README, 16/12 2014, Dorte Haltrup Hansen, CST

Konvertering af STO-SIMPLE til LMF

med udgangspunkt i filen: nysimpletotal.sgml

Denne README-fil, filen *STO-SIMPLE-LMF-dokumentation.xsl* samt *SIMPLE-documentation-danishfinalny.pdf* udgør dokumentation for konvertering af STO-SIMPLE til LMF. Udgangspunktet er filen: nysimpletotal.sgml (fra 01-12-2005) og LMF**,** ISO-24613:2008, DTD_LMF_REV_16.dtd.

Under vejs har det været nødvendigt at lave manuelle mapninger og at konstruere (redefinere) strukturer. I STO foregår referencen til semantikken (SIMPLE) via syntaksen. I STO-LMF har vi taget den beslutning at *LexicalEntry*, som findes på både morfologisk, syntaktisk og semantisk niveau, er repræsenteret af GMU, MU og Lemma, hvilket betyder at både syntaksen og semantikken refererer tilbage til morfologien hvor alle bøjningsformer og varianter findes.

Den semantiske, leksikalske indgang i STO-SIMPLE-LMF ser i overordnede træk således ud:

```
LexicalEntry
        Feat: Id =GMU, MU
        Lemma
        Sense
                PredicativeRepresentation
                        Predicate (#ref_1)
                        Correspondences (#ref_2)
                SenseRelation

SemanticPredicate (#ref_1)
        SemanticArgument  (#ref_3)

SynSemCorrespondence Correspondences (#ref_2)
        SynSemArgMap  (#ref_3)
```

I STO-SIMPLE-LMF udtrykkes sammenhængen mellem syntaks og semantik altså gennem elementet *Sense.PredicativeRepresentation.correspondences* til strukturen *SynSemCorrespondence.*
SIMPLE-modellen vil ikke blive yderligere forklaret her, der henvises til *SIMPLE-documentation-danishfinalny.pdf*.

Der er vist et eksempel på den leksikalske indgang for ordet "forvalte" længere nede i dokumentet og sidst er LMF-dtd'en indsat.

Hele STO-SIMPLE-LMF-strukturen med dens forhold til SIMPLE-stukturen kan ses i STO-SIMPLE-LMF-dokumentation.xlsl/LMF-isocat-STO.

README, 16/12 2014, Dorte Haltrup Hansen, CST

Enkelte kommentarer til forholdet mellem SIMPLE og STO-SIMPLE-LMF

- Indholdet af strukturen *Sense* i LMF svarer stort set til de oplysninger der er at finde i *SemU* i SIMPLE.
-
- Elementet *Sense.PredicativeRepresentation.correspondences* er obligatorisk i LMF og derfor konstrueret ud fra Sense-id da værdien ikke eksisterer i SIMPLE.

- Strukturen *SemanticPredicate* (peges på fra *LexicalEntry.Sense.PredicativeRepresentation.predicate*) er konstrueret manuelt på baggrund af de predicate-values der fandtes i *Semu'erne* i SIMPLE. Grunden er at der i SIMPLE ikke var korrekt korrespondance mellem *predicate* i *semu* og *Predicate:id*.
  Fx *danse:*
  i SemU er predicate = ARG1hum
  men i Predicate id= PREDdanse_MOV_1

- *SynSemCorrespondence* der forbinder senseId, synuId, subkategoriseringsrammen og argumenter er genereret på baggrund af udtræk fra STO-basen.

- Mapning mellem Sense.id (Semu-id) og LexicalEntry.id (GMU-id) er lavet:
  1) på baggrund af udtræk fra basen (7229 stk.),
  2) automatisk via LMF-syntaks for de GMU-id der ikke var linket til Semu-id i basen (1811 stk.) samt
  3) manuelt for resten (203 stk.) Disse kan identificeres vha. et kommentarfelt:

```
<feat att="comment"
    val="Linked manually to id (GMU id) and morphologicalUnitId (MU id) because of spelling errors or multiple spellings"/>
```

Bemærk at *SemanticPredicate.SemanticArgument* i LMF ***ikke*** indeholder roller (fx Agent) som i SIMPLE

Der findes DUMMIES på 3 niveauer:

- Dummy-sense (2660 stk.)
  LMF kræver at de Sense.id's der peges på (fra fx *semanticRelation*) skal være defineret andet steds i dokumentet. Hvis indgangen (og dermed Sense-id'en) ikke fandtes, er den genereret som en dummy med værdien "DUMMY-SENSE". Alternativet til en DUMMY-SENSE er at den semantiske relation (hvorfra der peges) bliver fjernet. Dummies er skabt allerede i STO-SIMPLE og ikke ved konverteringen til LMF.

