**<D2.1.1 Documentation of the Corpora>**

ModelWriter

Text & Model-Synchronized Document Engineering Platform

Work Package: WP2

Task: T2.1 – Data Collection

Edited by:

Claire Gardent <claire.gardent@loria.fr> (CNRS/LORIA)

Name Surname <email@company.com> (organization)

…

Date: 30-August-2015

Version: 1.0.0

Apart from the deliverables which are defined as public information in the Project Cooperation Agreement (PCA), unless otherwise specified by the consortium, this document will be treated as strictly confidential.

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author(s) | Date | Remarks |
| 0.1.0 | Claire Gardent | 30-Aug-2015 | Draft |
| 1.0.0 | <name> | <date> | Initial Release |
|  |  |  |  |

Table of Contents

* 1. Role of the deliverable

This deliverable documents the data used to train, develop and text the NLP components (Semantic Annotator, Semantic Parser, Natural Language Generator) of ModelWriter. It might be updated during the project in case additional data is worked with.

* 1. Structure of the document

This document is organized as follows:

Section 1 introduces the document.

Section 2 describes for each use case: the scope and motivation, the approach and the available resources (corpora).

* 1. Terms, abbreviations and definitions

|  |  |
| --- | --- |
| Abbreviation | Definition |
| NLG | Natural Language Generation |
| NLP | Natural Language Processing |
| RDF | Resource Description Framework |
| WP | Work Package |
| UC | Use Case |

1. Corpora

The development and the evaluation of natural language processing systems required data: for training, for tuning and for testing. In the ModelWriter project, this includes textual data, knowledge data and ideally bi-texts i.e., aligned corpora of text and their corresponding knowledge representation.

Based on the use cases identified in WP1, we collected data to develop and evaluate three NLP tools necessary to achieve ModelWriter goals namely, a semantic annotator, a semantic parser and a natural language generator.

The *semantic annotator* is required to synchronised text and models. Its function is to annotate text with elements of the model whereby text elements may differ from model elements with respect to derivational (warn/Warning) or inflectional (pipe/Pipes) morphology , synonymy (pipe/tube) and/or syntax (procedure should be removed/procedureDeletion).

This data will then be used to identify tproche linguistic requirements set by the use cases; to train and test the semantic processors (parser and generator); and to acquire the language models useful for disambiguation (parsing) and fluency ranking (generation).

A *semantic parser* converts text into model representations. It can be used to extend the model (by adding to the current model the model expression representing the meaning of the parsed text) or to synchronise complex natural language expression with one or more model elements.

Conversely, a natural language generator maps model representations to text. It can be used to update a text which is synchronised with a model whenever this model is modified/extended.

2.1 Airbus Data

Use cases UC-FR4 and UC-FR5 target the synchronisation of Airbus SIDP (e.g. System Installation Design Principles) documents with an RDFS model.

The overall driving need for these two use cases is to reduce the time and the burden for the designers to consult a large set of regulation documents in order to retrieve design rules. Due to reasons such as technology push, process changes, etc., an increasing number of different regulation documents are issued by different stakeholders. They contain a high number of informal rules and the designers have difficulties following the information cascade and retrieving or rebuilding the correct information. This situation results in time waste, suboptimal designs and higher risks of error. In ModelWriter, our ultimte goal is to remedy this shortcoming by providing a synchronization mechanism between these documents and a model encoding the rules contained in these documents This is an ambitious goal which in effect, requires building a semantic parser and a generator that can map arbitrary text into formal rules and vice versa. To achieve these goals, we started by gathering the following data.

