Process and Agent Classification Based Interoperability in the Emerging XLIFF 2.0 Standard

David Filip, Asanka Wasala
Localisation Research Centre,
University of Limerick, Ireland
david.filip@ul.ie, asanka.wasala@ul.ie

Abstract

In this paper, we are going to present the XLIFF 2.0 Agents classification that has been developed based on XLIFF 1.2 Interoperability research, proposed to the XLIFF Technical Committee (TC) and a simplified version of it adopted in Committee Specification Drafts of XLIFF 2.0.

We are also reviewing the state of the art and literature assessing the interoperability of localisation data interchange based on the XLIFF 1.2 OASIS standard. This paper shows how the XLIFF 1.2 based interoperability research provides valuable input for the XLIFF 2.0 standard development.

Keywords: *XLIFF 2.0, agents, classification, oasis, standard, interoperability*

1. State of the art and Literature Review

1.1 XLIFF Support in Tools

A survey conducted by Morado Vázquez and Filip (2012) reports the status of XLIFF support in Computer Aided Translation (CAT) tools. This report appeared twice so far – second time as Filip, D., Morado Vázquez, L. (2013) and tracks changes in XLIFF support in all major CAT tools. The survey was based on a questionnaire designed by the XLIFF Promotion and Liaison Sub-Committe of XLIFF TC and it is aimed mainly at CAT tool producers but also at owners of corporate XLIFF generators. The main objective of this survey is to iteratively collect information that is useful for understanding the level of tool support for XLIFF. The subcommittee is now retiring the full XLIFF 1.2 oriented questionnaire and develops a similar method for tracking XLIFF 2.0 support.

The survey reports a detailed characterisation 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 (e.g. given that a tool "actively uses" the <file> element, it does not necessarily imply that it conforms to the XLIFF mandatory requirements for the <file> element). Also it is important to note that this survey is based on the toolmaker's self-assessment which is not being technically verified beyond spotting and eventually reporting grave inconsistencies in their answers. The work of the XLIFF promotion subcommittee was originally inspired and partially based on the work of Bly (2010), who analysed XLIFF support in commonly used tools and presented a matrix containing tools and their level of support for individual XLIFF elements.

Despite serious limitations in the methodology used by Bly (2010), he concluded a valuable insight, namely that tool vendors can conform to standards but still lock in users with their tools. Moreover, he discussed various problems associated with the XLIFF standard, such as its inability to support all the features offered by tools, and its lack of tight definitions. All this notwithstanding, Bly is convinced that "XLIFF is the best (only) hope for interoperability."

In another study, Morado-Vázquez and Wolff (2011) present the Open Source CAT tool "Virtaal" that claims to support XLIFF. They compare its level of XLIFF support with the matrix presented by Bly. Bly's (2010) top-down analysis of XLIFF implementations show their level of support for different XLIFF elements.

Morado-Vázquez and Wolff conclude that Virtaal is better in terms of XLIFF support than the average XLIFF editing tools checked in Bly's study.

Interestingly, Morado-Vázquez and Wolff point out a weakness in Bly's methodology. Bly's analysis does not take into account the relative importance of different parts of the XLIFF specification. To address this issue Morado-Vázquez and Wolff propose a "weighted sum model" as a possible improvement, however details have not been included. Furthermore, they highlight the importance of an element's attribute and attribute values for tool interoperability. Similar to Bly's, the exact methodology used to evaluate their tool, the test suites or the use of the term "support" are not included in their publication. Although they mention the use of Bly's analysis methodology to evaluate Virtaal, the paper does not make explicit references to Bly's methodology or test suites for evaluating Virtaal.

In Anastasiou and Morado-Vázquez (2010), several interoperability tests were performed with three XLIFF compliant CAT tools. Like Bly (2010), they classified selected tools into two categories: XLIFF converters (i.e. generators or extractors) 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 four combinations: for each converter they wanted to see if the other two editing tools could edit the generated content. The researchers' methodology involved five 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 localisation 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 too.

One of the reasons behind lack of tool support of XLIFF standard is the absence of a proper conformance clause in XLIFF (Filip 2011; Anastasiou and Morado-Vázquez 2010). This also reflects Bly's 'lack-of-definition' finding. Anastasiou (2011) stresses that "conformance clauses should include criteria about compliance with both Localisation and Semantic Web standards."

1.2 Different Levels of Tool Support

In this research, we compiled a large XLIFF corpus by collecting over 3000 XLIFF files with the aid of CNGL industrial partners and by crawling openly available XLIFF files in the web. We then analysed the XLIFF files for their XLIFF feature usage characteristics. In the following, we present the overall frequency distribution of XLIFF elements in our corpus in Fig 1.

An analysis of the above Graph reveals a connection between the lack of tools support for certain XLIFF elements and the frequency of use of XLIFF elements. Our research revealed that many XLIFF features are either not supported or only partially supported by tools (Anastasiou 2010; Bly 2010; Lieske 2011). This has inevitably led to interoperability issues.

