

Panel *Minimal and Modular XLIFF*

Asgeir Frimmannsson

Stefan Pries

Friedel Wolff

Christian Lieske (moderator)

1ST XLIFF INTERNATIONAL SYMPOSIUM

22 September 2010, Limerick, Ireland

The Idea behind the Panel

Observed Need

- Modular and minimal XLIFF (to allow easier implementation)

Choreography

- Speak about panelist's interest in modular and minimal XLIFF
- Discuss straw man proposal for approach, and core features

Participants

- Asgeir Frimannsson, Stefan Pries, Friedel Wolff, Christian Lieske (moderator)

Goal

- Get perspectives, ideas and possibly initiate further work
- Outline for questionnaires on modular and minimal XLIFF

Panel Parts

Introduction (incl. Why Minimal and Modular?)



How to Approach a Solution?



Which Processing Steps (Domains/Layers/Feature Sets) to Consider?



Which Data Category Clusters (Modules) to Consider?

Asgeir Frimmannsson - Bio Sketch

Primarily works on Java-based open source tooling for collaborative localization of e-content

Associated with Queensland University of Technology (Brisbane, Australia), working towards a PhD within localization technology and standards

Senior Software Engineer at Red Hat



Active participant in the development of localization standards

Voting member of the XLIFF TC, presently serving as the schema specification lead for the 2.0 revision of the Oasis XLIFF standard

„XLIFF as it is today is overly complex, ambiguous and not practical to implement. I would like to see a much simpler and modular XLIFF, with clearly defined processing expectations.“

XLIFF as both a container format within an online TMS, and as an exchange format for localisation resources for specific tasks within the localisation process

XLIFF as canonical data model within our localisation process

Build open source tooling to support the workflow around XLIFF

Stefan Pries – Bio Sketch

Working on interoperability between Andrä AG's TMS ONTRAM and other systems based on open standards

Main Focus on Localization Exchange / Translation Memory Exchange / Terminology Exchange

Software engineer for Andrä AG

Implemented translation filters for various file formats (incl. XLIFF)



Reduce challenges for TMS/CAT-Tool providers that have trouble implementing XLIFF completely and correctly

“XLIFF providers should not have to worry about how far advanced the client’s tool’s XLIFF support is”

Translation memory and terminology data in XLIFF

Use of standard environment (e.g. via XLIFF -> docx and docx -> XLIFF transformations)

Friedel Wolff – Bio Sketch

Leads the development of Virtaal and Pootle (both supporting XLIFF) – tools for localisation, localisation management and crowd-sourcing

Trained many localisers across Africa

Developer and localisation engineer at South African localisation company Translate.org.za.

Contributes regularly as a localiser to many Open Source software projects



Professional translators in the developing world who don't use CAT tools (e.g. since they shy away from their complexity)

„An ideal XLIFF standard would help tool developers bridge that gap instead of widening it.“

Easy implementation means a wealth of tools that solve many problems

Optional parts of the specification might give the power to tool developers to hide complexity from less experienced localisers

Christian Lieske - Bio Sketch

Knowledge Architect

Content engineering and process automation (including evaluation, prototyping and piloting)

SAP Language Services
- Globalization Services -
SAP AG



Main fields of interest:
internationalization, translation
approaches and Natural Language
Processing

Contributor to standardization at the
World Wide Web Consortium (W3C),
OASIS and elsewhere

Help content architects, localization managers, and tool developers to understand globalization-related facets (incl. XLIFF)

Combine innovation and investment protection

Start from existing ideas

- XLIFF Technical Committee
- W3C (Internationalization Tag Set (ITS))
- Localization Industry Standards Association
- EU sponsored Thematic Network www.multilingualweb.eu

Draw on Resource Description Framework (and other facets of the Semantic Web)

Question Areas

How do you mainly look at XLIFF?

- Implementor view? Commercial Tools User view? Open Source Tools User view? ...

Why are you interested in modular and minimal XLIFF?

- Current XLIFF too complex to implement? Too little interoperability? ...

Where would you start looking for modularization?

- Processing domains? ...

Where would you start looking for minimalization?

- Most frequently used data categories? ...

How would you see the “framework” for a modular and minimal XLIFF?

- Backwards compatibility as mandatory? New ingredients such as RDF allowed? ...

Are there any standards which could serve as examples of good modularization?

What do you think about the strawmen view on domains, framework and modules/data categories?

Approaches to a min. and mod. XLIFF (1/2)

The why
defines the
how

- Lower threshold for adaptation
- Less steep learning curve
- Decrease efforts for implementation
- Ease use
- Facilitate checking of conformance

The status quo
defines the
how

- What's optional/mandatory?
- Categories (e.g. inlines) define the modules

Some New
Ideas are
allowed

- Make pointer/access to original mandatory (in order to allow use in certain workflow steps (eg. rendering))

Top-Down

- Think about domains
- Think about data categories
- Discuss on macro layer

Bottom-Up

- Think about most frequently used XLIFF constructs
- Discuss on micro layer

Implementation aspects (eg. new schemata) and definition of „minimal“ are a separate story ...

