format standards" [8], a finding that suggests the standard may still be immature with respect to adopters' needs.

We aim to evaluate this potential immaturity issue by reporting on experiments where the usage of the XLIFF schema is assessed by our analytical framework. The framework will provide empirical evidence and statistics related to the actual usage of different elements, attributes and attribute values of this standard in-vivo.

More generically, this illustration demonstrates that the XML-DIUE framework proposed can also serve to address similar issues in XML based file format standards in other domains. The empirical results generated by the framework seem useful for identifying important criteria for standard development such as the most frequently used features[1], the least frequently used features, usage patterns and formats. The findings will also be helpful in identifying compliance issues associated with implementations supporting the standard under study. Furthermore, the results will be helpful for the development of interoperability test suites that target prevalent compliance issues, from a usage perspective. Thus, we believe that this framework will ultimately contribute to improved interoperability among tools and technologies in many verticals.

The remainder of the paper is organized as follows: Section 2 discusses related work in standards and localization, culminating in a section devoted to XLIFF. This provides a context for the XLIFF running example used in this paper. Section 3 describes the methodology underlying our framework, illustrating it by detailing the data collection performed in our XLIFF study, and the data analysis performed. Section 4 presents the experimental results

[1] Hereafter, we use the term "features" throughout the paper to refer to XML elements, attributes and predefined element values as well as attribute values.

derived from our illustration which are then discussed in Section 5. This section also outlines some more general limitations of evaluating standards in this fashion. Finally, the paper concludes with a summary and pointers to future work in Section 6.

2. Related work

Standards are crucial to achieve significant aspects of interoperability among diverse systems [9]. In this review, we focus on the evaluation of data-exchange standards. Specifically the review briefly focuses on research that has considered the end-user usage of standards, as this is a core consideration for XML-DIUE. Subsequently, the review targets research evaluating XLIFF, as it is the subject standard for our illustratory case study.

2.1. Usage information in standard evaluation

Soderstrom [10] points out that the users of a standard are highly important in the standard development process; however they are not widely discussed in the literature. Soderstrom goes on to mention that users are “the ones who ultimately decide the success of a standard - by using it or not using it”. Likewise, Lelieveldt [11] argues that “every standard requires certain decisions to be made about their use and realization”. As such, lack of user engagement and feedback adversely affect the standardization process.

Shah and Kesan [12, 13] tried to embody this concern in their evaluation of Office Open XML (OOXML) and Open Document Format (ODF). In their initial holistic evaluation of these formats they identified significant interoperability issues which resulted in the loss of formatting and the loss of content. They hypothesized that the formats lacked full interoperability because users did not need support for all features in their tools. Consequently, they focused on frequently used features of the standard, to assess whether implementations are “good enough for most users”. These routinely used features were identified by “examining various instructional materials for using office productivity software” [13], although the details of their procedure for identifying these features are not given.

Our framework continues in this vein. It facilitates implicit user feedback by analysing the user-generated files of a standard. It not only helps to identify the most frequently used features, but also other important information regarding the usage of a standard, such as the least frequently used features and frequently used features across companies (thus features that are important for cross-organizational interoperability).

In 2005, Google carried out work analogous to that proposed here. They analyzed over a billion HTML/XHTML documents and presented some important findings regarding frequently used HTML elements, attributes, class names, and the average number of unique elements in a web page [14]. For some of these statistics, possible explanations are also given. However, neither important conclusions nor important recommendations have been made regarding improvements to the HTML based on their analysis. Our framework goes beyond Google's work, by making explicit the architecture used to analyze a standard and by designing several standard-usage analysis metrics that describe the connection between statistics and their significance for the development process of a standard.

2.2. The XLIFF standard in localization

In this research we will focus on identifying interoperability issues related to the XLIFF standard. The following section (Section 2.2.1) gives a brief introduction to the XLIFF standard. In Section 2.2.2 we summarize relevant literature mainly focusing on interoperability issues related to XLIFF.

2.2.1. The XML Localization Interchange File Format

The XLIFF standard was first developed in 2001 by a technical committee formed by representatives of a group of companies including Oracle, Novell, IBM/Lotus, Sun, Alchemy Software, Berlitz, Moravia-IT, and ENLASO Corporation (formerly the RWS Group). In 2002, the XLIFF specification was formally published by the Organization for the Advancement of Structured Information Standards (OASIS) [15, 16].

The purpose of XLIFF as described by OASIS is to "store localizable data and carry it from one step of the localization process to the other, while allowing interoperability between tools." By using this standard, localization data can be exchanged between different companies, organizations, individuals or tools.

XLIFF was originally developed to address the problems arising from the huge variety and ever-growing number of source formats, and has increasingly been used to store and exchange localization data. The XLIFF standard is supported by many tools [17] and is being continuously developed further by the OASIS XLIFF Technical Committee (TC). This TC has very recently published XLIFF 2.0[2], a major release of the standard [18] after six years of the previous version, XLIFF 1.2. XLIFF 2.0 is backward incompatible with XLIFF 1.2 and in this research we focus on XLIFF 1.2, based on the prevalence of XLIFF 1.2 at this time.

2.2.2. Evaluation of the interoperability of the XLIFF standard

A survey conducted by Morado-Vázquez and Filip [19] reports the status of XLIFF support in computer-aided translation (CAT) tools. This report tracks quarterly changes in XLIFF support in all major CAT tools. The survey was based on a questionnaire designed by the XLIFF Promotion and Liaison Sub-Committee of the XLIFF TC and it is aimed at CAT tool producers. The survey reports detailed characterizations of these tools with respect to XLIFF version support, use of custom extensions and XLIFF element and attributes support. In their survey, they avoid the use of the word "support" due to its ambiguous and prompting nature. Instead, they used the phrase "actively used elements" during the data collection phase. Only "Yes" and "No" answers have been collected. As such, the level of tool support for a certain element or attribute is questionable (for example, given a tool "actively uses" the `<file>` element, it does not necessarily imply that it conforms to the XLIFF mandatory requirements for the `<file>` element).

Bly [17] analyzed XLIFF support in commonly used tools and presented a matrix containing tools and their level of support for individual XLIFF elements. The level of support is divided into three categories: no support, partial support and full support. Unfortunately, again the author has not defined the terms "support" and "partial-support" precisely. This can lead to confusion (for example, given the fact that tool *T* supports element *E*, does this imply that *T* can process all the attributes of *E*?). Furthermore, the work, as presented, lacks significant details. For example, the methodology used to conduct the study is not revealed. Likewise, the development of the test suites has not been described and the test suites have not been published. This makes the study difficult to replicate and refine.

Nevertheless, Bly [17] contributes valuable insights: He concludes that tool vendors can conform to standards but still lock-in users to their tools. Moreover, he discusses various problems associated with the XLIFF standard, such as its inability to support all the features offered by tools, and its lack of

[2] On the 5th of August 2014

tight definitions. He also points out tools' ability to achieve the same effect using different tag configurations and semantics, suggesting an ambiguity in the standard. Notwithstanding, he is convinced that "XLIFF is the best (and only) hope for interoperability" in this domain.

In another study, Morado-Vázquez and Wolff [20] proposed a "weighted sum model" as a possible improvement to Bly's methodology: They highlight the importance of elements' attributes and attribute values for tool interoperability [20]. Our work can be considered as contributing to this agenda by providing usage information that may be used in calculating or validating such a weighting.

Wasala et al. [21] present an extensive comparison of XLIFF with a similar proprietary file format, the Localization Content Exchange (LCX), Microsoft's internal localization file format. They discuss interoperability issues between them. They also highlight interoperability issues associated with the XLIFF standard and propose improvements to both the XLIFF standard and the LCX file formats. Likewise Imhof [22] provides an overview of limitations of XLIFF. These are mainly related to XLIFF's extensibility features, segmentation approach, and inline elements. Imhof [22] identifies three levels of XLIFF support in tools: tools belonging to the 1st level treat XLIFF files as just XML files; tools belonging to the 2nd level support XLIFF partially (i.e. these tools are capable of interpreting source and target content, but only a limited set of features of the XLIFF standard are implemented); and tools with full XLIFF support are grouped into the 3rd level. Imhof [22] performed an analysis similar to [17] but also fails to describe the methodology followed in his study.

In Anastasiou and Morado-Vázquez [23] several interoperability tests were performed with three XLIFF compliant Computer Aided Translation (CAT) tools. Like Bly [17], they classified selected tools into two categories: XLIFF Converters (i.e. generators) and XLIFF editors. Out of the three CAT tools selected, they found that two had the capability to generate XLIFF content and three had the capability to edit XLIFF content, so they were interested in 4 combinations: for each converter they wanted to see if the other 2 editing tools could edit the generated content. The researchers' methodology involved 5 steps:

1. conversion of a selected HTML file into XLIFF (using the two converters);
2. validation of the converted XLIFF file;
3. manipulation of the XLIFF file using the editors;

4. manual analysis of the XLIFF file;
5. back conversion of the file into HTML and a manual analysis of the converted file.

The results showed that out of the four combinations (i.e. XLIFF generators and editors) considered in this research, only one pair of tools seems to interoperate successfully. The authors recommend "simplicity, better communication between localization stakeholders and clarification of specification" of the standard and suggest future work on expanding the experiment with more CAT tools as well as different file types. It should also be noted that their experiment only considers the back-conversion of the XLIFF files using the tool used to generate the XLIFF file. A better analysis could be carried out if all possible scenarios were taken into account during the back-conversion process.

The authors of that study built on Bly's work to identify several limitations of the XLIFF file format, again mostly related to interoperability. Among the limitations noted are the complexity of the file format (i.e. 38 elements, 80 attributes, 269 pre-defined values), extensibility, and a lack of conformance clauses, reflecting Bly's 'lack-of-definition' finding. In addition, they identify a general lack of awareness of the standard amongst developers. All these findings suggest that usage of the standard should be studied.