- Dummy-synu (1289 stk.)
  Det er ikke alle indgange som har en *PredicativeRepresentation* der også har en tilsvarende kodning i syntaksen. I disse tilfælde er mapningen fra *SynSemCorrespondence.targetSynuId* erstattet af værdien "dummy-SYNU". Alternativt skulle hele *PredicativeRepresentation* slettes dvs. selektionsrestriktionerne der findes via *predicate*, fjernes.

- Dummy-gmu (6 stk.)
  Der er enkelte indgange med semantisk kodning hvor der ikke er en tilsvarende morfologisk indgang. GMU-id'en er her udfyldt af værdien "GMU_DUMMY".

README, 16/12 2014, Dorte Haltrup Hansen, CST

Eksempel ”forvalte”


```
<LexicalEntry>
        <feat att="id" val="GMU_FORVALTE_1"/>
        <feat att="morphologicalUnitId" val="FORVALTE"/>

        <Lemma>
          <FormRepresentation>
                    <feat att="writtenForm" val="forvalte"/>
          </FormRepresentation>
        </Lemma>

          <Sense id="USEM_V_forvalte_REA_1">
                    <SenseExample>
                              <feat att="example"
                                        val="Almindeligvis lader man et pengeinstitut eller et pensionsforsikringsselskab om at forvalte de opsparede
                                        pensionsmidler"/>
                    </SenseExample>
                    <Definition>
                              <feat att="definition"
                                        val="sørge for noget at sker på en planmæssig og ordnet måde, fx udførelsen af et arbejde, ledelsen af en
                                        organisation el. kontrol med pengesager (NDONY)"/>
                    </Definition>
                    <feat att="ontoType" val="RelationalAct"/>
                    <feat att="ontoSuperType" val="Act"/>
                    <feat att="semanticFeature" val="EventType:Process"/>
                    <feat att="classificateur_de_verbe" val="SOCIAL"/>

                    <PredicativeRepresentation predicate="ARG12hum_concrabs" correspondences="SynSemCor_V_forvalte_REA_1">
                              <feat att="typeoflink" val="MASTER"/>
                    </PredicativeRepresentation>

                    <SenseRelation targets="USEM_N_handling_ACT_1">
                              <feat att="weight" val="PROTOTYPICAL"/>
                              <feat att="semanticRelation" val="Isa"/>
                    </SenseRelation>
          </Sense>
</LexicalEntry>


<SemanticPredicate id="ARG12hum_concrabs">
          <SemanticArgument>
           <feat att="arg1" val="Human"/>
           </SemanticArgument>
          <SemanticArgument>
          <feat att="arg2" val="ConcreteEntity"/>
           <feat att="arg2" val="AbstractEntity"/>
          </SemanticArgument>
</SemanticPredicate>


<SynSemCorrespondence id="SynSemCor_V_forvalte_REA_1">
          <feat att="targetSenseId" val="USEM_V_forvalte_REA_1"/>
          <feat att="targetSynuId" val="SYNU_FORVALTE_1"/>
          <feat att="subcategorizationFrame" val="Dv2N"/>
        <SynSemArgMap>
                    <feat att="correspondence" val="arg12"/>
          </SynSemArgMap>
</SynSemCorrespondence>
```

README, 16/12 2014, Dorte Haltrup Hansen, CST

STO-SIMPLE-LMF-dokumentation.xlsl indeholder følgende ark:		
1	Statistics	Optælling af LexicalEntries, dummies, verber, substantiver, adjektiver, predikater og korrespondancer mellem syntaks og semantik
2	LMF-isocat-STO	Redegørelse af hvilke værdier i STO-SIMPLE der er mappet til hvilke værdier i LMF samt isocat-links for features
3	Onto Types	Liste af ontologiske type fra SIMPLE-ontologien brugt i STO-SIMPLE-LMF
4	Onto + ontoSuper types	Liste af ontologiske type samt deres ontologiske supertype fra SIMPLE-ontologien brugt i STO-SIMPLE-LMF. Bemærk at der ikke altid er defineret ontologiske supertyper i indgangene.
5	SIMPLE ontology	SIMPLE ontologien kopieret fra: http://www.ilc.cnr.it/clips/Ontology.htm
6	Semantic features	Liste af semantiske features brugt i STO-SIMPLE-LMF. Da LMF ikke tillader indlejrede strukturer for disse, men kun frature -values, der værdierne lidt kunstige fx Age:Adult. Alternativt skulle alle venstredele før kolon (fx Age) være udtrykt som fratures og højresiden som value; mne derved ville oplysningen om at der er tale om "semantic features" vs. "semantic relations" gå tabt.
7	Semantic relations	Liste af alle semantiske relationer brugt i STO-SIMPLE-LMF
8	Domains	Liste af alle domæner brugt i STO-SIMPLE-LMF
9	Predicates	Liste af alle prædikatnavne brugt i STO-SIMPLE-LMF
10	Type of links	3 forskellige type links mellem sense og predicate
11	Weight	2 forskellig vægtning (for semantisk relation)