According to Shan and Kesan (2008) a valid reason for tool developers not to offer full interoperability is the lack of need to support all the features in their tools. This seems to be due to two main reasons:

1. Business case requirements

Depending on the requirements of different business cases, different tools have been implemented to support different parts of the XLIFF specification. There are localisation tools that do not support XLIFF at all, which is mainly due to the fact that there is no strong business case demanding XLIFF support from these tool vendors.

2. Complexity and limitations of the standard

Although XLIFF's formal tool compliance is easy to achieve, complete XLIFF feature implementation in tools is difficult due to the complexity of the standard (Anastasiou and Morado-Vázquez 2010). Eventually the document conformance is relatively well addressed in the XLIFF 1.2 OASIS standard despite the lack of a formal Conformance Clause. However, there are virtually no conformance hints targeting application conformance. We will see how the XLIFF TC learnt from this and introduced an explicit application conformance target in its current Public Review Drafts (Comerford, T., Filip, D., Raya, R.M., Savourel, Y.; Eds. 2013a, b). The application conformance target was explicitly added between the 1st and 2nd public reviews along with the Processes and Agents classification, based on the discussions concluded in the June 2013 XLIFF TC face to face meeting.

2. Agent Classification: Proposed Methodology I

Shah and Kesan (2008) state that users expect 100% interoperability among implementations for various reasons including actual requirements as well as avoiding potential problems. In order to achieve 100% interoperability among agents, ideally the agents need to implement all the features specified in the standard. However, as we identified in Section 1.2, this is not realistic, at least not in the XLIFF 1.2 case.

In this paper, we propose some alternative methodologies for improving interoperability among agents. This is by classifying agents based on their supported features or process driven feature subsets.

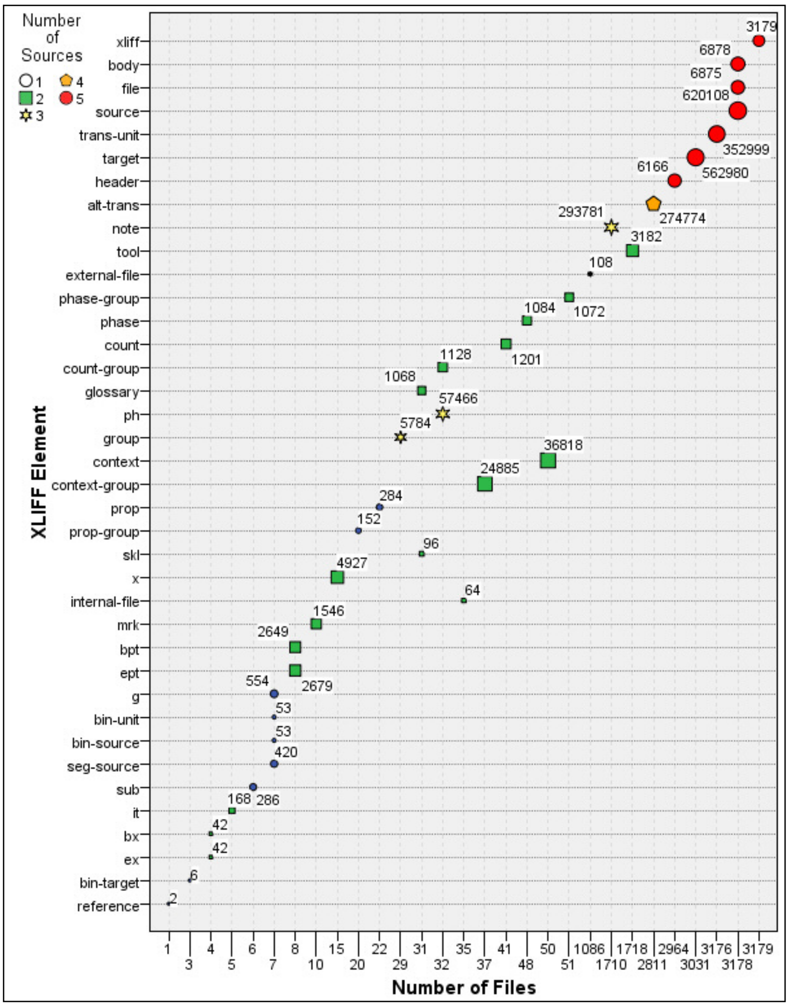

Figure 1. XLIFF 1.2 Feature Distribution in Files and Organisations

The first presented methodology is based on the assumption that agents that implement the same set of features will have full interoperability among those agents.

The features used in both a majority of organisations and files can be considered as most important features of the standard in terms of their usage. Then based on the importance, features can be categorised into several levels. This categorisation should be carried out by the standardisation committee, such as the XLIFF Technical Committee (TC). The frequency distribution of features can be used as an aid to define different levels. An example classification is given below.

level 4:	xliff, file, header, body, trans-unit , source, target;
level 3:	alt-trans, note, ph, group;
level 2:	tool, external-file, phase-group, phase, count, count-group, glossary, context, context-group, prop, prop-group, skl, x, internal-file, mrk, bpt, ept;
level 1:	g, bin-unit, bin-source, seg-source, sub, it, bx, ex, bin-target, references.