Interestingly, our literature review suggested that bottom up analysis of XLIFF constructs in files has been already identified as an approach for addressing issues associated with XLIFF in [24]. Unfortunately, this proposal lacks details. Specially, a systematic methodology towards the bottom-up analysis of XLIFF files is not given in their study. We aim to address this gap in our research.

3. The XML-DIUE framework[3]

This section presents the XML Data Interoperability Usage Evaluation (XML-DIUE) framework.

The Parser-Repository-Analysis architecture proposed in the 90s for reengineering toolkits [25, 26] is appropriate here based on the similarity of the concerns. Both involve static analysis (parsing) of structured documents,

[3] Pronounced 'dieu', as in 'adieu'

followed by viewing of interesting information derived from parse-based analysis. The Parser-Repository-Analysis architecture (see Fig. 1) isolates the parsing components from the view-generation components, by persisting the parsed-analysis in the repository. This allows the addition of new viewing tools against the repository without the requirement of building a new parser. Additionally, if the repository can be made agnostic to a new variant of XML to be analyzed, the same suite of viewing tools can be employed without alteration. Only a new parser needs to be developed.

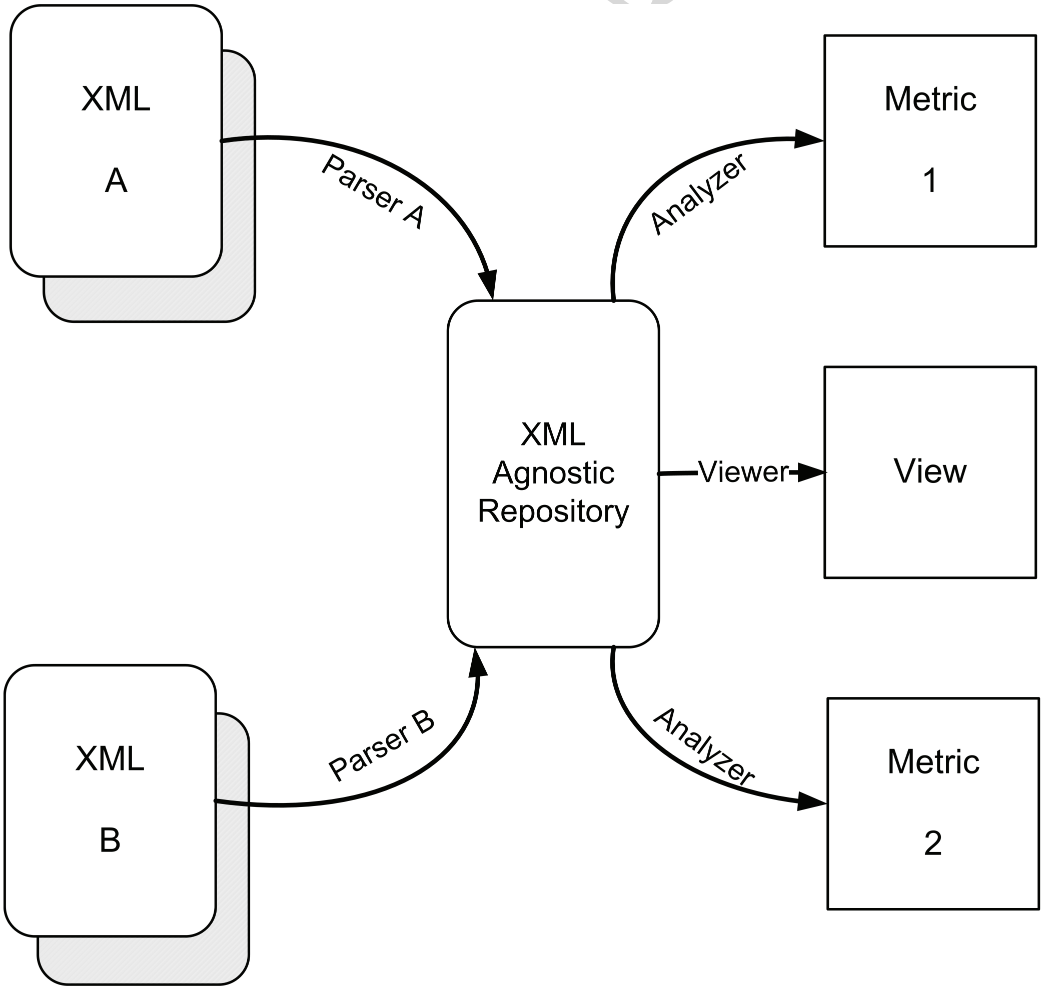

Figure 1: The Parser-Repository-Viewer architecture proposed for XML-DIUE.

This section describes the methodology used for the XML-DIUE framework and each stage of the methodology is illustrated by means of the demonstrative XLIFF case-study performed, as an open, explicit protocol enhances replicability [27]. The methodology involves the construction of an in-vivo corpus of the file type to be analyzed (in this case the XLIFF XML files that are positioned at the left-most column in Fig. 1); the extraction of data from the collected files; the construction of the repository, data profiling by means of designing usage-analysis criteria, the development of analyses/queries off the repository and finally, the presentation of the results. The core step in this process is "designing usage-analysis criteria" and this will be covered in the greatest detail. However, the above steps are all discussed in the following sections, with reference to the running example from the localization domain.

3.1. Construction of a corpus

The first step in our process involves building an authentic reference corpus of standard-based files. These can be from industry or open-source (OS), but their origin should be noted in the XML-DIUE framework protocol. Industry offers the potential of insight into commercial use, but OS software is becoming increasingly prevalent in many verticals [28]. Thus vertical, availability and study's focus should ultimately define the choice of corpus.

Often availability is the dominant factor in gathering commercial data but, if availability is not an issue, the sample should be significant enough to allow external validity and generalization. External validity refers to the sample being representative of the community and conditions of study in a real life context [29]. Generalization refers to the scale of the data: whether it is large enough for statistical techniques to imply that this is representative of a population [29]. It should be noted that in studies of data exchange standards, the focus is often on the attributes and elements in the files. So while both concerns refer to the file sample, they also refer to the number of elements and attributes in the file. Thus the size and representative variety of the files in the corpus are equally important. Measures taken to address both concerns should be reported on for the study, but the corpus size should only be reported after a certain amount of pre-processing (see Section 3.1.1).

The search process should be transparent and, during the gathering of material, good ethical guidelines should also be adhered to. Whenever possible, creators of non-open source XML files should explicitly give their informed consent, and should be made aware of their ability to withdraw at any time. They should also know whether they (and their organizations) are guaranteed anonymity and the conditions under which the data will be held.

In the illustrative XLIFF case study, files were collected using mainly two methods: 1) through industrial partners of the CNGL[4] project and 2) by crawling openly available XLIFF files on the world wide web. The CNGL industrial partners were made aware of this experiment and invited to contribute their XLIFF files. Three industrial partner organizations contributed to the XLIFF corpus under condition of anonymity. In a second step, openly available XLIFF files on the web were scraped on two occasions: on the 26th and 29th of August 2011 respectively. The Google search engine was used for this purpose, mainly because of its popularity and widespread use. Using Google's specific file types search facility (i.e. "filetype:" keyword), all files with the extensions .xlf, .xliff and .xml were initially downloaded, with several additional keywords (e.g. +xliff, +body + "trans-unit") being used to locate XLIFF content and to filter results.

XML files were initially downloaded as XLIFF allows XLIFF documents, or parts thereof, to exist within XML documents [15]. However, these files were quickly discarded due to the increased complexities involved in developing pre-processing and parsing tools to handle them.

During crawling, it was evident that one open source software project *(Eclipse)* uses a significant amount of XLIFF files in its localization process. Therefore, these files were crawled separately from their project sites.

The next step involves pre-processing the files obtained, as described in the following section.

3.1.1. Cleaning and pre-processing

In the illustrative study, manual analysis of random files revealed various issues that demanded pre-processing. These issues included the use of different encoding methods, duplicated content and inconsistencies. Therefore, prior to the analysis stage, the following pre-processing operations were performed on the files:

1. cleaning of the file names (especially some of the crawled files contained file names with special characters). These were manually renamed, from

[4] The Centre for Global Intelligent Content (CNGL) is an Academia - Industry partnership that conducts research related to localization, machine translation and speech technologies. More information about the centre is available at: `http://www.cngl.ie/`

names like “dialog.xlf-spec=svn8-r=8”;

2. removal of non-XLIFF files by analyzing content (especially from the crawled files). The content was analyzed with the aid of a Python script. Filtered non-XLIFF files were manually validated as inappropriate before they were discarded;
3. removal of duplicated files (by analyzing content). A proprietary tool[5] was used for this purpose;
4. removal of Unicode Byte Order Marker (BOM) and conversion of encoding to UTF-8 without BOM (files were found in various encodings such as UTF-16, UTF-8 with BOM, UTF-8 without BOM etc.);
5. removal of XML directive and DOCTYPE declarations at the beginning of a file (some files included either or both of these directives, while others did not);
6. manual extraction of embedded XLIFF content in downloaded HTML web pages;

Resultantly, we constructed what we believe to be the first significant XLIFF corpus available for research, containing 3,179 XLIFF files distributed in the following manner:

- Company A: 38 files;
- Company B: 29 files;
- Company C: 1004 files;
- Crawled: 444 files;
- Crawled from Eclipse project: 1,664 files.

The above files were stored in 5 separate folders based on their source (i.e. the individual companies, those found by crawling and Eclipse).

In order to give an indication of the variation of XLIFF content in our XLIFF corpus, the total file sizes in kilobytes (KB) and their averages were calculated; likewise, the total and average `<trans-unit>` count of individual sources are given below in Table 1. The `<trans-unit>` element has been chosen as it is the most important XLIFF size-indicating element as it encompasses both the source text and its corresponding translations.

[5] Auslogics Duplicate File Finder (freeware) can be obtained from: `http://www.auslogics.com/en/software/duplicate-file-finder/` [accessed 22 April 2015]

Table 1: File sizes and `<trans-unit>` counts of individual sources.