Look at domains

Determine framework

Select data categories

Domains/Feature Sets

Extraction/Filtering (including reverse ie. insertion/merging)

Constraint Setting

- Determine size requirements
- Determine character set requirements

Internationalization

- Mark "not translation relevant"
- Create note

Automated Linguistic Processing

- Segmentation
- Automatic TM Leveraging
- Machine Translation
- Terminology identification/extraction
- Automated linguistic checking (spelling, grammar, style, terminology)

Human Translation

- Creation of target text
- Display, creation or annotation of notes
- Special operations (eg. cloning)

Localization

- Adaptation of viewports
- Adaptation of URLs

Reviewing

- Insertion of annotations
- Display, creation or annotation of notes

Inclusion of reviewing results

Workflow Events

- Status updates

Tool-specific Events

- Trigger rendering

Technical QA checks

- Missing "target" after translation
- Check size requirements
- Check character set requirements

Packaging

Backwards compatible wherever possible?

Cover new requirements (eg. advance leveraging involving MT)?

Introduce new ideas

- Allow access to original, or store copy (similar to skl)?
- Turn to Resource Description Format (RDF)?

Frequently Used Data Categories

Payload (ie. „source“ and possibly pre-filled „target“)

String length constraints (min. or max. length)

Resource type (e.g. different User Interface controls – label, button, ...; overlap with internationalization)

Inlines („ph“, „x“, ...)

Identifiers (for processing)

Names (ie. the identifier used in the native format – key in a Java property file ...)

Notes (e.g. explanations and other annotations; overlap with internationalization)

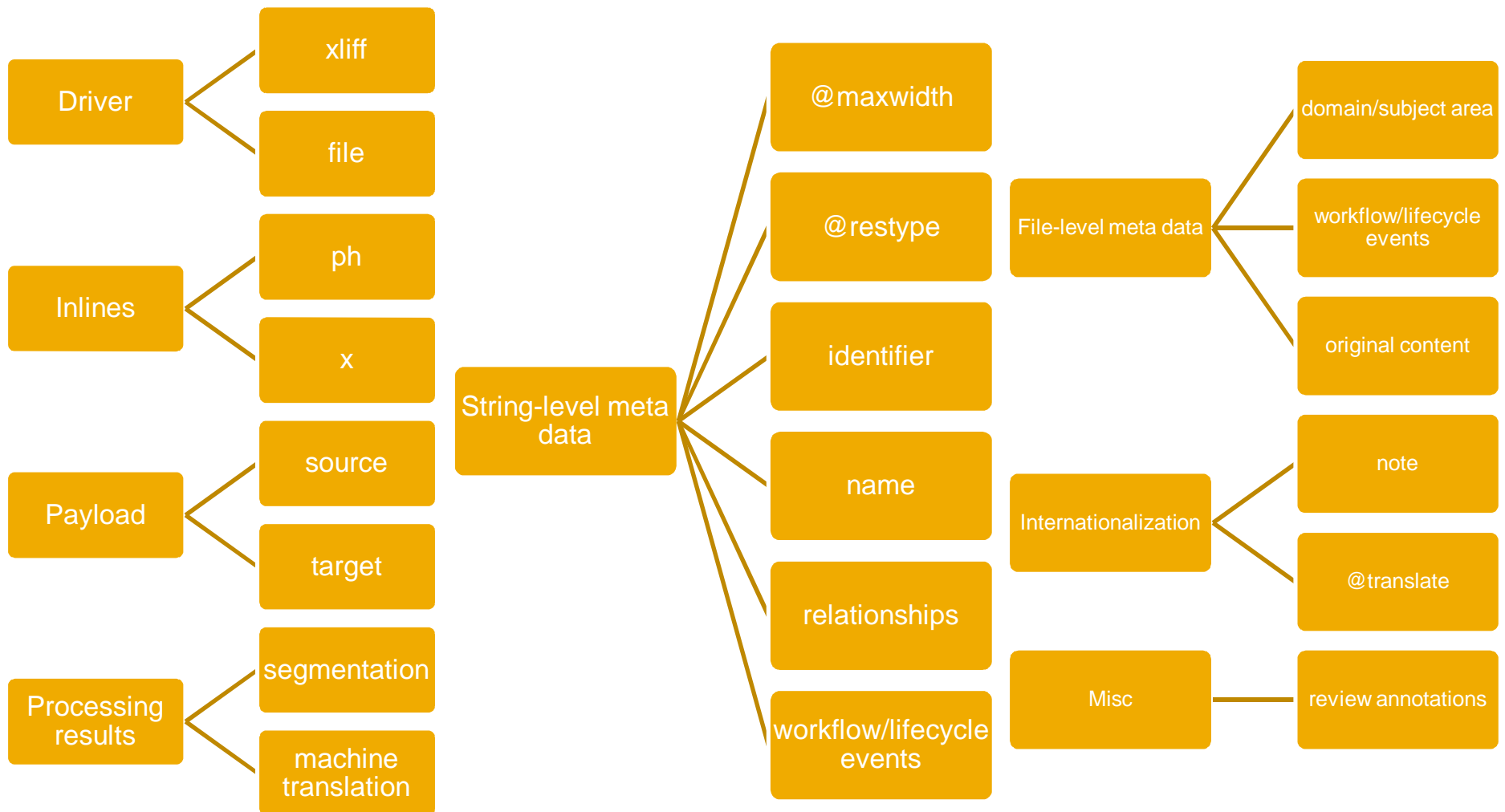
Internationalization (e.g. „translate“)