The above example has been mainly derived by analysing the XLIFF 1.2 elements distribution in our corpus. Levels have been primarily defined based on the importance of features. For example, under the level 4, the features used by all five organisations are listed. From the corpus, it is evident that these features are also used in majority of files. Similarly, the features used by at least 4 organisations have been categorised as level 3 and so forth.

It is important to note that elements cascade from top to bottom in these levels. For example, elements that belong to level 3 include all the elements listed under level 4 in addition to the explicitly listed elements (i.e. the complete feature set of level 3 consists of all level 4 elements and elements: alt-trans, note, ph, group).

Once different levels have been established in agreement with the TC of the standard, agents can be classified into these levels based on agent's feature support. An agent that has implemented features "xliff, file, header, body, trans-unit , source, target" is classified under the level 4, whereas an agent that has implemented features "xliff, file, header, body, trans-unit , source, target, alt-trans, note, ph, group" is classified as level 3. Therefore, an agent that has implemented all the features is classified as a level 1 agent.

After defining the levels, the next major step involves preparations of test suites. Separate test suites have to be developed covering the feature subset defined for each of the above levels. Then these test suites can be used to evaluate agents' level of XLIFF support. Finally, the agents can be classified by their level of XLIFF support based on the test results.

However, it is likely that agents may only support a sub-set of features of each level. In such scenario, the TC should come to agreements on essential tests of each level that need to be passed by an agent, in order to be able to classify it under a certain level.

This proposed methodology has been devised based on XLIFF 1.2 and would be usable as is if XLIFF TC continued in development of an interchange standard backwards compatible with 1.2. However, XLIFF 2.0 uses a different set of elements and cannot therefore use an XLIFF 1.2 elements based categorization of agents. The solution for XLIFF 2.0 however builds on lessons learnt from the XLIFF 1.2 based interoperability research.

3. Agent Classification: Proposed Methodology II

This agent classification has been developed specifically for XLIFF 2.0. Instead of generic support levels without any specific process focus, as discussed in Section 2, this classification methodology is based on the generalized Business Process Model of the XLIFF payload and metadata interchange. See Fig 2. This approach was inspired by the business process driven development of the UBL standard (Bosak, J., McGrath, T., Holman, K.G.; Eds., 2006, 2013).

Based on lessons learnt from the XLIFF 1.2 adoption that have been extensively discussed in Section 1, the XLIFF TC decided to specify a small core specification as the smallest common denominator and thus a secure base for interoperability.

Based on XLIFF TC discussions and a formal ballot, the TC decided to include in core only elements and attributes that are essential for extraction, translation

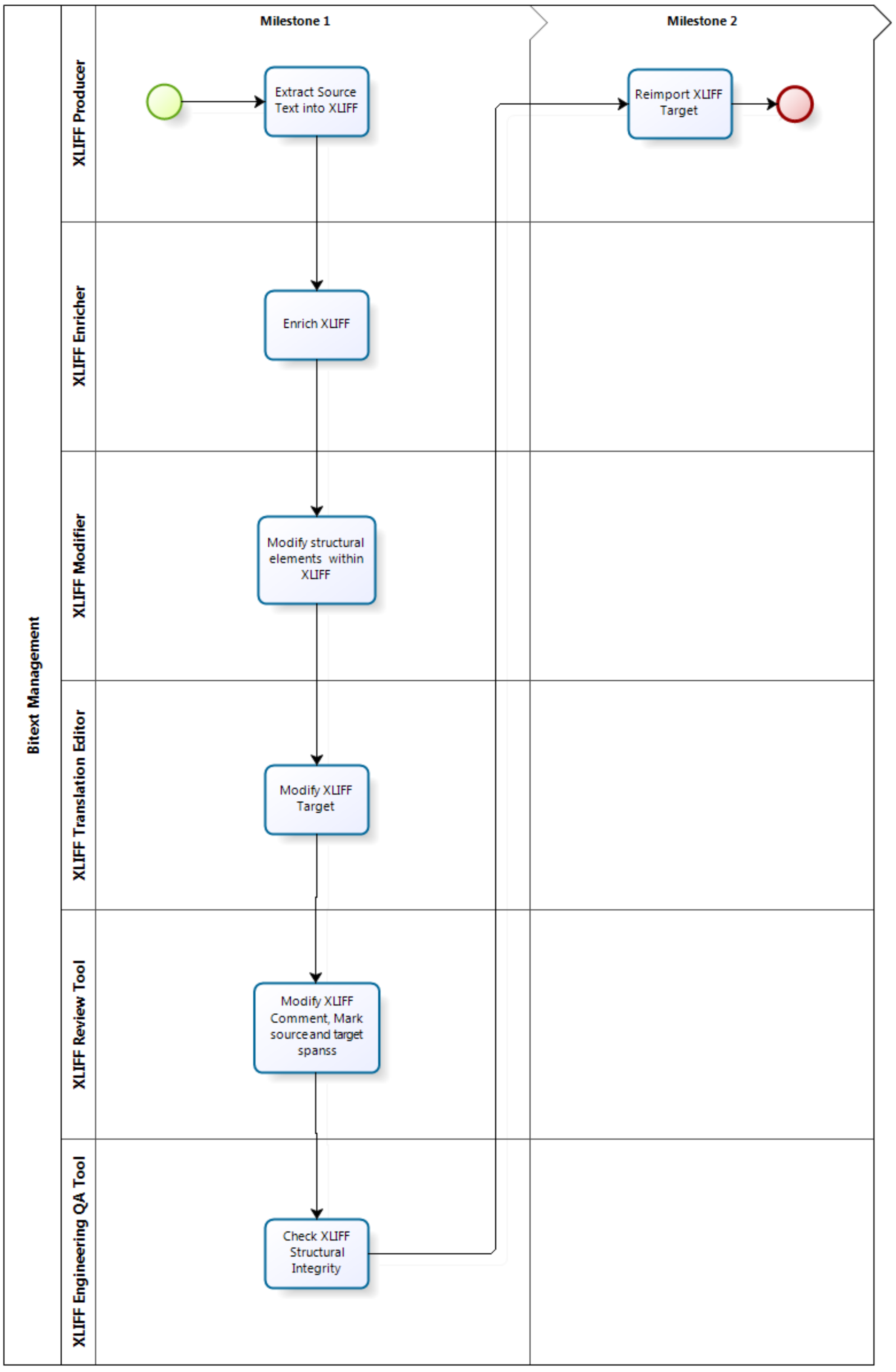
Bitext Management
XLIFF Producer
XLIFF Enricher
XLIFF Modifier
XLIFF Translation Editor
XLIFF Review Tool
XLIFF Engineering QA Tool
Milestone 1
Milestone 2
Extract Source Text into XLIFF
Enrich XLIFF
Modify structural elements within XLIFF
Modify XLIFF Target
Modify XLIFF Comment, Mark source and target spanss
Check XLIFF Structural Integrity
Reimport XLIFF Target