	File size (KB)		`<trans-unit>`	
	total	average	count	average
Company A	8462.40	222.69	19310	508.16
Company B	616.37	21.25	1391	47.97
Company C	17288.10	17.22	20437	20.39
Crawled	40903.56	92.13	124090	249.48
Eclipse	126479.15	76.01	187740	112.82
Overall	193749.58	60.95	353003	111.04

3.2. Data extraction and construction of a database

The next step involves extracting data from the corpus for analysis. In order to persist the data and facilitate analysis (and indeed future, unforeseen analysis) the framework suggests that the data from the XML files be parsed and transferred into a repository. The schema for such a repository could be dependent on the specific data-exchange file format under-study and/or, to a lesser extent, the type of analysis envisaged. However, it should be noted that there is a relationship between the detail of the repository schema, the variety of XML files it can accommmodate and the analysis that can be performed. For example, a less structured schema would allow for a greater variety of XML file types storage but would limit structural analysis. If, incontrast, the schema accommodated both detailed and structural information, then additional analysis tools could be added later to probe the structural aspects of the XML files without any additional parsing effort. The parsing method and resultant schema should be made explicit and justified.

In our study, the XLIFF corpus could not be analyzed using ordinary corpus analysis tools and so in-house analysis tools were developed to populate the repository. Python scripts were written to populate a database of XLIFF information. These scripts processed one file at a time iterating through all the files in the corpus. The ultimate objective then was to run various SQL queries on the database, analyzing the XLIFF content. In this case the database schema was simple, consisting of four database tables (as below) that allowed for count-based analysis of various XLIFF features and value-based analysis, as appropriate for the illustrative analyses envisaged in this study. We acknowledge that this schema would need significant evolution, possibly even evolving to an object oriented or XML database, to remain XML-agnostic through more complex instantiations of the XML-DIUE framework:

- Tags: This database table is used to store data about different elements and their values in individual XLIFF files. The database table consists of fields like, tag, uri and value. The uri field is used to store information about the namespace for a particular element;
- Attributes: This database table is used to store information about attributes and their values;
- Children: This database table is used to store information about tags, their children and trailing content (if any) of tags.
- Validate: This database table consists of information about the individual files' well-formedness (i.e. XML parsing status): validation results after validating files against available schema, and reference to errors encountered during the parsing or validation process;

3.3. Analysis: suggested usage metrics

Designing the usage-analysis is at the core of the XML-DIUE framework. It is here where the designer identifies the usage characteristics of interest and thus, this analysis activity should be considered in parallel with the development of the repository schema and the parsers. It should be stated however that (as noted above) such an approach tends to limit the advantages of the Parser-Repository-Analysis architecture and so a better approach is to build a more generic repository structure that persists a superset of the information required for any specific set of evaluations.

In Table 2 below, we identify several research questions that could drive usage-analysis. Each research question has an associated measurement, which provides valuable insights (but not necessarily absolute answers) into the research question. These evaluations provide empirical evidence useful for different stakeholders of the standard. Although there can be a number of stakeholders involved in the standardization process [10], in this research, we mainly considered developers of the standard; users of the standard; tool vendors who implement the standard in their tools and other standard-related service providers such as those who provide consultancy or training and education activities around the standard.

Table 2: Proposed XML-DIUE usage metrics.

	Research Question	Measure taken	How it might inform the stakeholders
1	Are there recurrent syntactic conformance issues of the standard? If, so, what are they?	Identify validation prevalence and rank individual errors.	Helpful in identifying common validation errors for users and those responsible for documenting the standard.
2	How might the standard be simplified?	Identify least frequently used features.	Gives an indication of the features that might be deprecated by standard developers.
3	What are the core useful features of the standard?	Identify prevalently used features across organizations.	Allows standard developers to identify the features that have widest ripples effects upon change.
4	Do organizations deviate from the norm with respect to usage of the standard?	Identify relatively frequently used features by individual organizations.	This provides a basis for assessment and comparison of organizations' individualistic standard usage practices.
5	What candidate features could be usefully introduced to the standard?	Identify prevalent custom features and frequently added extensions.	This information could be used to provide guidance on new features that should be adopted by standard developers.
6	Are there common usage practices for the standard?	Identify frequent feature usage patterns.	These patterns may provide insight into elegant and consistent solutions to recurring problems in the usage of the standard promoting best-usage practices.
7	Do semantic ambiguities exist within the features of the standard?	Identifying different values used for the same features and the same values under different features.	These may imply semantic conflicts for users and standard developers.

Details of above individual evaluations for XLIFF study are discussed in the following sections (Section 3.3.1 - Section 3.3.7).

3.3.1. Validation Errors

Identification of frequent validation errors is especially helpful for the standard developers, tool developers and the standard service providers. Validation errors may indicate implementation errors in XLIFF by tools, errors associated with parsers within validation tools or errors associated with manual creation. This analysis may also reveal erroneous or unintended usage of the specification[6]. For example, this analysis may help to identify unexpected or unforeseen feature usage patterns that are valid according to the specification, but invalid according to a schema. Such scenarios might occur due to loosely defined criteria in the standard specification, that is more formally implemented in the schema.

The degree of syntactic conformance to a standard can be calculated by computing the ratio of the number of valid files (of a selected version of the standard) to the total number of files (of the same version) in the entire corpus as a percentage.

The XLIFF standard (version 1.2) has been specified in two flavours[7] [15]: 1) Strict schema: where deprecated elements and attributes are not allowed and 2) Transitional schema: where deprecated items are allowed to be used by applications that produce older versions of XLIFF. For example, the degree of syntactic conformance to the XLIFF version 1.2 Transitional Schema can be calculated using the following formula (see eq. 1):

$$\frac{Number\,of\,valid\,XLIFF\,version\,1.2\,(transitional)\,files}{Total\,number\,of\,XLIFF\,version\,1.2\,(transitional)\,files} \times 100 \qquad (1)$$

The higher the above ratio, the better the conformance to the standard. The result of this formula could be complemented by a list of violation types, ranked by prevalence, detailing the specific violations that cause problems. This ranked list provides a basis for improving the standard conformance; for example, by facilitating users in editing the files or, more generally, through the provision of focused test suites.

Two main limitations can be identified in this analysis, that the user should be aware of. The first is that the ratio metric does not take into account the relative complexity of individual files. For example, consider two

[6] Here, we use the term "specification" to refer to the documentation of a standard that describes the semantics, syntax, conformance criteria, processing requirements etc. in writing, whereas by "schemas" we refer to the implementation of rules and constraints defined in the standard by means of a schema language such as DTD, Relax-NG, Schematron so that files of the standard can be validated against the schema.

[7] http://docs.oasis-open.orgxliffv1.2osxliff-core.html#Intro_Flavors [accessed 16 April 2015]

files: a file with one thousand lines, and fifty unique elements which contain a single syntactic error. The other file just contains two lines of code, where it only uses a single unique element that also contains a single syntactic error. Under the above analysis, both files will be counted as invalid, contributing the same amount to the ratio, while the first one has much more content and is relatively complex compared to the second one.

The other limitation is that although a file might have several validation errors, most of the validation tools are only capable of detecting the first error: an issue highlighted in discussions at [30], [31], and [32]. These discussions suggest that the validation tools behave in this way because, after the first error, the rest of the document is potentially unpredictable and difficult to accurately validate further.

3.3.2. Least frequently used features

The least frequently used features may give an indication of the features of the standard that are not widely used, that users are unaware of or of features that are not implemented by the majority of tools. This information is useful for the standard developers to make decisions on removing features of the standard which are never used or used only by a minority of users. By removing these features, the standard can be simplified. For the purpose of categorizing whether a certain feature is a less frequently used feature or not, we propose to calculate relative usage of individual features using the following formula. Relative usage of an element or attribute X is given by:

$$\frac{Number\,of\,occurrences\,of\,X\,in\,the\,corpus}{Average\,occurrence\,of\,elements\,or\,attributes\,in\,the\,corpus} \times 100 \qquad (2)$$

The denominator is computed by counting all occurrences of attributes in the entire corpus and dividing by the number of attribute-types in that version of the standard. It serves merely to normalize the results over different versions of XLIFF for longitudinal studies, but makes no difference in studies of one version of XLIFF only.

It is worth noting that there can be highly important features with a low frequency of usage (for example, the version attribute of `<xliff>` is highly important, but it only appears once in a file). This problem of important features being categorized as less frequently used features, can be addressed to a certain degree by categorizing individual features into groups and then identifying the least frequently used features within each category. [33] identified the following groups:

- content features (features used to represent content);
- structural features (feature used to organize or maintain relationship or hierarchy between several related features);
- presentation features (features used to format or present content) ;
- meta-data features (features that carry various metadata).

Alternatively the metric identified in Section 3.3.3 could be used in combination with the metric presented here to identify the spread of the feature. Features with low commonality across companies and a low number of occurrences could then be considered candidates for deprecation/modularization away from the core.

3.3.3. Most prevalently used features across organizations

The most commonly used features are defined here as those features used by the biggest variety of users, in this case the 3 companies, the Eclipse project and the set crawled from the web (see Fig. 2 as an illustrative example).

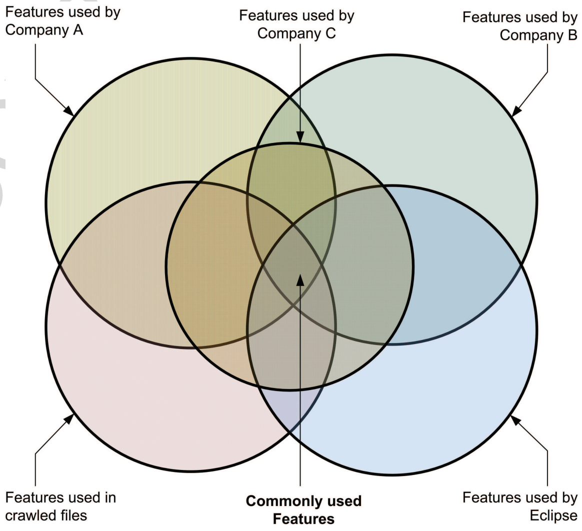
Features used by Company A
Features used by Company C
Features used by Company B
Features used in crawled files
Commonly used Features
Features used by Eclipse

Figure 2: The concept of commonly used features.

