

1st XLIFF International Symposium

22 September 2010, Limerick, Ireland

LCX and XLIFF: A Comparison

Asanka Wasala

Dag Schmidtke

Reinhard Schäler

Microsoft

Localisation Research Centre 

- Introduction
- XLIFF and LCX Usage
- XLIFF and LCX File Format Comparison
 - Results: XLIFF & LCX Features
- XLIFF ↔ LCX Conversion
- Discussion
 - XLIFF & Interoperability Issues
- Summary

Introduction

- Surveying localisation file formats and tools
- Assessing extent of format support in different localisation tools
 - SDL Passolo, Trados 2009, Alchemy Catalyst 8
- Review and Comparison of
 - Open Standard Localisation File Format (XLIFF)
 - Proprietary Localisation File Format (LCX)
- Investigation of extent of conversion possible between XLIFF and LCX

- Address issues related to exchange, storing and transport of localisation data and meta-data
- An initiative within OASIS: Organization for the Advancement of Structured Information Standards
- XLIFF:
 - Allow interoperability between tools (eliminate vendor lock-in)
 - Support for localisation workflow
 - Eliminates the need to deal with many formats
 - Meant for both program data and documentation content
 - Based on XML: All advantages unique to XML
- Gaining increased acceptance within the localisation community

Introduction

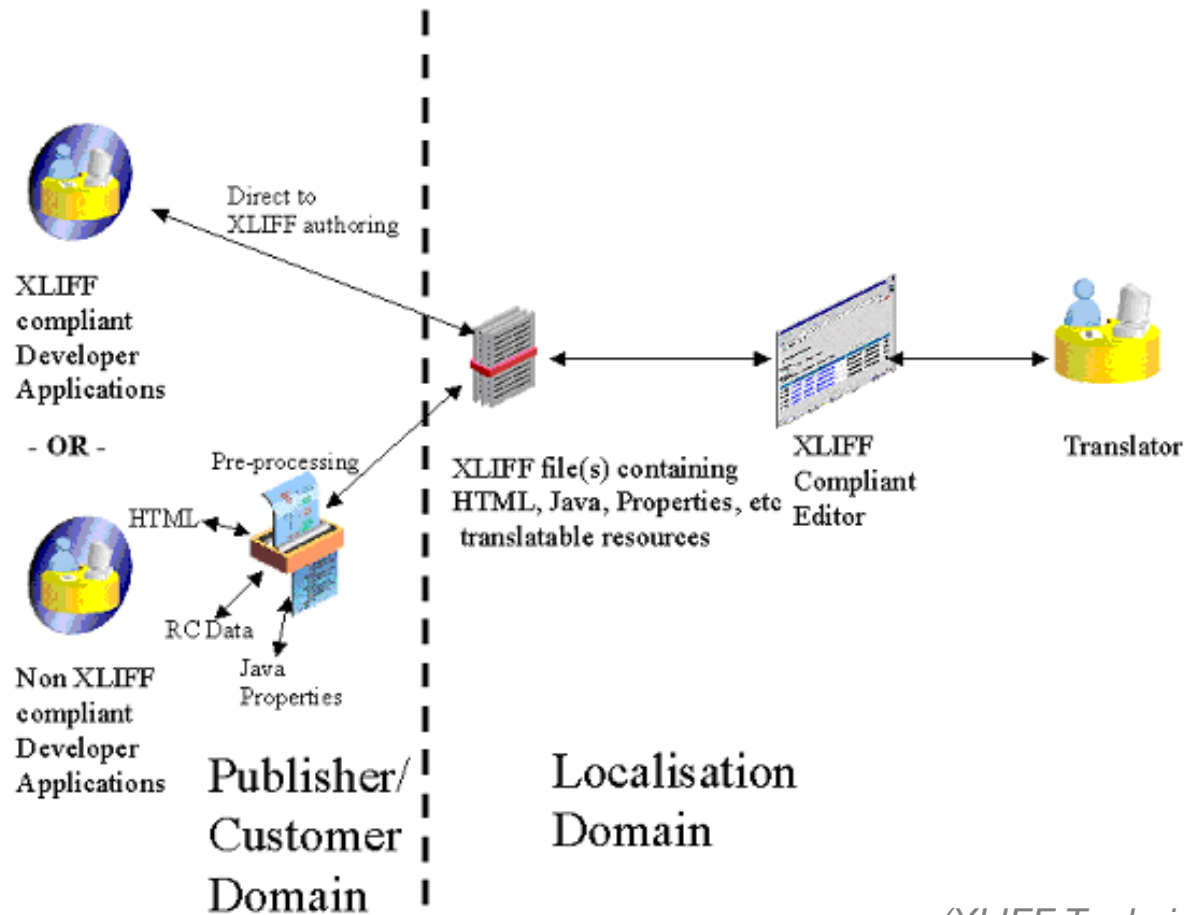
LCX: Localisation Content Exchange File Format