README, 16/12 2014, Dorte Haltrup Hansen, CST

```
<?xml version='1.0' encoding="UTF-8"?>
<!--*****************************************************************************
*                                    LMF DTD
                Strict dtd for validation of DK-SIMPLE-LMF
*****************************************************************************-->

<!-- ********************************  Core package ***********-->

<!ELEMENT LexicalResource (feat*, GlobalInformation, Lexicon+)>
<!ATTLIST LexicalResource
   dtdVersion CDATA     #FIXED "16"
   xmlns:dcr  CDATA     #FIXED "http://www.isocat.org/ns/dcr">

<!ELEMENT GlobalInformation (feat*)>
<!ELEMENT Lexicon (feat*, LexicalEntry+, SemanticPredicate*, SynSemCorrespondence* )>

<!-- ********************************** LexicalEntry ******************************-->
<!ELEMENT LexicalEntry (feat*, Lemma, Sense* )>
<!ATTLIST LexicalEntry
   id                          ID #IMPLIED>
<!ELEMENT Lemma (feat*, FormRepresentation*)>
<!ELEMENT FormRepresentation (feat*)>

<!-- ********************************** Sense ****************************-->
<!ELEMENT Sense (SenseExample*, Definition*, feat*, PredicativeRepresentation*, SenseRelation*)>
<!ATTLIST Sense
   id                  ID #IMPLIED>
<!ELEMENT SenseExample (feat*)>
<!ATTLIST SenseExample
   id              ID #IMPLIED>   <!-- Vi har ikke id på vores eksempler -->
<!ELEMENT Definition (feat*)>
<!ELEMENT PredicativeRepresentation (feat*)>
<!ATTLIST PredicativeRepresentation
  predicate            IDREF #REQUIRED
  correspondences IDREFS #REQUIRED>
<!ELEMENT SenseRelation (feat*)>
<!ATTLIST SenseRelation
   targets  IDREFS #REQUIRED>
<!-- ********************************** SemanticPredicate ******************************-->
<!ELEMENT SemanticPredicate (feat*, Definition*, SemanticArgument*, PredicateRelation*)>
<!ATTLIST SemanticPredicate
   id                  ID #REQUIRED
   semanticTypes IDREFS #IMPLIED>  <!-- Vi har ikke semanticType -->
<!ELEMENT SemanticArgument (feat*)>
<!ATTLIST SemanticArgument
   id                  ID #IMPLIED
   semanticTypes IDREFS #IMPLIED> <!-- Vi har ikke id eller semanticType -->
<!-- ********************************** SynSemCorrespondence ******************************-->
<!ELEMENT SynSemCorrespondence (feat*,SynSemArgMap*)>
<!ATTLIST SynSemCorrespondence
   id              ID #REQUIRED>
<!ELEMENT SynSemArgMap (feat*)>
<!-- ********************************  for data category adornment: feat stands for feature ***********-->
<!ELEMENT feat EMPTY>
          <!-- att=constant to be taken from the DataCategoryRegistry -->
          <!-- val=free string or constant to be taken from the DCR-->
<!ATTLIST feat
   att (argADJ|arg0|arg1|arg2|arg2AO|arg2E|arg2EAO|arg2P|arg2PAO|argASSOC|argP2|AS|ASSOC|
     classificateur_de_adjectif|classificateur_de_nom|classificateur_de_verbe|
     comment|correspondence|subcategorizationFrame|domain|id|languageCoding|languageIdentifier|
     morphologicalUnitId|semanticFeature|semanticRelation|
     definition|example|ontoSuperType|ontoType|targetSenseId|targetSynuId|typeoflink|weight|writtenForm)          #REQUIRED
   val      CDATA #REQUIRED
   dcr:valueDatcat CDATA #IMPLIED
   dcr:datcat CDATA #IMPLIED
   >
```