While this data-set is relatively small and we need to be cautious drawing significant conclusions from it, this analysis type is important because every slight modification to the features used across companies has the potential to have a great effect on a variety of users and tools. These are the core features that are important in terms of cross-organizational data interoperability. They are also important for providing backward compatibility for new versions of the standard. From the point of view of tool vendors, prioritized implementation of these features in their tools may help to maximize their coverage of the standard for minimum effort. Similarly, for the standard users, maximized use of these features instead of feasible alternatives could then ensure better cross-organizational interoperability. Therefore, this analysis indicates features that could be considered the most influential and the core useful features of the standard.

We propose the calculation of a commonality ratio for individual features, for the purpose of categorizing whether a feature is commonly used or not. The commonality ratio of feature X can be calculated using the following formula:

$$\frac{Number\,of\,organizations\,that\,use\,X}{Total\,number\,of\,organizations\,contributing\,to\,the\,corpus} \times 100 \qquad (3)$$

The higher the commonality ratio of a feature, the more useful is the feature for a majority of organizations. While features with a 100% commonality ratio can be considered as the most widely used features of the standard, an alternative is to determine a commonality threshold depending on the number of organizations who contributed to the corpus.

3.3.4. Relatively frequently used features of individual organizations

Identification of the most frequently used features of individual organizations gives a basis for comparing features used in different organizations. It measures the spread of a certain feature in specific organizations. It may be useful for those users in an organization who have to assess the effect of either removal or modification of a certain existing feature in a new version of the standard. In identifying where they deviate from the norm they may also be able to identify commonly accepted, and possibly more optimal, alternatives. Such analysis might improve their efficiency while, at the same time, contributing towards improving potential data interoperability with other organizations.

While the above metric "Least frequently used features" introduced in Section 3.3.2 considers the entire corpus for determining the least frequently used features of a standard, a modified version of the same formula can be used to determine the most frequently used features (and the least frequently used features) of individual organizations. Here the equation is being applied to features derived from XML files obtained from individual organizations. The modified equation for determining the relative frequency of usage with an organization is given below. The relative usage of element or attribute $\boldsymbol{X}$ in the organization $\boldsymbol{Y}$ is given by:

$$\frac{Number\,of\,occurrences\,of\,X\,in\,the\,files\,obtained\,from\,Y}{Average\,occurrence\,of\;X\,in\,the\,corpus} \times 100 \qquad (4)$$

During the calculation of this metric, features used by individual organizations can be sorted in the descending order of their relative usage frequencies and listed in a table. This table can be used to manually compare the difference of individual feature-usage across different organizations.

3.3.5. Prevalent custom features and frequently added extensions

Commonly used custom features (i.e. attributes and attribute values, elements and element-values) that are not defined in the schema can be manually examined to identify potential features that could be standardized in future. Custom feature identification can be automated, for example by programmatically filtering all features and feature values that are not defined in the schema or specification. Then the subset of commonly used custom extensions may give subtle hints as to where new features can be adopted for the standard. Various external schemas used and external features implemented in the files can be analyzed under this metric. However, this involves manual investigation, towards semantically relating the identified extensions, and consequently could be an effort intensive activity.

3.3.6. Feature usage patterns

Frequently used feature configuration patterns (XML snippets) are useful in identifying possible best usage practices of the standard. Manual or automated [34] pattern identification mechanisms can be employed for this purpose. This is especially useful for the standard developers, helping them to identify syntactically and possibly even semantically related feature configurations. Identification of best usage practices will be helpful to promote such usage. Tool vendors may also find this information useful, to optimize their tools, by providing templates for commonly occurring patterns. This will also serve to improve the tools' interoperability.

3.3.7. Feature values

For each attribute and element, manual analysis of attribute values and elements values is especially useful for the standard developers to identify semantic issues associated with them [35]. This analysis may provide strong evidence of [36]:

- attributes or elements that are used to represent the same type of information (i.e. ambiguous features that hold the same value);
- the use of different values or unit dimensions in attributes or elements to represent the same information (i.e. ambiguous feature values).

For example, features where values have been represented as different expressions (city = Dublin, Dub) and to different degrees of precisions (e.g. grade = A+, very good) can lead to semantic interoperability issues.

Analysis of this kind may provide some useful statistics about the usage of pre-defined feature values (defined in the specification). However, manual analysis of values may not be practical for attributes or elements where large or infinite numbers of possibilities exist for their values. Therefore, we recommend the application of this evaluation for specific and pre-determined sets of features only.

4. An XLIFF study based on XML-DIUE

In this section we expand upon the case study we are using to illustrate the XML-DIUE framework. This study focuses on XLIFF, a data-exchange standard from the localization domain. Section 3 has referred to how the data was captured for this case-study. In this section we discuss the specific research questions for the case study, in the context of the representative research questions suggested for the XML-DIUE framework, and present the associated results and discussion.

Table 3 is a re-print of Table 2 where the last column has been replaced with the research questions that this XLIFF case addresses. Due to space considerations and the modelling limitations of our prototype repository, only research questions one, two, three and seven from the XML-DIUE framework are targeted in this illustrative XLIFF study.

Table 3: Research questions of the XLIFF case study.

	XML-DIUE Research Question	Measure taken	Research question of the XLIFF case study
1	Are there recurrent syntactic conformance issues of the standard? If, so, what are they?	Identify validation prevalence and rank individual errors.	How conformant are the files contained in the corpus to the XLIFF standard and what are the prevalent non-conformance issues?
2	How might the standard be simplified?	Identify least frequently used features.	What elements might be removed from the XLIFF specification? What attributes might be removed from the XLIFF specification?
3	What are the core useful features of the standard?	Identify prevalently used features across organizations.	What are the features of the XLIFF standard with the widest prevalence across the companies in the data-set?
7	Do semantic ambiguities exist within the features of the standard?	Identifying different values used for the same features and the same values under different features.	What attributes and attribute values are suggestive of semantic interoperability issues[8]?

For each of the above illustrative research questions, different standard usage analyses were identified and computed over the database using SQL. For example, the SQL query for identifying the most/least frequently used elements by 'Company A' was:

```
select tag, uri, count(tag)  as frequency from tags
    where source = "Company A"
    group by tag, uri
    order by frequency desc
```

It is pertinent to mention that analysis of a single research question may involve several standard usage analyses and several SQL queries too. For example, the third research question involves running the above query for

[8] The XLIFF elements are not analyzed under this study, mainly due to the fact that most of the XLIFF elements are have an infinite amount of possibilities for the content, which makes manual analysis impossible.

each company and calculating the average occurrence of the elements across companies.

4.1. Results

Although the current XLIFF corpus might not represent the true distribution of XLIFF content in production environments, our literature review suggests that this is the largest corpus of XLIFF files currently available to researchers and we believe that the early results presented here provide interesting insights in guiding future work on interoperability of localization data, in addition to demonstrating the utility of the XML-DIUE framework. The results of the analyses performed on the XLIFF corpus are summarized below.

4.1.1. Validation results

The XLIFF files have been manually validated against the schemas of version 1.2, 1.1 and 1.0 of XLIFF. For this purpose, XLIFFChecker (version 1.0-2)[9] a freely available tool specifically designed for validating XLIFF files was used. However, XLIFFChecker only reports on the first validation error it finds.

The results show that 1.2 is the predominantly used XLIFF version. Indeed, given the amount of XLIFF version 1.2, this suggests a strong backward-compatibility requirement for the new version (version 2.0). Unfortunately, according to the specification, XLIFF 2.0 is not fully backward compatible with XLIFF 1.2 [18].

Out of the 2,758 XLIFF version 1.2 files, 2,362 files are found to be invalid. Out of the valid XLIFF 1.2 files, 22 files were transitional schema valid and 374 files were strict schema valid. The results for all versions are summarized in Fig. 3. According to the formula described in Section 3.3.1, the overall degree of syntactic conformance to the XLIFF 1.2 schema is computed at an alarming 14.36 per cent. This means, the vast majority of the generated XLIFF 1.2 compatible files (85.64%) are invalid. The lack of conformance to the standard, as per XLIFFChecker, is clearly evident from these results. This lack of conformance would seem to impact strongly on the potential for interoperability.

[9] The XLIFFChecker is available to download from: `http://www.maxprograms.com/products/xliffchecker.html` [accessed 22 March 2013]

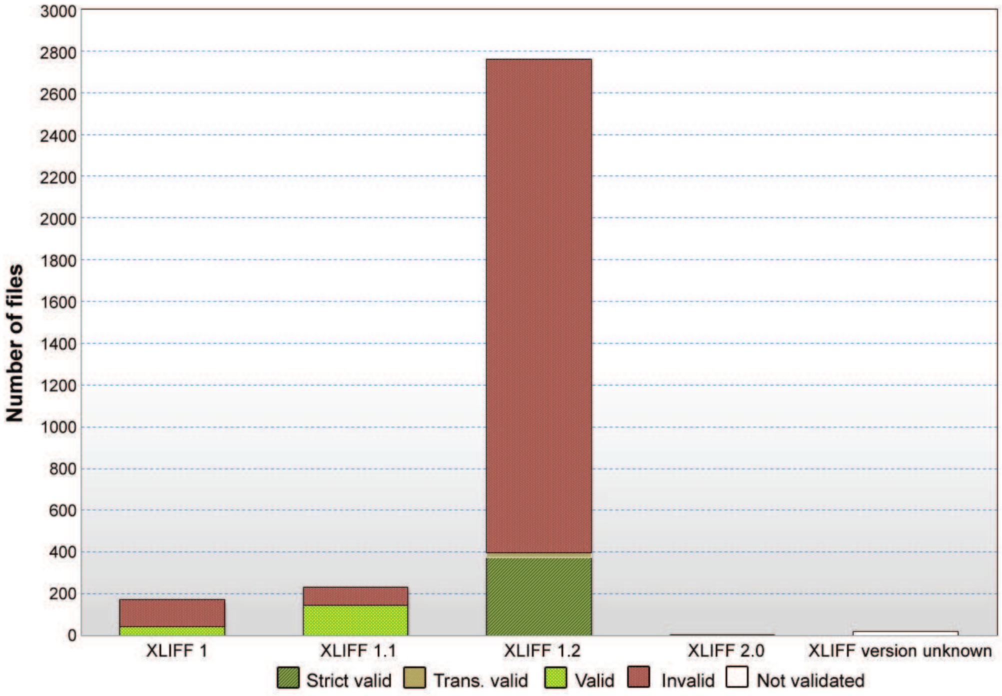

Figure 3: Validation results[10].