- Microsoft-internal file format to store software localisation data and meta-data
 - Localisation project and transportation format
- A series of XML files (a.k.a LCX Container)
- Data transparency through a complete object model
 - Specify how to access localisation content programmatically
- Has a unique extension mechanism
- Extensive use of binary data
- Much simpler compared to XLIFF

LCX & XLIFF Usage

XLIFF Usage

- Simple Workflow :
 - Transient XLIFF files *(Viswananda and Scherer, 2004).*

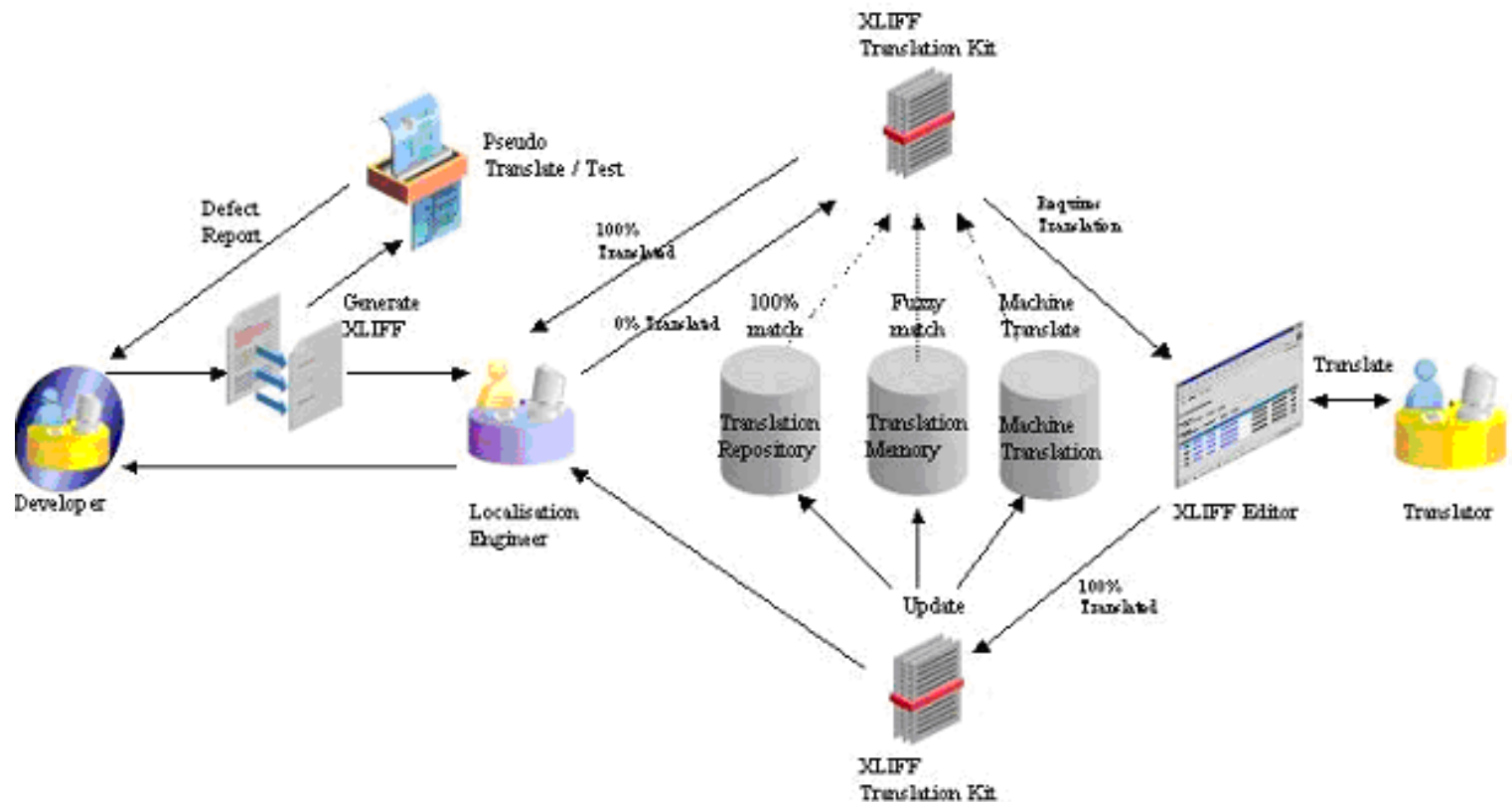


(XLIFF Technical Committee, 2008b)

LCX & XLIFF Usage

XLIFF Usage

- Advanced Workflow:
 - Permanent XLIFF files (Viswananda and Scherer, 2004).



(XLIFF Technical Committee, 2008b)

LCX & XLIFF Usage

LCX Usage

Simple workflow

1. Generate LCX source files (at MS)

1. Process localizable files (EXE, DLL, HTML, etc)
2. Incorporate developer and localization pilot team comments, instructions, validation rules
3. Create combined LCX files

2. Preparation and handoff

1. Auto-translation
2. Transfer to vendor

3. Translation (at vendor)

1. LCX files are translated by vendor in LocStudio, including dialog sizing
2. Validation performed by LocStudio based on information in LCX file, per resource

4. Hand-back & validation

5. Build target files (at MS)

1. Generate localised EXE, DLL, HTML files from original files and LCX

XLIFF and LCX Format Comparison

Methodology

- Literature survey on XLIFF and LCX formats
- Analysis of sample files and tools
- Element by element schema analysis and comparison
- Interviews and further clarifications with Microsoft staff
- Designing of schema mapping between LCX and XLIFF