* SIDP document SIDP 92A001V. This SIDP document contains the system installation design principles applicable to the electrical and optical system installation. It provides an example of how design rules are formulated in SIDP documents and of how these documents are structured. The SIDP document SIDP 92A001 includes text and graphics and contains 6311 word forms. It is available in French-Consortium/airbus/text/SIDP92A001V.docx
* Semi-Structured SIDP rules. The Airbus System Installation team has built an SQL database of SIDP rules which encodes installation rules in a semi-structured format.In effect these rules provide a simplified, semi-normalised version of the rules contained in the SIDP documents thereby facilitating natural language processing (less diversity in the syntactic structures and lexicon, less ambiguity, rules formulated as one sentence rather than across several sentences, fewer anaphoric references etc). We therefore gather these rules to develop a first version of the NLP tools (semantic annotator, semantic parser and text generator) that works on these semi-structured rules. Currently, the semi-structured rules available to the French consortium consists of 986 rules and 13178 word forms. These rules are available in two formats: an excel file whose columns are used to label each part of the rule (French-Consortium/airbus/text/rules.xls) and a text file where this labelling is ignored. (French-Consortium/airbus/text/rules.txt). The excel file is used to automatically construct an RDFS version of the rules while the text file is used to test the NLP tools.
* RDFS knowledge base for Semi-Structured SIDP Rules. To support querying and construct a parallel data/text corpus, AIRBUS specified and implemented a method for automatically converting the semi-structured SIDP rules into an RDFS knowledge base. The resulting parallel data-text corpus is available in French-Consortium/airbus/kb/@ANNE: please add KB derived by your system in that github repository and insert its name here.
* Domain Model. Semantic Parsing maps text to meaning representations which can then be queried and synchronisation links text and model elements via semantic annotation To support both KB querying and synchronisation, Airbus manually developped a knowledge base modeling the domain of SIDP92A001V namely the domain of the electrical and optical system installation. . Currently, this knowledge base includes XX @Anne: please fill in concepts and XX @Anne: please fill in relations. It is available at @Anne: please upload KB in MW githu and replace this placeholder with the path of the file.

Because of confidentiality issues, the Airbus data could only be shared after a Non Disclosure Agreement was signed by all interested parties namely, all French partners. This agreement was finalised on June 1st and access to the data was given shortly thereafter. Currently, this data includes:

During the first year of the project, we used this data as follows.

* The SIDP semi-structured rules were processed by CNRS/LORIA and by AIRBUS to automatically construct an RDFS knowledge base encoding the content of these rules.
* The domain specific KB was used by CNRS/LORIA for the semantic annotation of the SIDP rules. More generally, the current version of the semantic annotator can annotate arbitrary text with concepts from the domain specific KB developed by Airbus.
* The domain specific KB is also used to support SPARQL queries on the RDFS knowledge base automatically derived from the SIDP semi structured rules by allowing for e.g., subclass information to be taken into account. Suppose for instance that the KB includes the knowledge that hose pipes, electrical pipes and water pipes are all pipes, then a query asking for all SIDP rules involving a pipe will return rules involving not only pipes but also all rules involving hose pipes, electrical pipes and water pipes.

For the second year of the projet, the aim is twofold.

First, we plan to use the semantic annotator to annotate arbitrary text with KB concepts. The resulting annotated text will then be used as a basis to develop a semantic parser and a generator.

Second, we will investigate whether the parallel data-text corpus build for the SIDP semi structured rules can be used to train/develop a semantic parser capable of mapping SIDP rules contained in SIDP documents to RDFS models.

2.2 Obeo Data

To be filled by Obeo/Samuel

2.3 Turkish Data

To be filled by Mantis

Annex 1

The following summarizes the corpora (text, data and knowledge bases) developed and used by the ModelWriter project.

A1.1 Airbus Data

Example SIDP Document

<https://github.com/ModelWriter/French-Consortium/airbus/text/SIDP92A001V.docx>

Semi-structured design rules:

<https://github.com/ModelWriter/French-Consortium/airbus/text/rules.txt>

<https://github.com/ModelWriter/French-Consortium/airbus/text/rules.xsl>

Domain Model (RDFS Knowledge Base modelling plane components)

<https://github.com/ModelWriter/French-Consortium/airbus/kb/airbusComponentsKB_03072015.rdf>

RDF Knowledge Base derived from Semi-Structured Design Rules

[https://github.com/ModelWriter/French-Consortium/airbus/kb/rules.rdf](https://github.com/French-Consortium/airbus/kb/rules.rdf)

@Anne: please upload the RDF KB derived by your stagiaires as rules.rdf in the repository listed just above this comment.

A1.1 Obeo Data

To be filled by Obeo/Samuel

A1.1 Turkish Data

To be filled by Mantis

1. Appendixes
2. Appendix 1

Doloreptium dic temquo qui voluptate dellabo. Ut labo. Et pel maxim resed molore nit andios volorumenis eum enihiti nciasim olorepeles ea aut maximolupta vendae sundites dolecaborem ni nonseque poreri dolora plati quid ut lab iuscia volorio rporemp edisitatis sed quis aut explit, to cuptas sendae volor ad moloreium dollat lande iduci dolupta eribus etur, sintem quae videbit estiore ommosapel ea delia volesciustio quiam, sit evero blabore