These XLIFF 1.2 files were further analyzed for their complexity in terms of their `<trans-unit>` count and their validation results to see if conformity was associated with size. Fig. 4 shows the number of invalid, valid and strict valid files vs. the log (`<trans-unit>` count).

Ranked list of validation errors. The top 10 most-frequent validation errors can be found at `www.localisation.ie/sites/default/files/AppendixB-RankedErrorList.pdf`. Here we restrict ourselves to the validation errors that were detected more than 30 times in our analysis.

Investigation of the XML Schema Definition (XSD) for XLIFF 1.2 indicates that the error with the highest frequency of occurrence is that the tools mentioned in the `tool-id` attribute of the `<alt-trans>` units are not defined under the `<tool>` element, which should reside within the `<header>` section

[10] The XLIFF files were validated against the version of the XLIFF they stated themselves.

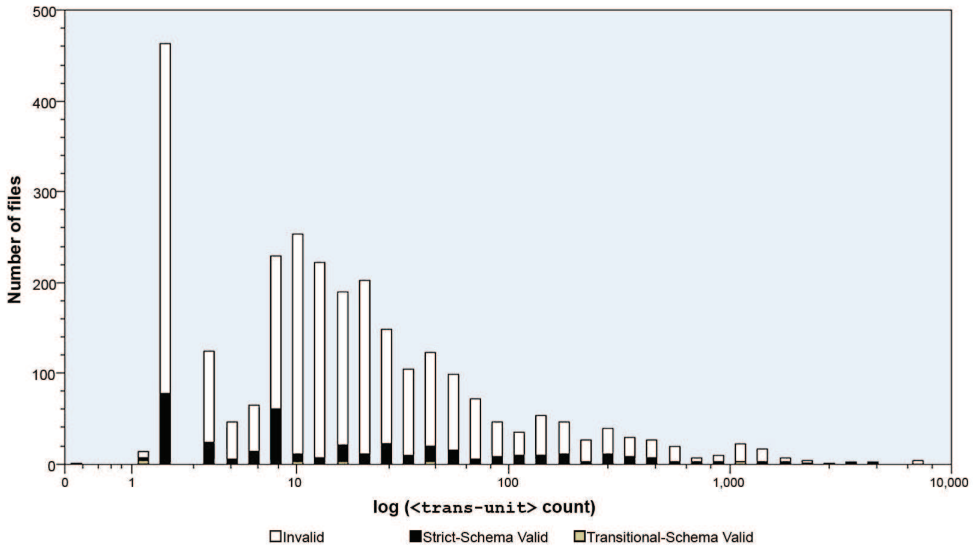

Figure 4: The graph of validation result vs log (`<trans-unit>` count).

of the `<file>` element. This error predominantly occurred in Eclipse XLIFF files but this is meta-data and does not significantly affect the processing of XLIFF content.

The second most common error was due to invalid attributes found in different elements. These attributes have been used without defining their respective namespaces. Depending on the attribute this can have a more significant effect on the processing of XLIFF content. The third most common error was a large degree less prevalent than the first two. It was the use of duplicated identity values (IDs) within files, where the IDs of elements should be unique within files. Surprisingly in fourth place we have observed just a message called "null" provided by the validation tool. Although this is very likely to be due to an issue associated with the validation tool itself, further manual analysis of these files and the use of different validation tools is needed to confirm this.

Most of the other errors are due to violation of rules specifying different patterns and pre-defined values or formats in the XLIFF specification (e.g. using invalid formats for representing languages and dates). Many of these were invalid formats for languages and, for obvious reasons, such errors would have significant effects on tools' abilities to process the XLIFF content.

4.1.2. Suggested simplification of the standard

Least frequently used XLIFF elements and attributes have been identified, using the metric presented in Section 3.3.2. Due to the significant variation of XLIFF content in the individual files, for each element, the relative usage frequencies was computed with respect to the total `<trans-unit>` element count.

Least frequently used elements. The total number of occurrences of XLIFF elements in the corpus is 2,275,559. The number of XLIFF elements defined in the XLIFF specification is 38. Hence the average use of an element in the corpus is calculated to be 59883.13. Then, using the equation (see eq. 2) described in Section 3.3.2, the relative usage of individual elements was calculated for each element. In addition, a slightly modified version of the same formula was used to calculate the relative usage of individual elements with respect to total `<trans-unit>` count (i.e. element count / total `<tans-unit>` count). Normalization of the feature counts with respect to `<tans-unit>` count is more specific to XLIFF but is more suited to our research here because it more accurately represents the complexity of each XLIFF instance than normalizing for the average occurence of features in the corpus, which is more suited to longitudinal comparisons. The elements with a relative usage of less than 1% for either measure are considered the least frequently used elements in this data-set. These are listed in Table A.1 of Appendix A with their relative usage frequencies calculated with respect to the average element usage in the corpus, and the total `<trans-unit>` element count.

Least frequently used attributes. The total number of occurrences of XLIFF attributes in the overall corpus is 1,519,314 and the average number of attribute usage is calculated at 18,528.22. Using these figures, the relative usage of individual attributes was computed using the formula (see eq. 2) described in Section 3.3.2. They were also computed with respect to the total `<trans-unit>` count. These are listed in Table B.1 of Appendix B.

Findings with respect to least frequently used features. The "importance" of a feature depends on both the commonality ratio and the relative usage frequency of that feature. Therefore, to identify the candidate features for deprecation (or removal), both the frequency of usage and the commonality ratio of the feature has to be taken into account. Features with very low relative frequency of usage and a low commonality ratio are likely to be good candidates for consideration for deprecation from the standard. In our research we considered the features of low frequency to be those normalized with respect to the total <trans-unit> count.

We analyzed the commonality ratios of the identified least frequently used features. Then we chose features with a commonality ratio of 40% (i.e. used by at most two organizations) that are also in the "least frequently used" set, as potential candidates for deprecation.

Analysis of commonality ratios of the least frequently used elements revealed that <xliff> has a commonality ratio of 100% while the <external-file> element has a 60% commonality ratio. The rest of the elements have a 40% commonality ratio or less. Thus, this technique suggests the following as candidates for deprecation: <reference>, <bin-target>, <ex>, <bx>, <bin-source>, <bin-unit>, <internal-file>, <skl>, <prop-group>, <it>, <prop>, <sub>, <seg-source>, <g>, <glossary>, <phase-group>, <phase>, <count-group>, <count>, <mrk>, <bpt>, tool>.

The attributes alttranstype, annotates, assoc, clone, comment, extype are never used in the corpus. Therefore, these attributes would seem like the ideal candidates for deprecation. The analysis of the commonality ratios of the "least frequently used" attributes revealed that product-name, product-version, build-num, date, href and tool are the only attributes from the list with a 60% commonality ratio. The rest of the attributes have a commonality ratio of 40% or less. Therefore the technique suggests the following list of candidate attributes for possible deprecation: alttranstype, annotates, assoc, category, charclass, clone, comment, company-name, contact-email, contact-name, contact-phone, count-type, crc, css-style, ctype, equiv-text, equiv-trans, exstyle, extype, font, form, help-id, job-id, maxbytes, maxheight, maxwidth, menu, menu-name, menu-option, merged-trans, mid, mime-type, minbytes, minheight, minwidth, mtype, phase-name, pos, priority, process-name, prop-type, reformat, rid, size-unit, state-qualifier, style, tool-company, tool-name, tool-version, uid, unit, xid.

While our framework helps to identify possible features that can be deprecated to reduce the complexity, the ultimate decision on deprecation of a certain feature has to be taken by the developers of the standard (e.g. the XLIFF TC), with the consensus of all members. Moreover, the standard developers may also decide to isolate such features from the core useful features and construct separate modules instead of deprecating them.

Analysis of the standards themselves has shown that <prop-group> and <prop> elements have been already deprecated in XLIFF version 1.1. Moreover, lack of use of <bin-unit>, <bin-source> and <bin-target> elements has already been identified by the XLIFF TC, and the creation of an optional Resource Data Module[11] that caters for the functionality of those elements has been incorporated in the new version of the standard (i.e. XLIFF 2.0). The lack of use of the <it> inline element as well as issues associated with other inline elements such as <ex>, <bx> and <g> were discussed[12] within XLIFF TC meetings, and are addressed by the inline element sub-committee of XLIFF in XLIFF 2.0.

This alignment with the XLIFF TC's deliberations suggests that the technique has utility in identifying features for deprecation or modularization. It also suggests that the other features identified by the approach, and not in the TC's thoughts, should be considered by the TC for deprecation/modularization.

4.1.3. Prevalently used features of the standard

We utilized the metric described in Section 3.3.3 in order to identify prevalently used features of the XLIFF standard across the companies in our data-set. First, we calculated the commonality ratio of XLIFF elements and identified the prevalently used elements by considering elements with over 60% commonality ratio (i.e. features at least used by three of the five organizations/sources). Then, attributes of more than a 60% commonality ratio were identified for each element. The core features identified by this analysis are given below in Table 4.