- Development of XSLTs
- Development of prototype converter
- Consideration of alternative schema mappings

XLIFF and LCX Format Comparison

Results

- **Both file formats have advantages and disadvantages unique to each**
- **Closely related and complementary features**

Similarities

- Both file formats are based on XML with extensibility
- Both intended to abstract from specific resource formats
- Both support meta-data, processing automation

Differences

- XLIFF intended for transport of primarily text data between localisation tools
- LCX designed as storage container for all elements of software localisation resources
- XLIFF supports inline tagging
- LCX treats string elements as CDATA

XLIFF and LCX Format Comparison

XLIFF: Features, Advantages and Disadvantages

- Can represent virtually any file format with localisation resources
- Supports multiple languages in the same file
- Can store other supplementary information such as references to glossaries and workflow meta-data
- Can store alternative translations corresponding to particular localisation item
- There are 8 inline elements defined in XLIFF. These elements are useful to accurately retain the formatting (and other) information.
- Provides a mechanism to specify proper segmentation of translation units

XLIFF and LCX Format Comparison

XLIFF: Features, Advantages and Disadvantages

```
<xliff>
  <file original='helloworld.txt' source-language='en'
target-language='de-DE' datatype='plaintext'>
  ..
  </file>
  <file original='notepad.exe' source-language='en'
target-language='si-LK' datatype='winres'>
  ..
  </file>
</xliff>
```