Figure 2. Generalized Business Process Model of the XLIFF Payload and Metadata Interchange

and merging back of content into the source format in the target natural language. Another possible approach would have been to look at all possible process areas that are being served by XLIFF as the interchange format and include basic functionality from each area in the core while handling the advanced functionalities in modules and possibly extensions. This approach however was not chosen by the TC. So there is, for instance, not even a basic size restriction mechanism in XLIFF core, while a comprehensive and extensible general size and length restriction mechanism has been specified as one of XLIFF 2.0 modules.

Among other popular features known from XLIFF 1.x that did not make it into the core of XLIFF 2.0, we can mention the `<alt-trans>` element. Instead of the former core element for candidate translations (and other related versions) there is a comprehensive Translation Candidates module that however sheds the semantic overload of the original XLIFF 1.2 element.

3.1 The original proposal for Normative Process and Agent related Terminology in XLIFF 2.0

The original proposal has been presented to the TC on various occasions the last time as an Initial Public Review Comment (Filip 2013).

The following has been proposed as a set of normative process and agent definitions and requirements, along with non-normative notes and warnings:

[Definitions]

Agent – a tool that does anything to an XLIFF file from extract to merge inclusively.
Extract/Extraction – the process of encoding localizable content from a native content or User Interface format as XLIFF payload, so that localizable parts of content in the source language are available for translation into the target language along with necessary context.
Extractor (Agent) – an Agent that performs the Extraction process.

Merge – the process of importing XLIFF payload back to the originating native format, based on the full knowledge of the extraction mechanism, so that the localized content or User Interface strings replace the source language in the native format.

Merger (Agent) –an Agent that performs the Merge process.

Warning:

Unless specified otherwise, Merger is deemed to have the same knowledge of the native format as the Extractor throughout the specification. Mergers independent of Extractors can succeed, but it is out of scope of this specification to specify interoperability for merging back without the full extractor knowledge of the native format.

[Definitions]

Enrich/Enriching – the process of associating module and extension based metadata and resources with the extracted XLIFF payload.

[Processing Requirements] PR:

Enriching MAY happen at the time of extraction.

[Definitions]

Enricher (Agent) – an Agent that performs the Enriching process.

Note:

Extractor knowledge of the native format is not assumed while Enriching.

[Definitions]

Modify/Modification – the process of changing core XLIFF structural elements and bin-file module elements.

Modifier (Agent) – an Agent that performs the Modification process.

[Processing Requirements] PR:

Structural elements MAY be Modified and Enriched at the same time. Modifier MUST be able roll back the core and bin-file structure of an XLIFF that it has previously modified.

Note:

Enricher or Extractor knowledge of the native format is not assumed while Modifying.

Warning:

Rollback of Modifications performed by a different Modifier is out of scope.

[Definitions]

Edit Source/Source Editing – the process of changing payload of <source> children of

<segment> elemensts.

Source Editor (Agent) – an Agent that performs Source Editing.

[Processing Requirements] PR:

Source editing MAY be performed at the time of Modification.

[Definitions]

Translate/Translation - a rendering of the meaning of source text, expressed in the target language. As an XLIFF based process it means creating and changing <target> children of <segment> elements.

Translation Editor (Agent) – an Agent that performs the Translation process.