In this table, the elements <xliff>, <file>, <header>, <body>, <trans-unit>, <source>, <target> all had 100% commonality ratio, meaning they have been used in all five sources. This is to be expected for the mandatory structural elements like <xliff>, <file>, <body> and <source>. While the features <header>, <trans-unit> and <target> are not specified as

[11] See XLIFF 2.0 Resource Data Module Specification http://docs.oasis-open.org/xliff/xliff-core/v2.0/os/xliff-core-v2.0-os.html#resourceData_module [accessed 18 September 2014]

[12] For example, see XLIFF TC discussion: http://markmail.org/search/?q=ex+bx++inline+xliff#query:ex\%20bx\%20\%20inline\%20xliff+page:1+mid:jnw2tjfdukhwhbtk+state:results [accessed 17 January 2013]

mandatory in the standard, it would be surprising to find an XLIFF file without these features.

Table 4: Core features of the XLIFF standard.

Element	Attribute
xliff	version
file	original, source-language, target-language, product-name, product-version, build-num, tool, datatype
header	-
body	-
trans-unit	id, approved, translate, resname, restype
source	xml:space
target	xml:lang, state
alt-trans	
external-file	href
note	from
ph	id
group	resname, restype

The analysis of commonly used elements across different sources (see Section 3.3.3) leads to the identification of important features in terms of cross-organizational data-exchange interoperability. The XLIFF specification [15] as well as Frimannsson and Hogan [37] illustrate a minimal XLIFF document. The elements and attributes presented in the minimal XLIFF document are given in Table 5.

Table 5: Elements and attributes found in the minimal XLIFF document (adopted from [15]).

Element	Attributes
xliff	version
file	original, source-language, datatype
body	-
trans-unit	id
source	-

While all of the core useful features identified using our framework (see Table 4) are presented in that minimal XLIFF document, our research has revealed several more additional attributes and elements that are in common use. The additional elements we discovered are `<header>`, `<target>`, `<alt-trans>`, `<external-file>`, `<note>`, `<ph>` and `<group>`, two of which were in all organizations' files. It is worth noting that our results are highly consistent with the features of the XLIFF Core in XLIFF 2.0[13]. In Table 6, we compare XLIFF 2.0 core elements with the core elements of XLIFF 1.2 as identified in this study.

Table 6: Comparison of XLIFF 2.0 core elements with the core XLIFF 1.2 elements found in this research.

XLIFF 1.2 Core Element (as identified in this study)	XLIFF 2.0 Core Element
xliff	xliff
file	file
trans-unit	unit
source	source
target	target
note	note
ph	ph
group	group
header	-
body	-
alt-trans	-
external-file	-
-	notes
-	skeleton
-	segment
-	ignorable
-	originalData
-	data

From Table 6, it can be seen that most of the elements that we system-

[13] See the specification of XLIFF 2.0: `http://docs.oasis-open.org/xliff/xliff-core/v2.0/os/xliff-core-v2.0-os.html`[accessed 18 September 2014]

atically identified as important elements in this research are retained as core elements of XLIFF 2.0. In addition to those elements, a few new elements have been introduced in XLIFF 2.0, for example the `<notes>` element, which can be used to group individual `<note>` elements.

4.1.4. Attributes and attribute values that can cause interoperability issues

In XLIFF, attributes play a major role by carrying important metadata in the localization process. We manually analyzed XLIFF attributes and their values to identify attributes and attribute values that can cause both semantic and syntactic interoperability issues.

We obtained all the unique values used for individual attributes from the database. The values were mainly analyzed for the following criteria: formats used to represent values, typical values found, default value usage and the predefined value usage. Several attributes and their values found in the corpus are given in Table 7, but the full list can be accessed from: `http://www.localisation.ie/sites/default/files/AppendixA-Attributes.pdf`.

Table 7: Attributes and their values.

Attribute	Typical values/formats found in the XLIFF corpus
category	String, Enterprise Software, resourceBundle, Survey Project
date	04/02/2009 23:24:18, 11/06/2008, 2001-04-01T05:30:02, 2006-11-24, 2007-01-01, 2010-03-16T21:58:27Z
match-quality	100, 100%, 78.46, fuzzy, String, Guaranteed
source-language	EN, EN-US, ENGLISH, cs, da, de, de-DE, en, en-DE, en-US, en-us, en_US, es, fi, fr, fr-FR, it, ja, ko, pt, ru, sv, sv_SE, unknown, x-dev, zh_CN, zh_TW

The manual analysis of the attributes and attribute values revealed that use of certain pre-defined attribute values was non-existent (e.g. the pre-defined value `lisp` for `datatype` attribute is never used in our XLIFF corpus). Pre-defined attribute values that no-one uses increase the complexity of the standard and therefore standard developers could consider deprecating these pre-defined attribute values to reduce complexity.

Moreover, this analysis revealed several potential interoperability issues associated with attributes. For example:

- use of custom values regardless of the existence of predefined values (e.g. the use of the `pofile` value instead of the predefined `po` value for the `datatype` attribute);

- use of different expressions to denote the same value (e.g. use of `xml` and `x-text/xml` values for `datatype` attribute), and the use of different units to represent values (e.g. use of values such as 100, 100%, 78.46, and `fuzzy` for the match-quality attribute);

- the use of the same values in different attributes (i.e. ambiguous attributes). For example, the use of the user-defined value `x-tab` in both the `context-type` attribute and the `ctype` attribute; the use of the predefined value `database` in both `datatype` and `context-type` attributes; the use of the user defined value `x-button` in the `context-type` attribute and button in `restype` attribute;

Issues like the ones outlined above can be addressed by:

- standardizing most frequently used custom values;

- agreeing on the semantics of values as well as attributes;

- tightening the standard by strict conformance clauses to cover those semantics.

- publishing of conformance test suites. This may help tool developers to identify and address such issues.

We also noticed the use of extreme values in some features. For example, the use of extremely lengthy strings for IDs occurred several times. The lengthiest ID (i.e. value for the `ID` attribute) found within our corpus was 8,583 characters. Although the specification recommends avoiding spaces within IDs, IDs have been found not only to use spaces but also to use several line breaks. These extreme values, though allowed in the specification, may affect interoperability. Standard developers should take measures to tighten the conformance clauses to avoid these kinds of behaviors.

As shown in Table 7, our analysis also revealed attributes that may cause syntactic interoperability issues through mainly the use of incorrect formats to denote values in some attributes. Examples include representing language in formats other than formats specified in BCP 47[14] (values such as `en_US, ENGLISH, x-dev, unknown, uz-UZ-Cyrl`) or representing dates not as specified in ISO 8601[15] (values such as `04/02/2009 23:24:18, 11/06/2008`). These kinds of issues may also be addressed by developing and publishing conformance test suites.

5. Limitations of the current framework

This paper has illustrated the potential of the XML-DIUE framework to help discover 'bloat' in standards' implementations and standards specification. It has also shown its utility in identifying important features of standards that are core to successful interoperability.

However, there are several limitations of the current framework and the associated illustration using XLIFF. The external validity of our XLIFF study may be considered low, mainly due to the lack of representativeness of our XLIFF corpus. The corpus must be "representative" in order to deduce broad generalizations concerning the XLIFF standard. Indeed, building a representative corpus for XML-DIUE-based studies in general may not be trivial. For example, the XLIFF files targetted here could carry confidential information and therefore companies may not be prepared to share them, as was the case in our case study. Additionally, factors such as sampling with respect to proprietary and non-proprietary systems, different companies, size and complexity of files and several other factors must all be considered when building the corpus, for drawing accurate conclusions. It should be noted however that the study reported on here has built the largest corpus of XLIFF files available to researchers.

Only a few usage metrics have been selected and analyzed under this study. More usage analysis metrics should be introduced (for example, an analysis of the usage of mandatory attributes, or an analysis of the tools involved in the production of files, with respect to the files they create).

Another limitation associated with usage metrics is that they may not provide precise results, regarding the information required for a certain issue. They will only generate results that provide indications or implications towards improving interoperability related aspects. It is likely that most

[14] `http://tools.ietf.org/html/bcp47` [accessed 13 March 2013]

[15] `http://www.iso.org/iso/catalogue_detail?csnumber=40874` [accessed 13 March 2013]

analysis tasks will require human intervention to make intelligent decisions towards improving a standard. For example, even though the frequency of usage of an element, as calculated by the XML-DIUE framework, suggests that it should be deprecated, the TC should ultimately decide its fate.

The current framework mainly focuses on identifying deficits in a file format standard through a bottom-up approach. A complimentary top-down analysis, while not covered by our framework, might be helpful in identifying future requirements for standards. An example is the framework proposed by Mykkänen and Tuomainen [38]. However, neither the framework proposed by Mykkänen and Tuomainen [38] nor our own framework currently consider aspects such as the cost of implementation of standards.

6. Conclusion and future work

This paper describes a novel empirical framework for the usage analysis of a corpus of standard XML documents. The paper illustrates the utility of the framework by focusing on usage of the XLIFF standard, and its effect on interoperability.

The research has shown the potential utility of XML-DIUE. It has illustrated how the framework can help discover some of the limitations of XML-based data-exchange standards as used, identifying possible improvements with regard to data-exchange interoperability. This suggests that, the XML-DIUE empirical framework can be used as a tool to evaluate data-exchange standard usage and thus inform on the development and maintenance of such standards.

Specifically the illustrative study has revealed some important findings related to XLIFF, and has produced important recommendations aimed at improving the interoperability of XLIFF-based data. As discussed in Section 5, the nature of our XLIFF corpus makes it difficult for us to generalize the results fully. However, our XLIFF corpus was significant enough to illustrate the features that are core to the standard, and the features that are possible candidates for deprecation and modularization. The study also showed the low level of conformance achieved by XLIFF files in real-life and highlighted several syntactic and semantic issues related to attribute values that should be addressed.