XLIFF and LCX Format Comparison

XLIFF: Features, Advantages and Disadvantages

```
<header>  
  <reference>  
    <external-file href="segments.srx"/>  
  </reference>  
</header>
```

```
<trans-unit id='hi'>  
  <source>Hello world</source>  
  <target>Bonjour le monde</target>  
  <alt-trans>  
    <target xml:lang='es'>Hola mundo</target>  
  </alt-trans>  
</trans-unit>
```

XLIFF and LCX Format Comparison

XLIFF: Features, Advantages and Disadvantages

```
<trans-unit id= "1">  
  <source>First sentence.Second sentence.</source>  
  <seg-source>  
    <mrk mtype="seg" mid="1">First sentence.</mrk>  
    <mrk mtype="seg" mid="2">Second  
sentence.</mrk>  
  </seg-source>  
  <target>  
    <mrk mtype="seg" mid="1">Translated first  
sentence.</mrk>  
  </target>  
  <alt-trans mid="2" match-quality="75%">  
    <target>The translated second sentence.</target>  
  </alt-trans>  
</trans-unit>
```

XLIFF and LCX Format Comparison

XLIFF: Features, Advantages and Disadvantages

- XLIFF does not provide a powerful mechanism to represent GUI components attached to localisation items.
- In XLIFF, Notes or Comments can not be further categorized.
- In XLIFF same information can be represented in different ways.
 - e.g. grouping of items, segmentation and inline elements
- Absence of a mechanism to control or manage custom extensions
 - Different flavours of XLIFF
- There is no way associate meta-data with external resource references. e.g `<reference type="workflow">`

XLIFF and LCX Format Comparison

XLIFF: Features, Advantages and Disadvantages

```
<xliff>  
  <trans-unit id = '209' resname = '2' restype =  
    'button' style = '0x50030001' coord =  
    '84;44;32;14'>  
    <source>Cancel</source>  
  </trans-unit>  
</xliff>
```

XLIFF and LCX Format Comparison

LCX: Features, Advantages and Disadvantages

- LCX's binary encoded representation scheme provides an accurate, secure and powerful representation mechanism of GUI resources
- LCX format has unique extensibility mechanism through its property bags feature
 - A property bag can store several *key, value* pairs.
 - The *keys* can be defined to hold *values* in different data types including Numbers, Boolean, Stings, and Dates
- Provides a mechanism to embed target content validation rules along with the localisation content
- In LCX format, it is possible to associate comments with most elements
- LCX format has an object model, which defines how to programmatically access and manipulate LCX format content

XLIFF and LCX Format Comparison

LCX: Features, Advantages and Disadvantages

```
<Item ItemId="259" ItemType="#;WIN_DLG_CTRL" PsrId="3"
Leaf="true" >
  <Str Cat="Static Text" >
    <Val><![CDATA[&Encoding:]]></Val>
  </Str>
  <Bin BinId="9">
<Val><![CDATA[AQAAAAAAAAAAAAAAAAAAAAAAAAAA==]]></Val>
  </Bin>
  <Disp Icon="Dlg"/>
</Item>
```

```
<trans-unit id = '206' resname = '259' restype = 'static' style =
'0x50000000' coord = '68;1;40;40'>
  <source>&Encoding:</source>
</trans-unit>
```

XLIFF and LCX Format Comparison

LCX: Features, Advantages and Disadvantages

```
<LCX SchemaVersion="6.0" Name="notepad.exe" PsrId="3" FileType="4"
SrcCul="en-US" TgtCul="fr">
  <Props>
    <Str Name="CustomKey1" Val="Custom Value1"/>
    <Str Name="CustomKey2" Val="Custom Value2"/>
  </Props>
  ...
  <Item ItemId="1" ItemType=";ACCEL" PsrId="3" Leaf="true">
  ..
    <Props>
      <Str Name="CustomKey" Val="Custom Property Value"/>
    </Props>
  </Item>
  ..
</LCX>
```

XLIFF and LCX Format Comparison

LCX: Features, Advantages and Disadvantages

```
<Item ItemId="23" ItemType=";WIN_MENU" PsrId="3" Leaf="true">
..
  <Cmts>
    <Cmt Name="Loc"><![CDATA[This is a Comment]]></Cmt>
  </Cmts>
</Item>
```