Review/Revision – the process of creating or changing any annotation elements, attributes or values specified in core, modules or extensions. This includes marking spans of <source> and <target> children of <segment> elements.

Revision Agent – an Agent that performs the Revision process.

[Processing Requirements] PR:

Revision MAY be performed at the time of Translation.

Revision MAY be performed at the time of Modification.

[Definitions]

Validate/Validation – the process of checking XLIFF payload against any rules specified via the Validation Module or any general, core or module specific Processing Requirements.

Validator (Agent) – an Agent that performs the Validation process.

[Processing Requirements] PR:

Validators MAY add test results via the Validation Module.

Validator MUST not modify any other XLIFF elements, attributes or values.

Validation MUST NOT be performed at the time of Merge.

Note:

Extractor/Merger Agents will benefit from use of a Validator. Even though the Validator will often be a part of the same tool/platform, it is important to distinguish between validation and the actual Merge. Validation will routinely precede Merge, while failed Validation will lead to exception handling, such as returning the XLIFF file to a Revision Agent or Translation Editor, Modifier etc.

3.1 Further Developments

The original Processes and Agents proposal as presented in Section 3.1 has developed based on XLIFF TC discussions, notable the face to face discussion at the second FESGILTT event in London, June 2013. The proposal also had to adapt to developments of the XLIFF 2.0 core and modules specification.

The first TC approved draft of the solution appeared in the Second Public Review of the XLIFF 2.0 Committee Specification Draft (Comerford, T., Filip, D., Raya, R.M., Savourel, Y.; Eds. 2013a). The TC did not agree to inclusion of Processing Requirements that would prescribe a partial ordering for the XLIFF facilitated processes and the TC also did not use the full detailed scale of process and agent definitions.

As result, the specification only recognizes Extractors, Mergers, Enrichers, Modifiers, and Writers, as specific types of Agents. All Processing Requirements targeting unspecified Agents are deemed to target any type of agents irrespective of process specialisation. This debate also led to explicit differentiation between static Constraints (on top of data type constraints based on elements and attribute definitions that can be expressed in schema) and Agents targeting Processing Requirements. Processing Requirements in the specification now always specify what types of transformations unspecified or specialized Agent can perform on the static documents.

Other types of agents were not deemed necessary for the normative provisions that the standard specification has to provide. So Validators and Revision Agents are considered specific types of Enrichers. Source and Translation Editors are considered special types of Modifiers. Interestingly, source payload editing is not an allowed XLIFF transformation as per the current XLIFF 2.0 Working Draft; source content can be only enriched or resegmented but not modified.

Those subtypes that did not make it into the normative specification might be still useful to discern in the context of a non-normative interoperability debate, for instance for describing use cases in a language accessible for industry.

Writers that did not feature in the original proposal are a superset of Extractors, Enrichers, and Modifiers. However, there might as well be Writers that are neither of the former, since XLIFF is intended as an interchange format and not as a processing format.

3.3 The Resulting Solution

The following is the resulting XLIFF 2.0 classification of Processes and Agents that is included in the XLIFF 2.0 specifications as of the 2nd Public Review Draft (Comerford, T., Filip, D., Raya, R.M., Savourel, Y.; Eds. 2013a).

Definitions

Agent

any application or tool that generates (creates), reads, edits, writes, processes, stores, renders or otherwise handles ***XLIFF Documents***.

Agent is the most general application conformance target that subsumes all other specialized user agents disregarding whether they are defined in this specification or not.

Enrich, Enriching

the process of associating module and extension based metadata and resources with the ***Extracted*** XLIFF payload

Processing Requirements

- ***Enriching*** MAY happen at the time of ***Extraction***.

Note

Extractor knowledge of the native format is not assumed while ***Enriching***.

Extract, Extraction

the process of encoding localizable content from a native content or User Interface format as XLIFF payload, so that localizable parts of the content in the source language are available for ***Translation*** into the target language along with the necessary context information

Extractor, Extractor Agent

any ***Agent*** that performs the ***Extraction*** process

Merge, Merging

the process of importing XLIFF payload back to the originating native format, based on the ***full knowledge*** of the ***Extraction*** mechanism, so that the localized content or User Interface strings replace the source language in the native format

Merger, Merger Agent

an ***Agent*** that performs the ***Merge*** process

Warning

Unless specified otherwise, any ***Merger*** is deemed to have the same knowledge of the native format as the ***Extractor*** throughout the specification.

Mergers independent of ***Extractors*** can succeed, but it is out of scope of this specification to specify interoperability for ***Merging*** back without the full ***Extractor*** knowledge of the native format.

Modify, Modification

the process of changing core and module XLIFF structural and inline elements that were previously created by other ***Writers***

Processing Requirements

- XLIFF elements MAY be ***Modified*** and ***Enriched*** at the same time.

Note

Extractor or ***Enricher*** knowledge of the native format is not assumed while ***Modifying***.

Modifier, Modifier Agent

an ***Agent*** that performs the ***Modification*** process

Warning

Unless specified otherwise, any ***Merger*** is deemed to have the same knowledge of the native format as the ***Extractor*** throughout the specification.

