TERMINAE User Manual[1]- V14-3

Sylvie Szulman (Paris 13)

2014 December

[1] A big thanks to Eve Paul for its review and its suggestions for improving Terminae

Contents

1 Introduction 1

2 The Terminae method 2

3 Technical Characteristics 3

3.1 Installation . . . 3

3.2 How to start . . . 3

3.2.1 Project location and structure . . . 3

3.2.2 How to import a project . . . 4

3.2.3 How to create a project . . . 5

3.3 Hidden files . . . 5

4 Main menu 6

5 Project management perspective 8

5.1 `Project actions` menu . . . 8

6 Terminological level (step 1) perspective 11

6.1 Term extractor uses . . . 11

6.1.1 TermoStat web service . . . 11

6.1.2 YaTeA tool . . . 11

6.2 Data: Terminological files . . . 12

6.2.1 TermoStat Term files . . . 12

6.2.2 Yatea Term files . . . 13

6.2.3 Named entity files . . . 13

6.2.4 Term list files . . . 14

6.3 Perspective overview . . . 14

6.4 `Linguistic actions` menu . . . 15

6.4.1 `File` submenu . . . 15

6.4.2 `Display` submenu . . . 16

6.4.3 `Cleaning` submenu . . . 17

6.4.4 `Term Management` submenu . . . 18

6.4.5 To terminological step submenu . . . 19

6.5 Occurrence view - Popup menu . . . 19

7 Terminological level (step 2) perspective 21

7.1 Perspective overview . . . 21

7.2 Data: Terminological forms . . . 22

7.3 `Terminological actions` menu . . . 23

7.3.1 File management submenu 24
7.3.2 Display submenu 24
7.3.3 Collaboration submenu 24
7.3.4 Form management submenu 25
7.3.5 Feature management submenu 26
7.3.6 To Terminological level (step 1) action 27
7.3.7 Termino-concept management submenu 28

8 TerminoConceptual level perspective 29
8.1 Perspective overview 29
8.2 Data: Termino-conceptual forms 30
8.3 TerminoConceptual actions menu 31
8.3.1 File submenu 31
8.3.2 Display submenu 32
8.3.3 Thesaurus entity management submenu 32
8.3.4 Feature management submenu 34
8.3.5 Modify author 34
8.3.6 Neon ontology submenu 34

9 Neon toolkit Conceptual level (OWL) perspective 37
9.1 Perspective overview 37
9.2 Terminae links menu 37

10 Annotator perspective 40
10.1 Input files 40
10.2 How to proceed 41
10.3 Some caveats 42

11 Collaboration perspective 43

12 Annex 44
12.1 XML backup DTD for terms 44
12.2 XML backup DTD for ENs 44
12.3 EnsLexUnit DTD 45
12.4 Terminological from DTD 46
12.5 DTD export set of forms to skos 47
12.6 Thesaurus DTD 48
12.7 TreeTagger English Tagset 48
12.8 TreeTagger French Tagset 49
12.9 Use ANNIE to extract named entities 50
12.10Gate named entity type file 51
12.11Key binding 52
12.12DTD configuration file 52

Chapter 1

Introduction

This document is the user guide of TERMINAE platform. TERMINAE is a method and a platform which supports the method.

It can be used by terminologists to build terminological forms and by knowledge engineers to build either thesauri expressed in SKOS or ontologies expressed in OWL organising concepts and lexical units in a formal way supporting inferences. It can be also used to link textual elements to semantic resources already défined.

The platform assists users in the design of termino-ontological resources from texts. The corpus may contain one or several documents. The supported languages are English and French.

The user guide describes the functionalities of the TERMINAE platform. Chapter 2 gives a very short insight of the methodology. Chapter 3 gives the technical characteristics and the installation instructions. Chapter 4 presents the main menu and the following chapters (chapters 5 to 10) introduces the 6 perspectives of the platform and the related functionalities.

Chapter 2

The Terminae method

TERMINAE is a tool that is supported by a method, and some (very short) forewords on the method can help using the tool. The task is to build a domain termino-ontological resource (thesaurus or ontology). This is an expert task, since it needs to decide which concepts are really important for the domain, and how they are related. It has been experienced that linguistic tools, relying on texts specific of the domain, can help the expert. They do not do the work in his/her place, but they propose a good starting point to improve the coverage of the domain, and some ambiguities they raise reveal real and unseen ambiguities of the domain vocabulary.

The TERMINAE method starts from the linguistic results produced by a term extractor.
It has then three steps.

- At the linguistic level, the input is a list of **term candidates**, i.e. words or group of words which, on a linguistic basis, could possibly figure in a **terminology** of the domain (a list of its main terms). The goal of this level is in a first step (chapter 6) to constitute, clean and improve the list, removing parasistic or irrelevant proposals. A second step (chapter 7) involves grouping those which are morphologic variants of the same term and collecting linguistic relations. This work relies on the list of occurrences of each term, which are gathered with linguistic information in **terminological forms**. A terminological form is identified by a term.

- The **termino-conceptual** level (chapter 8) is specific to TERMINAE. Whereas terms are at the vocabulary level, the goal is now to analyse the use of terms in the corpus at the semantic level. The work is to recognize and distribute the various senses of this term into several **termino-concepts**, distributing also the occurrences of the term between senses. At the same time, the termino-concepts of the terminological form can be tagged as having a synonym in an other terminological form, or being otherwise (more loosely) related.