More importantly, prior to the development of our framework and corpus, the XLIFF Technical Committee (TC), and other committees like it, had no systematic methodology to evaluate and analyze the usage of the standard, thus limiting the amount of user feedback into the standard. Our research, for the first time, gathered sufficient (corpus) data and developed the corresponding framework to allow for a systematic analysis of the usage of the standard. Using the current framework, the XLIFF TC can now analyse the actual use of the specification and focus on the aspects of the standard that are more or less important to the actual stakeholders of that standard. It is also worth mentioning that XLIFF TC's Promotion and Liaison Sub Committee (P&L SC) has expressed their interest[16] in implementing the XML-DIUE framework and it has already started to collect XLIFF files, with the intention of constructing a representative corpus for XML-DIUE analysis.

Significant resources, in addition to the corpus and the empirical framework, emerged from this research, including the XLIFF database derived from the corpus, the technologies to crawl, pre-process, validate, and construct the database, and data analysis criteria. These resources can be used in the future to evaluate and improve both the XML-DIUE framework and the XLIFF standard further, while also guiding the evaluation of similar standards.

Future work in this area involves trialling of the XML-DIUE research questions that were not embodied in the illustrative case study, the identification of other relevant usage metrics and associated research questions, widening the domains and data-sets on which XML-DIUE is applied, repository evolution and the development of tools that aid XML developers comprehension and navigation of XML documents (an illustrative example of such a tool from the software development domain would be [39]).

Three of the research questions proposed for XML-DIUE type analysis have not been expanded upon in the study described here. Research question four, which assesses the degree to which companies deviate from the norm in their usage of a standard, is in line with the others presented here, being quite quantitative in nature. But research questions five and six probe more qualitative aspects of standards and this implies more in-depth analysis which may prove difficult or prohibitively expensive on the large data sets envisaged. However it is likely that such analysis will be rewarding at a deeper level,

[16] See initial discussion about this topic in the meeting minutes of the meeting of the XLIFF P&L SC: `https://lists.oasis-open.org/archives/xliff-promotion/201211/msg00002.html` [accessed 13 February 2013]

identifying best-practice for expansion of the standard in line with users needs. In addition, the identification of a larger set of usage-metrics that are congruent with the needs of builders/users of the standards is also a important. These metrics must have explicit traceability from those needs.

Larger data-set and data-sets from different domains should be sought. For example, in the related work section the issue of interoperability of word-processing documents was discussed. Intuition would suggest that the vast amount of word-processing software users only employ a small fraction of the processors' capabilities. This intuition suggests that the XML-DIUE framework could be successfully leveraged to streamline the data-interoperability standards in this area while maintaining high interoperability. We anticipate that this would be the case in many domains.

Finally, the repository is currently a relational database that is sufficient for analyses of the type performed in the XLIFF case study only. But it is insufficient for deeper analyses that rely on the detailed structural aspects of XML files. This modelling will not be trivial if the repository is to remain XML-agnostic. We see this modelling effort as a priority in our evolution of the XML-DIUE framework.

Acknowledgements

This research is supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation `www.cngl.ie` at the Localisation Research Centre (Department of Computer Science and Information Systems), University of Limerick, Limerick, Ireland. It was also supported, in part, by Science Foundation Ireland grant 10/CE/I1855 to Lero - the Irish Software Engineering Research Centre `www.lero.ie`. Thanks to Dr. Ian O'Keeffe and Dr. David Filip for all the guidance, suggestions and feedback. We would like to thank the CNGL industrial partner organizations that contributed to the XLIFF corpus. We would also like to thank our colleagues and fellow researchers from the Localisation Research Centre for their valuable input to this project and the paper.

Appendix A. Least frequently used elements

Table A.1: Least frequently used elements.

Element	Corpus average (%)	`<trans-unit>` count (%)
reference	0.00	0.00
bin-target	0.01	0.00
ex	0.07	0.01
bx	0.07	0.01
bin-source	0.09	0.02
bin-unit	0.09	0.02
interna-file	0.11	0.02
skl	0.16	0.03
external-file	0.18	0.03
prop-group	.25	0.04
it	0.28	0.05
prop	0.47	0.08
sub	0.48	0.08
seg-source	0.7	0.12
g	.93	0.16
glossary	>1%	0.3
phase-group	>1%	0.3
phase	>1%	0.31
count-group	>1%	0.34
count	>1%	0.34
mrk	>1%	0.44
bpt	>1%	0.75
tool	>1%	0.9
xliff	>1%	0.9

Appendix B. Least frequently used attributes

Table B.1: Least frequently used attributes.

Attribute	Corpus average (%)	<trans-unit> count (%)
alttranstype	0.00	0.00
annotates	0.00	0.00
assoc	0.00	0.00
clone	0.00	0.00
comment	0.00	0.00
extype	0.00	0.00
merged-trans	0.01	0.00
uid	0.01	0.00
equiv-trans	0.02	0.00
tool-company	0.03	0.00
tool-version	0.05	0.00
contact-phone	0.07	0.00
category	0.08	0.00
equiv-text	0.08	0.00
job-id	0.08	0.00
charclass	0.17	0.01
minwidth	0.17	0.01
maxheight	0.19	0.01
form	0.21	0.01
mime-type	0.32	0.02
help-id	0.37	0.02
menu	0.37	0.02
menu-name	0.37	0.02
menu-option	0.37	0.02
product-name	0.38	0.02
state-qualifier	0.43	0.02
product-version	0.52	0.03
css-style	0.80	0.04
unit	0.80	0.04
maxbytes	0.82	0.04
minbytes	0.82	0.04

Continued on next page

Table B.1: The least frequently used attributes – continued from previous page

Attribute	Corpus average (%)	<trans-unit> count (%)
minheight	0.82	0.04
exstyle	0.86	0.05
reformat	0.90	0.05
priority	>1%	0.06
pos	>1%	0.07
prop-type	>1%	0.08
rid	>1%	0.08
size-unit	>1%	0.09
xid	>1%	0.09
maxwidth	>1%	0.11
crc	>1%	0.15
font	>1%	0.24
mid	>1%	0.24
mtype	>1%	0.24
build-num	>1%	0.3
company-name	>1%	0.3
contact-email	>1%	0.3
contact-name	>1%	0.31
process-name	>1%	0.31
href	>1%	0.33
tool	>1%	0.33
count-type	>1%	0.34
style	>1%	0.41
ctype	>1%	0.44
phase-name	>1%	0.53
date	>1%	0.59
tool-name	>1%	0.9

References

[1] A. Geraci, F. Katki, L. McMonegal, B. Meyer, J. Lane, P. Wilson, J. Radatz, M. Yee, H. Porteous, F. Springsteel, IEEE standard computer dictionary: Compilation of IEEE standard computer glossaries, IEEE Press, 1991.

[2] L. Weihua, L. Shixian, Improve the semantic interoperability of

information, in: Intelligent Systems, 2004. Proceedings. 2004 2nd International IEEE Conference, Vol. 2, 2004, pp. 591–594 Vol.2. doi:10.1109/IS.2004.1344818.

[3] Y. Savourel, XML Internationalization and Localization, Sams White Book Series, Sams, 2001.
URL http://books.google.ie/books?id=U75QAAAAMAAJ

[4] S. Ray, Healthcare interoperability - lessons learned from the manufacturing standards sector, in: Automation Science and Engineering, 2009. CASE 2009. IEEE International Conference on, 2009, pp. 88–89. doi:10.1109/COASE.2009.5234092.

[5] D. Filip, Focus: Standards and interoperability-the localization standards ecosystem, Multilingual 23 (3) (2012) 29.

[6] L. Aouad, I. OKeeffe, J. Collins, A. Wasala, N. Nishio, A. Morera, L. Morado, L. Ryan, R. Gupta, R. Schäler, A view of future technologies and challenges for the automation of localisation processes: Visions and scenarios, in: G. Lee, D. Howard, D. Sĺzak (Eds.), Convergence and Hybrid Information Technology, Vol. 206 of Communications in Computer and Information Science, Springer Berlin Heidelberg, 2011, pp. 371–382. doi:10.1007/978-3-642-24106-2_48.
URL http://dx.doi.org/10.1007/978-3-642-24106-2_48

[7] A. Wasala, O. Ian, R. Schäler, Towards an open source localisation orchestration framework, Tradumàtica: traducció i tecnologies de la informació i la comunicació (9) (2011) 84–100.

[8] TAUS, Lack of interoperability costs the translation industry a fortune, [Online; accessed 25-February-2011] (February 2011).
URL http://www.translationautomation.com/downloads/finish/56-public-reports/328-taus-lack-of-interoperability-costs-the-translation-industry-a-fortune-february-2011

[9] G. Lewis, E. Morris, S. Simanta, L. Wrage, Why standards are not enough to guarantee end-to-end interoperability, in: Composition-Based Software Systems, 2008. ICCBSS 2008. Seventh International Conference on, 2008, pp. 164–173. doi:10.1109/ICCBSS.2008.25.

[10] E. Soderstrom, Casting the standards play - which are the roles?, in: Standardization and Innovation in Information Technology, 2003. The 3rd Conference on, 2003, pp. 253–260. doi:10.1109/SIIT.2003.1251212.

[11] S. L. Lelieveldt, Standardizing retail payment instruments, Information technology standards and standardization: a global perspective. K. Jacobs (ed.), Hershey (2000) 186–197.

[12] R. Shah, J. Kesan, Interoperability challenges for open standards: Odf and ooxml as examples, in: Proceedings of the 10th Annual International Conference on Digital Government Research: Social Networks: Making Connections Between Citizens, Data and Government, dg.o '09, Digital Government Society of North America, 2009, pp. 56–62.
URL http://dl.acm.org/citation.cfm?id=1556176.1556191

[13] R. Shah, J. Kesan, Evaluating the interoperability of document formats: Odf and ooxml as examples, in: Proceedings of the 2Nd International Conference on Theory and Practice of Electronic Governance, ICEGOV '08, ACM, New York, NY, USA, 2008, pp. 219–225. doi:10.1145/1509096.1509141.
URL http://doi.acm.org/10.1145/1509096.1509141

[14] Google, Web authoring statistics, [Online; accessed 15-August-2012] (2005).
URL https://developers.google.com/webmasters/state-of-the-web/2005/

[15] XLIFF-TC, Xliff 1.2 specification, [Online; accessed 08-June-2009] (February 2008).
URL http://docs.oasis-open.org/xliff/xliff-core/xliff-core.html

[16] R. Raya, Xml in localisation: A practical analysis, [Online; accessed 10-June-2009] (August 2004).
URL http://www.ibm.com/developerworks/xml/library/x-localis/

[17] M. Bly, Xliff: Theory and reality: Lessons learned by medtronic in 4 years of everyday xliff use, in: Proceedings of the 1st XLIFF International Symposium, 2010.