Mergers independent of ***Extractors*** can succeed, but it is out of scope of this specification to specify interoperability for ***Merging*** back without the full ***Extractor*** knowledge of the native format.

Translation, Translate
a rendering of the meaning of the source text, expressed in the target language

Writer, Writer Agent
an ***Agent*** that creates, generates, or otherwise writes an ***XLIFF Document*** for whatever purpose, including but not limited to ***Extractor***, ***Modifier***, and ***Enricher Agents***.

Note
Since XLIFF is intended as an exchange format rather than a processing format, many applications will need to generate ***XLIFF Documents*** from their internal processing formats, even in cases when they are processing ***XLIFF Documents*** created by another ***Extractor***.

These definitions provide a normative base for writing Processing Requirements throughout the whole XLIFF 2.0 specification and allow for specific application conformance targeting in the Conformance Section:

2. Application Conformance

a XLIFF *Writers* MUST create conformant *XLIFF Documents* to be considered XLIFF compliant.

b *Agents* processing conformant *XLIFF Documents* that contain custom extensions are not REQUIRED to understand and process non-XLIFF elements or attributes. However, conformant applications SHOULD preserve existing custom extensions when processing conformant *XLIFF Documents*, provided that the elements that contain custom extensions are not removed according to XLIFF Processing Requirements or the extension's own processing requirements.

c All *Agents* MUST comply with Processing Requirements for otherwise unspecified *Agents* or without a specifically set target *Agent*.

d Specialized *Agents* defined in this specification - this is *Extractor*, *Merger*, *Writer*, *Modifier*, and *Enricher Agents* - MUST comply with the Processing Requirements targeting their specifically defined type of *Agent* on top of Processing Requirements targeting all *Agents* as per point 3.[c] above.

e XLIFF is a format explicitly designed for exchanging data among various *Agents*. Thus, a conformant XLIFF application MUST be able to accept *XLIFF Documents* it had written after those *XLIFF Documents* were *Modified* or *Enriched* by a different application, provided that:
i. The processed files are conformant *XLIFF Documents*,
ii. in a state compliant with all relevant Processing Requirements.

4. Defining "Support"

The term "support" is a widely used term related not to XLIFF implementations (tools), it is a key interoperability term in general. A few examples pertaining to XLIFF specifically are given below:

- *the tool supports XLIFF;*
- *the tool supports feature X (e.g. tool supports in-line mark-up elements);*
- *the tool partially supports feature X.*

As discussed in detail in Section 1.1 lack of a proper definition for this term had led to many confusions. It can also be conjectured that the term "support" has been used by some of the tool developers just as a marketing slogan. Therefore, we recommend that one of the first steps towards addressing interoperability in the new incarnation of XLIFF is a rigid definition of the term "support" in the above contexts.

4.1 Defining "Support" Based on a Feature Complete Reference Implementation

One of the possible approaches to defining support assumes that a comprehensive open-source reference implementation exists, and given that this assumption has been fulfilled, "support" may be defined as follows:

- The tool must operate on X as expected and described by the XLIFF specification and in all possible variations of X (e.g. for all possible

attribute/value combinations),
AND
- The tool must operate on X in the same manner as the reference implementation does
OR
- The outcome of the tool manipulation of the X must be equivalent to the outcome of the feature manipulation by the reference implementation in all possible manipulation scenarios.

In order to claim that a tool ***supports*** XLIFF:

- Given any possible valid variation of XLIFF content, if and only if the outcome of the manipulated content by the tool is equivalent to the outcome of the same manipulation performed by the reference implementation, the tool ***supports*** XLIFF.

As far as the above approach to defining the term support depends on the existence of a feature complete reference implementation, it is not realistic to rely on the above definition in practice, at least not until such a comprehensive open source implementation that is backed by industry consensus exist.

Anyway, given the modular character of XLIFF 2.0 and following the current efforts of implementers that are likely to provide Statements of Use that are required for XLIFF becoming a Candidate Standard and an official OASIS Standard in the due course, it is possible that there soon (early 2014) will be a comprehensive Open Source implementation of the core. However, if we are looking for coverage for the whole XLIFF 2.0 specification including its 8 modules, we are more likely looking for an ecosystem of Open Source and closed source tools. Also standards specifications must be implementation agnostic. Of course, as standards they must be implementable, yet must not prescribe any specific implementation. Therefore "support" must be defined at a lower theoretical level that is grounded in the specification itself without referencing a particular implementation.

Defining "Support" Based on the Normative Provisions of the Specification

The new XLIFF specification contains a dedicated Conformance section. This is partially to conform to a new non-negotiable requirement that OASIS introduced as part of its TC process and, more importantly, due to the lessons learnt with compromised interoperability of XLIFF 1.x implementations. It is fair to say that while XLIFF 1.2 had covered static validity fairly well due to having relatively rigid element definitions and XML Schema based validation there had been zero guidance for application conformance. The good practices of static validation were of course adopted also for XLIFF 2.0. However, on top of explicit static document conformance, the XLIFF 2.0 specification now explicitly addresses XLIFF agents as its application conformance targets.

In analogy with the reference implementation based attempt, we could try and define "support" as follows.