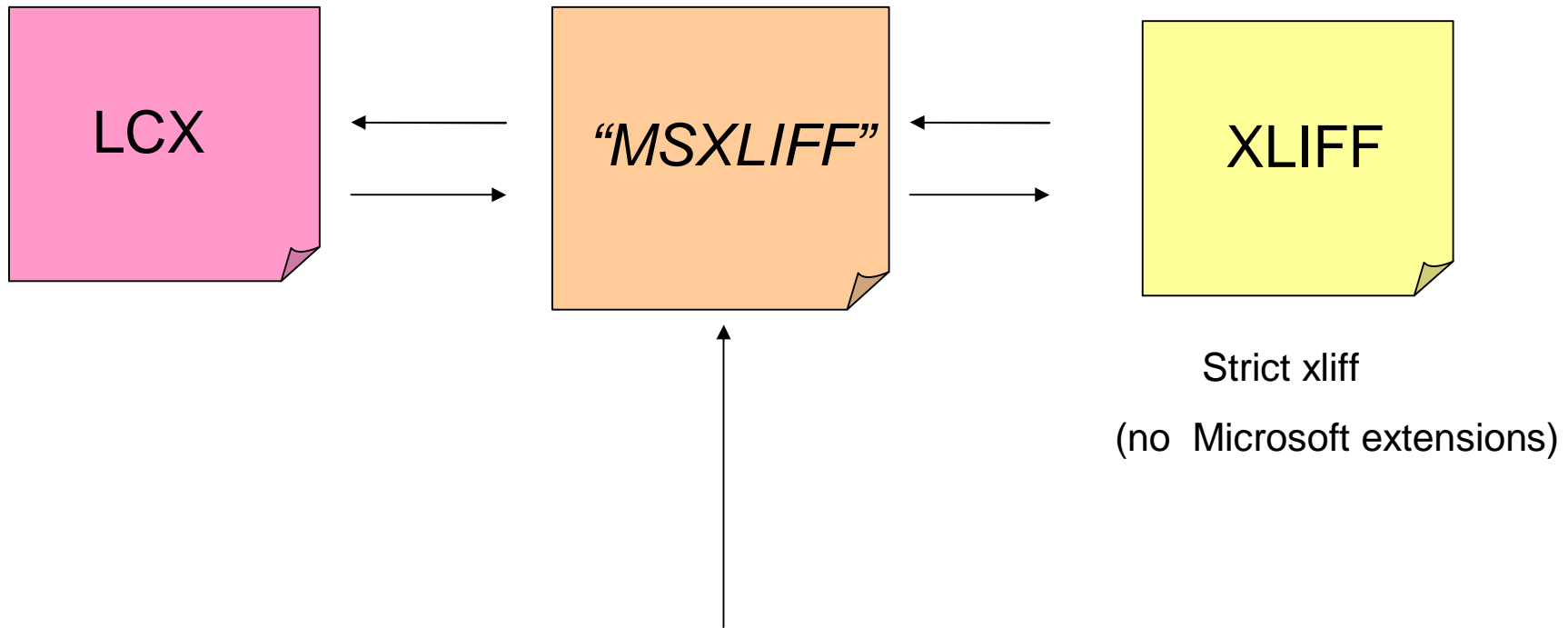
XLIFF and LCX Format Comparison

LCX: Features, Advantages and Disadvantages

- LCX format is strictly bilingual
- There are no specific elements to store alternative translations
- There are no specific elements to store contextual information about localisation items
- Does not provide a mechanism to represent proper segmentation methodology
- In LCX format, there is no mechanism to store references to external files

XLIFF ↔ LCX Conversion

Architecture: MSXLIFF



A valid XLIFF file (Transitional XLIFF schema)
(extended XLIFF to include LCX syntax)