Domain/subject area

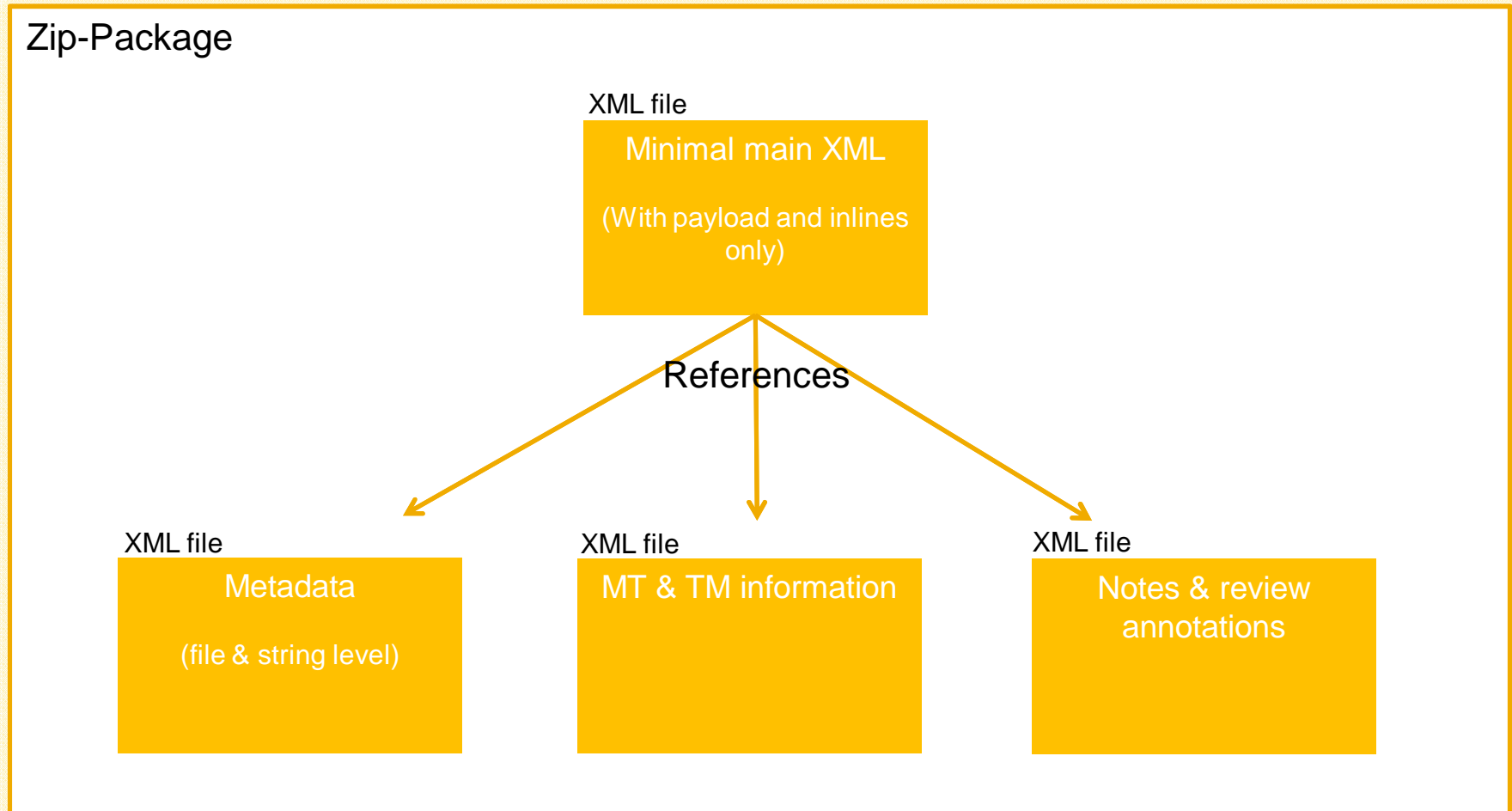
Relationships (e.g. between strings belonging to the same User Interface menu)

Creation (e.g. generator, and creation date)

Modules of Data Categories



Packaging - Archive-/Zip-File Architecture



Packaging - Archive-/Zip-File Architecture

1. Similar to for example Open Document Format (ODF)
2. All files (except for the optional original) would be XML