URL http://www.localisation.ie/oldwebsite/xliff/resources/presentations/xliff_symposium_micahbly_20100922_clean.pdf

[18] XLIFF-TC, Xliff version 2.0, [Online; accessed 07-September-2014] (August 2008).
URL http://docs.oasis-open.org/xliff/xliff-core/v2.0/os/xliff-core-v2.0-os.html

[19] L. Morado-Vázquez, D. Filip, Xliff support in cat tools, [Online; accessed 05-March-2012] (January 2012).
URL http://www.localisation.ie/sites/default/files/XLIFFSotAReport_20120210_0.pdf

[20] L. Morado-Vázquez, F. Wolff, Bringing industry standards to open source localisers: a case study of virtaal, Tradumàtica: traducció i tecnologies de la informació i la comunicació (9) (2011) 74–83.

[21] A. Wasala, D. Schmidtke, R. Schäler, Xliff and lcx: A comparison, Localisation Focus - The International Journal of Localisation 11.

[22] T. Imhof, Xliff - a bilingual interchange format, in: MemoQ Fest, 2010.

[23] D. Anastasiou, L. Morado-Vázquez, Localisation standards and metadata, in: S. Sánchez-Alonso, I. Athanasiadis (Eds.), Metadata and Semantic Research, Vol. 108 of Communications in Computer and Information Science, Springer Berlin Heidelberg, 2010, pp. 255–274. doi:10.1007/978-3-642-16552-8_24.
URL http://dx.doi.org/10.1007/978-3-642-16552-8_24

[24] C. L. Asgeir Frimannsson, Next generation xliff: Simplify-clarify-and extend, in: Proceedings of the 1st XLIFF International Symposium, 2010.
URL http://www.localisation.ie/oldwebsite/xliff/resources/presentations/2010-10-04-next-generation-xliff-frimannsson.pdf

[25] H. J. van Zuylen, The REDO compendium: reverse engineering for software maintenance, John Wiley & Sons, Inc., 1993.

[26] C. H. M. J. James H. Cross II, Elliot J. Chikofsky, Reverse engineering, Vol. 35 of Advances in Computers, Elsevier, 1992, pp. 199 – 254. doi:http://dx.doi.org/10.1016/S0065-2458(08)60596-3.
URL http://www.sciencedirect.com/science/article/pii/S0065245808605963

[27] M. P. OBrien, T. M. Shaft, J. Buckley, An open-source analysis schema for identifying software comprehension processes, Psychology of Programming workshop PPIG 2001.

[28] E. S. Raymond, The Cathedral & the Bazaar: Musings on linux and open source by an accidental revolutionary, " O'Reilly Media, Inc.", 2001.

[29] D. E. Perry, A. A. Porter, L. G. Votta, Empirical studies of software engineering: a roadmap, in: Proceedings of the conference on The future of Software engineering, ACM, 2000, pp. 345–355.

[30] Microsoft, How to: Identify multiple validation errors when using the msxml 4.0 sax parser in visual basic, [Online; accessed 17-April-2015] (June 2014).
URL https://support.microsoft.com/en-us/kb/309535

[31] XML-org, Any parsers support multiple validation errors from validation?, [Online; accessed 17-April-2015] (June 2000).
URL http://lists.xml.org/archives/xml-dev/200006/msg00536.html

[32] Qt-Forum, Validating an xml file with a schema file, [Online; accessed 17-April-2015] (n.d.).
URL https://forum.qt.io/topic/9786/validating-an-xml-file-with-a-schema-file/3

[33] I. Song, P. S. Bayerl, Semantics of xml documents, [Online; accessed 08-February-2012] (2003).
URL http://www.uni-giessen.de/germanistik/ascl/dfg-projekt/pdfs/inseok03a.pdf

[34] A. Algergawy, M. Mesiti, R. Nayak, G. Saake, Xml data clustering: An overview, ACM Comput. Surv. 43 (4) (2011) 25:1–25:41.

doi:10.1145/1978802.1978804.
URL http://doi.acm.org/10.1145/1978802.1978804

[35] M. Currie, M. Geileskey, L. Nevile, R. Woodman, Visualising interoperability: Arh, aggregation, rationalisation and harmonisation, in: Proceedings of the 2002 International Conference on Dublin Core and Metadata Applications: Metadata for e-Communities: Supporting Diversity and Convergence, DCMI '02, Dublin Core Metadata Initiative, 2002, pp. 177–183.
URL http://dl.acm.org/citation.cfm?id=1344614.1344634

[36] K. Abdalla, A model for semantic interoperability using xml, in: Systems and Information Engineering Design Symposium, 2003 IEEE, 2003, pp. 107–111. doi:10.1109/SIEDS.2003.158012.

[37] A. Frimannsson, J. M. Hogan, Adopting standards-based xml file formats in open source localisation, Localisation Focus–The International Journal for Localisation 4 (4) (2005) 9–23.

[38] J. A. Mykkänen, M. P. Tuomainen, An evaluation and selection framework for interoperability standards, Inf. Softw. Technol. 50 (3) (2008) 176–197. doi:10.1016/j.infsof.2006.12.001.
URL http://dx.doi.org/10.1016/j.infsof.2006.12.001

[39] M. Desmond, M. Storey, C. Exton, Fluid source code views, in: Program Comprehension, 2006. ICPC 2006. 14th IEEE International Conference on, IEEE, 2006, pp. 260–263.

Dr. Asanka Wasala

Asanka Wasala is a Postdoctoral Researcher and the Lead Developer at the Localisation Research Centre at University of Limerick, Ireland. He is also a voting member of the OASIS XML Localization Interchange File Format (XLIFF) Standard Technical Committee. He has published in many different areas including Natural Language Processing, Software Localization, Data Exchange Standards, and Speech Processing. In 2004, Asanka graduated from the prestigious University of Colombo, Sri Lanka, receiving the best student award (gold medal) and being the only person to obtain a First Class qualification that year. After completing his BSc in Physical Science, he worked in the PAN Localization project as a Senior Research Assistant at the University of Colombo School of Computing, Sri Lanka. He received a full scholarship by Microsoft Ireland to pursue his Masters degree in University of Limerick, before transferring to the PhD program in 2010. He completed his PhD thesis on identification of limitations of localization data exchange standards, in 2013. His thesis won the LRC Best Thesis Award (2013) and a prize from the Microsoft Ireland.

Dr. Jim Buckley

Jim Buckley obtained a honours BSc degree in Biochemistry from the University of Galway in 1989. In 1994 he was awarded an MSc degree in Computer Science from the University of Limerick and he followed this with a PhD in Computer Science from the same University in 2002. He currently works as a lecturer in the Computer Science and Information Systems Department at the University of Limerick, Ireland. His main research interests are in theories of information seeking, software reengineering and software maintenance. In this context, he has published actively in many peer-reviewed journals/conferences/workshops. His work has involved extensive collaboration with companies in the Financial services sector, the flood mapping sector and with IBM, with whom he was a Visiting Scientist from 2006-2010. He was general Chair of WCRE 2011 and currently coordinates 2 research projects at the University: one in the area of software feature location and the other in architecture-centric re-engineering and evolution.

Mr. Reinhard Schäler

Reinhard Schäler has been involved in the localisation industry in a variety of roles since 1987. He is the founder and editor of Localisation Focus – The International Journal of Localisation, a founding editor of the Journal of Specialised Translation (JosTrans), a former member of the editorial board of Multilingual Computing (Oct 97 to Jan 07, covering 70 issues), a founder and CEO of The Institute of Localisation Professionals (TILP), and a member of OASIS. He has attracted more than €5.5m euro in research funding and has published more than 50 articles, book chapters and conference papers on language technologies and localisation. He has been an invited speaker at EU and international government-organised conferences in Africa, the Middle East, South America and Asia. He is a Principal Investigator in the Centre for Next Generation Localisation (CNGL), a lecturer at the Department of Computer Science and Information Systems (CSIS), University of Limerick, and the founder and director of the Localisation Research Centre (LRC) at UL, established in 1995. In 2009, he established The Rosetta Foundation and the Dynamic Coalition for a Global Localization Platform: Localization4all under the umbrella of the UN's Internet Governance Forum.

Dr. Chris Exton

Chris Exton is currently a lecturer in the department of Computer Science and Information Systems at University of Limerick and is the Research Director of the Localisation Research Centre. He holds a B.Sc. in Psychology and a Ph.D. in Computer Science from Monash University, Melbourne. He has worked extensively in the commercial software development field in a variety of different industries and countries included Software Engineering positions in Australia, Ireland and the UK, where he has worked in a number of diverse companies from electronic manufacturing to banking. In addition his academic positions includes a number of Schools and Departments including Monash University Australia, University College Dublin and Uppsala University, Sweden. He has worked on a number of research projects in the area of crowdsourcing, programmer psychology and software tools and more recently in the areas of software localisation and medical decision support systems. He has researched and published in these areas for over 15 years, as well as taking an active role as a reviewer for the International Journal of Software Maintenance and Evolution and The Software Quality Journal.

Dr. Asanka Wasala

Dr. Jim Buckley

Dr. Chris Exton

Mr. Reinhard Schäler

Highlights

- A framework that can evaluate practitioners' usage of data-exchange standards.
- Specific usage-analyses to identify possible refinements of such standards.
- An illustrative application of the framework on the XLIFF data-exchange standard.
- Results indicating core features/candidates for deprecation/modularization in XLIFF.