- Schema representation mechanisms:
 - Minimalist Approach:
 - Conversion of text only elements of LCX
 - Maximalist Approach
 - Conversion of both binary and textual elements of LCX
- Prototype to demonstrate the XLIFF ↔ LCX conversion using Minimalist Approach:
 - Complex XSLT Style Sheets
 - Application: VB.NET + Saxon XSLT processor (Saxonica)

XLIFF <-> LCX Conversion

Prototype Converter

```
<Item ItemId="259" ItemType="130;WIN_DLG_CTRL_" PsrId="3"
Leaf="true">
  <Str Cat="Static Text">
    <Val><![CDATA[&Encoding:]]></Val>
  </Str>
  <Bin BinId="9">
    <Val><![CDATA[AQAAAAAggAAAA==]]></Val>
  </Bin>
  <Disp Icon="Dlg"/>
</Item>
```

```
<trans-unit id = '206' resname = '259' restype = 'static' style = '0x50000000'
coord = '68;1;40;40' lcx:ItemId="259" lcx:ItemType="130;WIN_DLG_CTRL_"
lcx:PsrId="3" lcx:Leaf="true">
  <source>&Encoding:</source>
  <lcx:Disp Icon="Dlg"/>
</trans-unit>
```

XLIFF <-> LCX Conversion

Prototype Converter

```
<Item ItemId="259" ItemType="130;WIN_DLG_CTRL_" PsrId="3" Leaf="true">
  <Str Cat="Static Text">
    <Val>&lt;![CDATA[&Encoding:]]&gt;</Val>
  </Str>
  <Bin BinId="9">
    <Val>&lt;![CDATA[AQAAAAAggAAAA==]]&gt;</Val>
  </Bin>
  <Disp Icon="Dlg"/>
</Item>
```

```
<trans-unit id="d0e462" resname="259" lcx:ItemId="259"
lcx:ItemType="130;WIN_DLG_CTRL_" lcx:PsrId="3" lcx:Leaf="true">
  <source>&Encoding:</source>
  <lcx:Str Cat="Static Text">
    <lcx:Val>&lt;![CDATA[&Encoding:]]&gt;</lcx:Val>
  </lcx:Str>
  <lcx:Bin BinId="9">
    <lcx:Val>&lt;![CDATA[AQAAAAAggAAAA==]]&gt;</lcx:Val>
  </lcx:Bin>
  <lcx:Disp Icon="Dlg"/>
</trans-unit>
```

XLIFF and Interoperability Issues

- Leading software industry publishers still prefer using their ***proprietary file formats*** rather than the XLIFF standard
- Open source community still prefer using: ***Portable Object (PO) files***
- The reasons for lack of the adoption of XLIFF standard
 - Lack of awareness about XLIFF standard
 - Limitations and flaws of the standard
 - Improper implementation of the standard
 - Lack of tool support for the XLIFF standard

- **Improper/incomplete feature implementation of the standard:**
 - Many XLIFF features are either not supported or only partially supported by tools
 - Business Case Requirements
 - Depending on the requirements of different business cases, tools have been implemented only to **support different parts of the XLIFF specification**
 - Complexity, Limitations or Flaws of the Standard
 - Due to features that are difficult to implement
 - Due to extreme flexible nature – different flavors of XLIFF
 - Vaguely defined criteria

● **XLIFF Conformance**

- Formal XLIFF compliance is easy
 - Easy to generate XLIFF files that conforms to XLIFF specification.
 - Custom Extensions, Custom Attribute Values
 - Many tools claim XLIFF compliant, however many features are missing
 - XLIFF TC is working on introducing a XLIFF conformance criteria

XLIFF and Interoperability Issues

- Observations w.r.t. XLIFF enabled tools:
 - **Cannot export into XLIFF v1.2 format (especially Win 32)**
 - **Inability to open XLIFF with multiple file elements**
 - Alchemy Catalyst, Passolo
 - **Source language format inconsistencies: en-US, en**

Imhof (2010) gives further examples:

