Possible Changes to OLIF 2.1

THE BEST-RUN BUSINESSES RUN SAP™

General Issues

Japanese

THE BEST-RUN BUSINESSES RUN SAP™

Overview

The following slides list issues that suggest possible changes to OLIF 2.1.

They are based among other things on experiences at SAP, acrolinx, and the European Patent Office.

Where possible a suggestion and a rationale for the change is given. For most possible changes, screenshots are given which show the code differences (left: old; right: new).

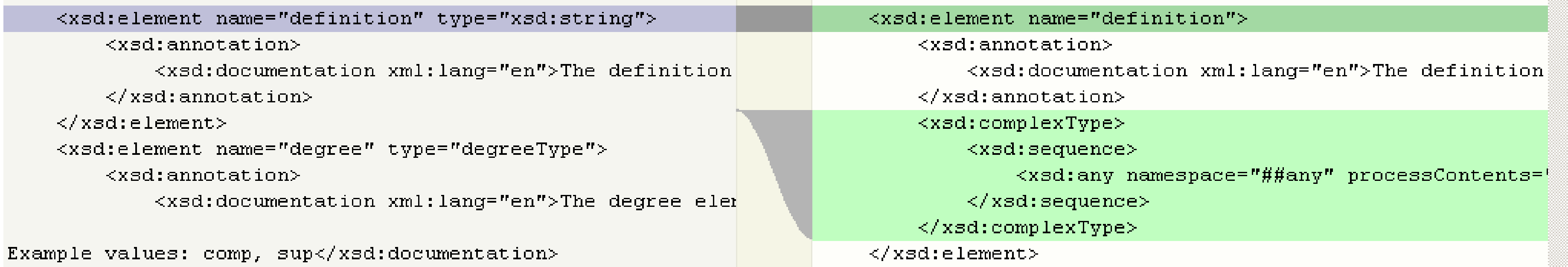
Classes of changes: fixes, alternations, additions

Approach to changes: what?, how?, when?

Content Model for *Definition*

Suggestion: Allow markup

Rationale: Enables use of XHTML and other formats



```
    <xsd:element name="definition" type="xsd:string">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">The definition
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="degree" type="degreeType">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">The degree ele

Example values: comp, sup</xsd:documentation>
```

```
    <xsd:element name="definition">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">The definition
        </xsd:annotation>
        <xsd:complexType>
            <xsd:sequence>
                <xsd:any namespace="##any" processContents="
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
```

Content Model for *Example* (and possibly others like *note*)

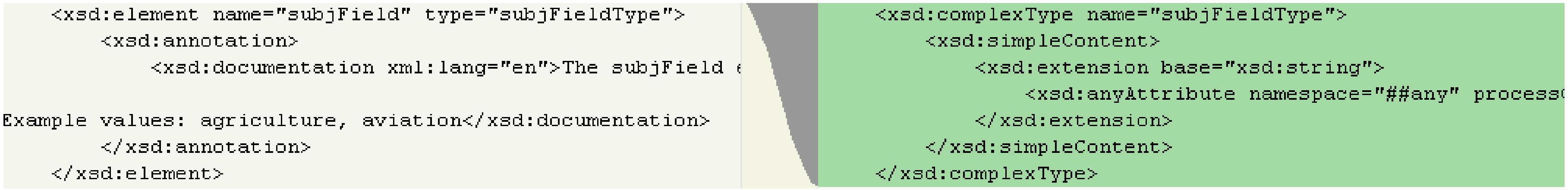
Suggestion: Allow markup

Rationale: Enables use of XHTML and other formats

Attribute for *subjField*

Suggestion: Allow attributes on *subjField*

Rationale: Enables easy use of proprietary information on subject fields (such as additional subject fields, or proprietary values)



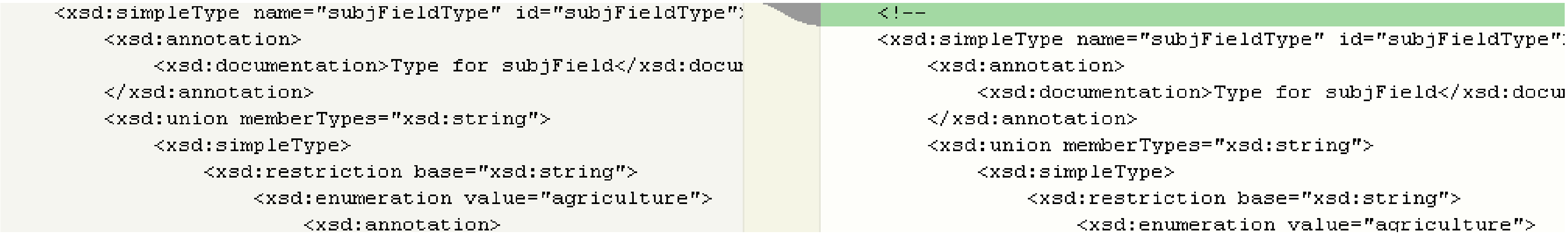
```
    <xsd:element name="subjField" type="subjFieldType">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">The subjField

Example values: agriculture, aviation</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
```

```
    <xsd:complexType name="subjFieldType">
        <xsd:simpleContent>
            <xsd:extension base="xsd:string">
                <xsd:anyAttribute namespace="##any" process
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
```

Content Model for *subjField*

Suggestion: Allow arbitrary strings as values

Rationale: Enables validation (at SAP for example non of the predefined values are used, and thus the existing XSD cannot be used for validation)