In order to claim that a tool ***supports*** feature X:

- The tool must operate on X as expected and described by the XLIFF specification and the tool outcomes must satisfy all static conditions and constraints as well as processing requirements related to the feature, that in all possible variations of X (e.g. for all possible attribute-value combinations).

In order to claim that a tool fully ***supports*** XLIFF, the tool must ***support*** all the XLIFF features.

The above can be called a maximalist definition of "support" that is unfortunately not very useful in real life scenarios. Luckily the spec now works with different subsets of agents based on what processes the agents are capable of ***supporting***.

1 For processing requirements addressing any or unspecified XLIFF Agents, all agents must conform to the given Processing Requirement
2 For processing requirements addressing XLIFF Writers, all writer agents must conform to the given Processing Requirement. As discussed above the Writers are a superset of Extractors, Enrichers, and Modifiers. However, there might as well be Writers that are neither of the former, since XLIFF is intended as an interchange format and not as a processing format.
3 For processing requirements addressing XLIFF Extractors, all Extractor agents must conform to the given Processing Requirement.
4 For processing requirements addressing XLIFF Enrichers, all Enricher agents must

conform to the given Processing Requirement.

5 For processing requirements addressing XLIFF Modifiers, all Modifier agents must conform to the given Processing Requirement.

6 For processing requirements addressing XLIFF Mergers, all Merger agents must conform to the given Processing Requirement.

Thus the defined support extent can be effectively sub-setted, so that specialized tools can claim support in an unequivocal and sensible way without the need to support irrelevant features or transformations.

So for instance a tool that specializes in Enriching XLIFF Documents with Translation Candidates does not need to worry about a significant subset of Processing Requirements that are addressing Modifiers, Extractors, and Mergers, while it must conform to all Processing Requirements set forth for (unspecified) Agents, Writers, and Enrichers.

5. Conclusion

The approach to assessing support described in Section 4.2 has immense advantages; it lowers the cost of standards based interoperability within and across supply chains. Instead of requiring that all tools of a certain level support all touched elements and attributes in all respects, adopters can look into building modular workflows of specialized contributing tools between the process bracket of Extraction and Merging of the translatable content. This surpasses the interchange paradigm of XLIFF 1.2, which was intended as a non-transitive "fire-and-die" format, whereas XLIFF 2.0 explicitly targets the whole tool chain between and including Extraction and Merging back of content.

If your Extractor/Merger supports the XLIFF Core and a module X, you are looking for Modifiers and Enrichers that support the core in their respective capacities and are capable of processing the required module. You can build a workflow that contains a tool that is unaware of a specific well defined subset of functionality of core or modules, yet you can be sure that if the specialized tool had done just its own job and stuck to all requirements relevant to its own transformations, the rest of the payload and metadata won't be harmed.

Because XLIFF 2.0 Core is the smallest possible common denominator, support for all static constraints and Processing Requirements targeting just the relevant type of Agents (as explained above) is binary (yes/no) and non-negotiable. Nevertheless, the Core and Modules can be supported by tools in different capacities.

The modular and process classification based approach to conformance targets allows the assessment of tools support in a practical way based on normative requirements set out in the specification itself, which was not possible in the XLIFF 1.2 predecessor standard. XLIFF 2.0 shall be a better standard thanks to the lessons learnt from XLIFF 1.2 adoption.

References

Anastasiou, D. (2010) 'Open and flexible localization metada', *MultiLingual,* 21(4), 50-52.

Anastasiou, D. (2011) 'The Impact of Localisation on Semantic Web Standards', *European Journal of ePractice,* 12(March/April 2011), 42-52.

Anastasiou, D. and Morado-Vázquez, L. (2010) 'Localisation Standards and Metadata' in Sánchez-Alonso, S. and Athanasiadis, I. N., eds., *Metadata and Semantic Research*, Springer Berlin Heidelberg, 255-274.

Berges, I., Bermudez, J., Goñi, A. and Illarramendi, A. (2010) 'Semantic interoperability of clinical data', in *Proceedings of the First International Workshop on Model-Driven Interoperability*, Oslo, Norway, 1866275: ACM, 10-14.

Bly, M. (2010) 'XLIFF: Theory and Reality: Lessons Learned by Medtronic in 4 Years of Everyday XLIFF Use', in *1st XLIFF Symposium*, Limerick, Ireland, 22 Sept, University of Limerick.

Bosak, J., McGrath, T., Holman, K.G. (Eds.) (2006) *Universal Business Language v2.0* [online], OASIS Standard. ed, Standard, OASIS, available: http://docs.oasis-open.org/ubl/os-UBL-2.0/UBL-2.0.html [accessed 11 Dec 2013].

Bosak, J., McGrath, T., Holman, K.G. (Eds.) (2013) *Universal Business Language v2.1* [online], OASIS Standard. ed, Standard, OASIS, available: http://docs.oasis-open.org/ubl/os-UBL-2.1/UBL-2.1.xml [accessed 11 Dec 2013].