- Support for Alternative Translation Units
 - MemoQ, Trados 2009
- Inline element usage
- Support for translate attribute and translate state attributes
 - Custom value usage
- Support for `<note>` element
 - MemoQ, Trados 2009

–Alchemy Catalyst 8

- Produce Reports in XLIFF Format (Only contain statistics about localisable resources e.g. word count etc)
- XLIFF 1.1

–SDL Passolo 2009

- Win32 resources can be exported by using a macro.
- However, **generated XLIFF files do not conform to XLIFF 1.2** specification.
- Generated XLIFF files** do not contain skeleton information (i.e. binary resources are not embedded with generated XLIFF files)
- XLIFF 1.1

–SDL Trados Studio 2009

- Supports XLIFF 1.2**
- Has its own flavor of XLIFF (SDLXIFF)
- Conforms to XLIFF specification, however it includes proprietary data!
- Does not support Passolo Generated XLIFF files from Win32 resources

- Is XLIFF really an Open Standard?

- Definition of Open Standard

http://en.wikipedia.org/wiki/Open_standard

- There is no agreed upon definition for open standard.
- Definition of 'Open Standard' itself is a research area
 - Open Standards Requirements, Krechmer, K. 2005
 - An Objective Definition of Open Standards, Tiemann, M. 2006
 - Open Standards and the Role of Politics, Shah, C., 2007

- Is XLIFF really an Open Standard?
 - Tiemann (2006) proposes four levels of Open Standards.
 - Open Standard Level 0: **"The standard is documented and can be completely implemented, used, and distributed royalty free"**.
 - Open Standard Level 1: **"There is specified Open Source Software that can interoperate with the standard."**
 - Open Standard Level 2: **"There is an open source software reference implementation of the standard"**
 - Open Standard Level 3: **"The implementation of the standard is an open source software implementation."**

- Is XLIFF really an Open Standard?
 - Tiemann (2006) proposes four levels of Open Standards.
 - Open Standard Level 0: "The standard is documented and can be completely implemented, used, and distributed royalty free".
 - OASIS
 - Open Standard Level 1: "There is specified Open Source Software that can interoperate with the standard."
 - {OLT, Transolution, XLIFF RoundTrip tool, XLIFF Tools}?
 - Open Standard Level 2: "There is an open source software reference implementation of the standard"
 - Open Standard Level 3: "The implementation of the standard is an open source software implementation."

- Is XLIFF really an Open Standard?
 - Open Standard Level 2: **"There is an open source software reference implementation of the standard"**
 - Provides highly robust mechanism to achieve the benefits of the underlying standard
 - Guaranteed exist strategy of a standard implementation
 - Provides the ability to review the actual implementation of the standard
 - Helps to identify the issues associated with the standard
 - Helps the continuous development of the standard

- Interoperability issues between XLIFF and LCX file formats
- Prominent issues associated with XLIFF standard

- Nice to have features in XLIFF
- Nice to have features in LCX

Discussion

Nice to have features in XLIFF

- Binary representation of User Interface (UI) elements.
- Ability to associate `<note>` element with any element and to further customize or categorize `<note>` element.
- A mechanism similar to LCX's 'Property Bag' feature
 - Store meta-data related to each localisable item.
- A mechanism to embed target validation rules
- A zipped XML data container that can include various meta-data (similar to Microsoft DOCX format)
 - Project specific meta-data, Original File, Converted File, Meta-data added by tools
- Ability to further describe internal and external file references
 - e.g. `<reference type="workflow">`

- Inline elements
- Ability to store references to external files/data sources
- Addition of first class elements in the schema to store alternative translations
- Addition of first class elements in the schema to store contextual information related to localisation items
- Ability to specify segmentation
- XML canonicalization (*W3C, 2001*)

Summary

- Studied LCX and XLIFF file formats in depth
- Identified similarities/dissimilarities of two file formats
- Advantages and disadvantages of two file formats
- Optimized conversion process between two file formats
- Issues associated with the conversion

- Identify interoperability issues associated with other localisation standards/tools in various stages of the localisation process
- Investigate XLIFF conformance issues in depth and find solutions