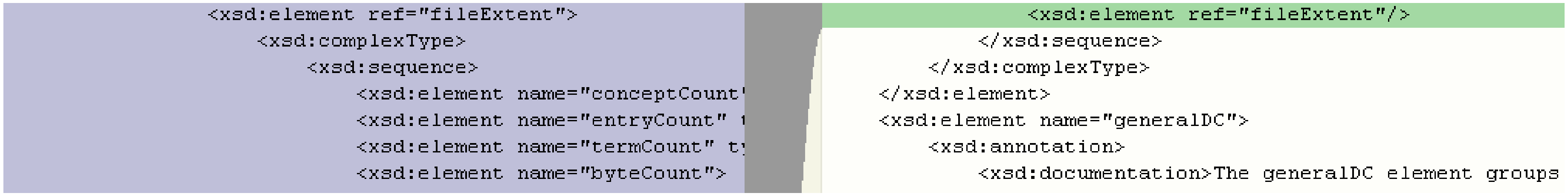
```
    <xsd:simpleType name="subjFieldType" id="subjFieldType">
        <xsd:annotation>
            <xsd:documentation>Type for subjField</xsd:docu
        </xsd:annotation>
        <xsd:union memberTypes="xsd:string">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="agriculture">
                        <xsd:annotation>
```

```
    <!--
    <xsd:simpleType name="subjFieldType" id="subjFieldType":
        <xsd:annotation>
            <xsd:documentation>Type for subjField</xsd:docu
        </xsd:annotation>
        <xsd:union memberTypes="xsd:string">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="agriculture">
```

Fix to *fileExtent*

Suggestion: Change definition of *fileExtent*

Rationale: Existing (buggy) definition disallows use of certain tools (such as XMLSpy)



```
<xsd:element ref="fileExtent">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="conceptCount'
            <xsd:element name="entryCount" t
            <xsd:element name="termCount" ty
            <xsd:element name="byteCount">
```

```
            <xsd:element ref="fileExtent"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="generalDC">
    <xsd:annotation>
        <xsd:documentation>The generalDC element groups
```

Issues from acrolinx

- **Term bank round-tripping: the import of OLIF into our term bank and the subsequent OLIF export from it should preserve all information.**
- **All deprecated terms should be kept (there may be deprecated terms without suggestions; apparently currently not possible)**
- **Term rules (possibly with links to instances ("plain" terms))**
- **Help information**
- **Custom-defined fields**
- **Options settings**

Data which have no meaning outside of acrocheck will be encoded in an "acrolinx:" name space (such as term harvesting settings for instance).

Issues from the European Patent Office

- **Inflection schemes/canonical forms: could we get more precision on this (e.g. regular expressions)**

- **Support for more languages (Italian, Dutch, Romanian, Swedish) => canonical form definitions, inflection schemes, ...**

- **Relational database vs. flat hierarchy**

- **Since we are interested in MT of patents and related texts which are not under our control we are not interested in defining terminology.**
  However we have the problem of specifying the best translation.
  Zug => train, move, trait, ...

  - How many entries do we need ? One or one for each translation?
  - Repetition of grammar information ?

Issues from SDL Trados

Date and time format

Non-mandatory fields

General Issues

Japanese

THE BEST-RUN BUSINESSES RUN SAP™

Integrate Japanese into OLIF

JMdict

- Multilingual Japanese-source dictionary project (targets in English, French, German)
- Extension of EDICT
- Format implemented as XML DTD

Features of JMdict

- Headwords represented by ‘kanji’ and ‘kana’ elements
- Administrative and grammatical information associated with the source
- Target language(s) equivalencies
- UTF 8 Unicode

Sample of JMdict DTD

```
<!DOCTYPE JMdict [
<!ELEMENT JMdict (entry*)>
 <!--                                                          -->
 <!ELEMENT entry (ent_seq, k_ele*, r_ele+, info*, sense+)*>
                <!-- Entries consist of kanji elements, reading elements, general
               information and sense elements. Each entry must have at least one reading
               element and one sense element. Others are optional.
                -->
 <!ELEMENT ent_seq (#PCDATA)>
                <!-- A unique numeric sequence number for each entry
                -->
<!ELEMENT k_ele (keb, ke_inf*, ke_pri*)>
                <!-- The kanji element, or in its absence, the reading element, is the
               defining component of each entry. The overwhelming majority of entries
               will have a single kanji element associated with a word in Japanese. Where
               there are multiple kanji elements within an entry, they will be orthographical
               variants of the same word, either using variations in okurigana, or
               alternative and equivalent kanji. Common "mis-spellings" may be included,
               provided they are associated with appropriate information fields. Synonyms
               are not included; they may be indicated in the cross-reference field
               associated with the sense element.
                -->
```

Sample of JMdict Values

adj	adjective (keiyoushi)
adj-na	adjectival nouns or quasi-adjectives (keiyodoshi)
adj-no	nouns which may take the genitive case particle`no‘
adj-pn	pre-noun adjectival (rentaishi)
adj-t	`taru' adjective
adv	adverb (fukushi)
adv-n	adverbial noun
adv-to	adverb taking the `to' particle
aux	auxiliary
aux-v	auxiliary verb
conj	conjunction
int	interjection (kandoushi)
iv	irregular verb
n	noun (common) (futsuumeishi)
n-adv	adverbial noun (noun, fukushitekimeishi)

Integration into OLIF

- **Map overlapping language-general features/values**
- **Add language-general JMdict features/values to OLIF, e.g., *style***
- **Integrate Japanese-specific features/values via OLIF extensibility options, i.e., XML namespace**