Comerford, T., Filip, D., Raya, R.M., Savourel, Y. (Eds.) (2013a) *XLIFF Version 2.0* [online], Committee Specification Draft 02 / Public Review Draft 02. ed, Standard, OASIS, available: http://docs.oasis-open.org/xliff/xliff-core/v2.0/csprd02/xliff-core-v2.0-csprd02.html [accessed 17 Oct 2013].

Comerford, T., Filip, D., Raya, R.M., Savourel, Y. (Eds.) (2013b) *XLIFF Version 2.0* [online], Committee Specification Draft 01 / Public Review Draft 01. ed, Standard, OASIS, available: http://docs.oasis-open.org/xliff/xliff-core/v2.0/csprd01/xliff-core-v2.0-csprd01.html [accessed 22 Jul 2013].

Filip, D. (2011) 'XLIFF 2.0', in *Multilingual Web Workshop*, Pisa, Italy, 4-5 Apr.

Filip, D. (2013) 'Re: XLIFF 2.0 csprd01 comment by David Filip - classification of processes and agents to improve precision of conformance statements', available: https://lists.oasis-open.org/archives/xliff/201305/msg00054.html [accessed 11 Dec 2013].

Filip, D., Morado Vázquez, L. (2013) *XLIFF Support in CAT Tools*, Subcomittee Report 2, XLIFF State of the Art, OASIS XLIFF TC, available: http://www.localisation.ie/resources/SurveyReport2ndEditionApproved.pdf [accessed 3 Dec 2013].

Frimannsson, A. and Lieske, C. (2010) 'Next Generation XLIFF: Simplify-Clarify-and Extend', in *1st XLIFF International Symposium* Limerick, Ireland, 22 Sept, University of Limerick.

Heiler, S. (1995) 'Semantic interoperability', *ACM Comput. Surv.,* 27(2), 271-273.

Imhof, T. (2010) 'XLIFF – a bilingual interchange format', in *MemoQ Fest*, Budapest, Hungary, 5-7 May.

Lewis, G. A., Morris, E., Simanta, S. and Wrage, L. (2008) 'Why Standards Are Not Enough to Guarantee End-to-End Interoperability', in *Proceedings of the Seventh International Conference on Composition-Based Software Systems (ICCBSS 2008)*, 1343630: IEEE Computer Society, 164-173.

Lewis, D., O'Connor, A., Molines, S., Finn, L., Jones, D., Curran, S., & Lawless, S. (2012a). 'Linking localisation and language resources'. In *Linked Data in Linguistics*. Springer Berlin Heidelberg, 45-54.

Lewis, D., O'Connor, A., Zydron, A., Sjögren, G., & Choudhury, R. (2012b). 'On Using Linked Data for Language Resource Sharing in the Long Tail of the Localisation Market'. In *LREC,* 1403-1409.

Li, W. and Li, S. (2004) 'Improve the semantic interoperability of information', in *Proceedings. 2004 2nd International IEEE Conference Intelligent Systems*, 22-24 Jun, 591-594.

Morado Vázquez, L. and Filip, D. (2012) 'XLIFF Support in CAT Tools', available: http://www.localisation.ie/resources/XLIFFSotAReport_20120210.pdf [accessed 5 Mar 2012].

Morado-Vázquez, L. and Wolff, F. (2011) 'Bringing industry standards to Open Source localisers: a case study of Virtaal', *Tradumàtica,* 9, 74-83.

Ouksel, A. M. and Sheth, A. (1999) 'Semantic interoperability in global information systems', *SIGMOD Rec.,* 28(1), 5-12.

Park, J. and Ram, S. (2004) 'Information systems interoperability: What lies beneath?', *ACM Trans. Inf. Syst.,* 22(4), 595-632.

Ray, S. R. (2009) 'Healthcare interoperability - lessons learned from the manufacturing standards sector', in *Automation Science and Engineering, 2009. CASE 2009. IEEE International Conference on*, 22-25 Aug, 88-89.

Sartipi, K. and Yarmand, M. H. (2008) 'Standard-based data and service interoperability in eHealth systems', in *ICSM 2008. IEEE International Conference on Software Maintenance* 28 Sept-4 Oct, 187-196.

Savourel, Y., Reid, J., Jewtushenko, T., Raya, R.M. (Eds.) (2008) *XLIFF Version 1.2* [online], OASIS Standard. ed, Standard, OASIS, available: http://docs.oasis-open.org/xliff/v1.2/os/xliff-core.html [accessed 3 Dec 2013].

Shah, R. and Kesan, J. (2008) '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*, Cairo, Egypt, 1509141: ACM, 219-225.

Lieske, C. (2011) 'Insights into the future of XLIFF', *MultiLingual,* 22(5), 51-52.

Wasala, A., Filip, D., Exton, C. and Schäler, R. (2012a) 'Making Data Mining of XLIFF Artefacts Relevant for the Ongoing Development of the XLIFF Standard', in *3rd International XLIFF Symposium, FEISGILTT 2012*, Seattle, USA, 17-19 Oct.

Waters, J., Powers, B. J. and Ceruti, M. G. (2009) 'Global Interoperability Using Semantics, Standards, Science and Technology (GIS3T)', *Computer Standards & Interfaces,* 31(6), 1158-1166.