- The **ontological** level (see chapter 9) now relies on termino-concepts and their relations to build the ontology. Synonym termino-concepts should only yield one concept. All the related termino-concepts help building the hierarchical relations and defining the roles, as can do some other linguistic information gathered during the process.

Chapter 3

Technical Characteristics

Terminae platform is an eclipse application. It is built as an eclipse RCP application (version 3.7).

- The current version of TERMINAE platform is compiled using Oracle 1.8 Java virtual machine.
- It relies on UTF-8 text encoding.
- It can be used for English and French.
- **It is forbidden to have file names with space characters or accented letters**.

3.1 Installation

To install TERMINAE, you need java,**version 1.8**. Download the version of the platform for your system from the web page and unzip the downloaded file.

`http://lipn.univ-paris13.fr/terminae/index.php/Download`

The default language is English but it can be changed. If you want to work with a French platform, edit the `terminae.ini` file and change the line `nl en` by `-nl fr_FR`.

This file is located in the `Terminae` directory on Linux and Windows systems and in the `Terminae.app/Contents/MacOS` directory on MacOS systems.

3.2 How to start

To launch the TERMINAE platform, click on the `Terminae` application (`Terminae` on Linux system, `Terminae.exe` on Windows system, `Terminae.app` on MacOS).

Initially, the project management perspective (`Terminae Project perspective`) is open and you have to import or create a project.

3.2.1 Project location and structure

In any case, you have to define your project directory. On Linux and Windows systems, it is advised to locate it in the `workspace` directory created by the eclipse application.

A project has a fixed structure, represented as the 6 following subdirectories:

- `corpora`: Contains the corpus data (raw and tagged) and the results of named entity recognition tools. The current version of the platform is designed to work with TreeTagger[1] and ANNIE named entity recognition tool[2].
- `linguae`: Contains the search patterns that have been designed and their results (no pattern design tool is available in the current version).
- `repExtractTerm`: Contains the results of term extraction tools. The current version of the platform is designed to work with:
  - YaTeA term extractor[3]
  - TermoStat term extractor which can be used through a web service or with a sample file involving terms (one term by line). [4].
  - a term list (one lemma by line) with the corpus and its tagged corpus (with TreeTagger).
- `system`: Contains some files automatically created by TERMINAE.
- `terminoFormDir`: Contains the terminological forms that are created using TERMINAE and output by it (format see 12.4).
- `thesauri`: Contains the termino-conceptual resources that are created using TERMINAE (format see 12.6).

3.2.2 How to import a project

A project to be imported is represented as a zipped file containing the project directory with all the required subdirectories and files of a given project. **You do not have to unzip the file**.

- Go to the main menu
- Click on `Project actions`
- Click on `Project management`
- Click on `Import project`
- A first dialog window appears in which you must indicate the zipped file to load.
- A second dialogue window appears, to propose the directory into which the project will be imported. If you do not accept, you'll be offered to choose another one.

When the project is imported, its main characteristics are presented in the `Project information` view on the left (by default) of the `Project perspective` and you can start working on it.

[1] http://www.ims.uni-stuttgart.fr/projekte/corplex/TreeTagger/

[2] http://gate.ac.uk/ie/annie.html

[3] http://search.cpan.org/%7Ethhamon/Lingua-YaTeA-0.621/

[4] http://olst.ling.umontreal.ca/∼drouinp/termostat_web/

3.2.3 How to create a project

To start working on a new project:

- Go to the main menu
- Click on `Project actions`
- Click on `Project management`
- Click on `Create project`
- A first dialog window appears, in which you must indicate the name of the project.
- A second dialogue window appears, in which you must indicate in which directory you want to locate the project. A directory with the same name as the project is automatically created with 6 subdirectories.

To start working on your project to build termino-ontological resource from a given corpus, you need to have at least the following files in your project directory (more details in 6.2):

- In the `corpora` subdirectory:
  - The *raw corpus (.txt)*
  - A tagged version of the *raw corpus (.txt)* (.tt file as output by TreeTagger). The extension may be *.tt* or *.ttfr* or *.TT* or *.TTFR*.
- In the `repExtractTerm` subdirectory: the list of terms that have been extracted from the tagged version of the corpus by YaTeA (.xml file) or the list of terms extracted by TermoStat downloaded from the web service named termostat_res.txt or a list of lemmatized terms (one by line) in a file (.txt).

You must also give the name of the corpus if you exploit one and the name(s) of the authors(s) of the future resource(s).

When the project is created, its main characteristics are presented in the `Project information` view on the left (by default) of the project perspective and you can start working on it.

3.3 Hidden files

The software creates 2 hidden files to manage the Terminae application:

- The file `.Terminae` contains the name of the current project. It is created in the directory where you launch the Terminae application. You normaly do not need to modify it.
- The file `.nameOfProject.xcfg` defines the configuration of each project (the set of files exploited by the project). Advertised user may easily understand its content, and may happen to change it in tricky cases (e.g. for renaming directories or files), (see Annex 12.12).

These files are text files or modifiable xml files.

Chapter 4

Main menu

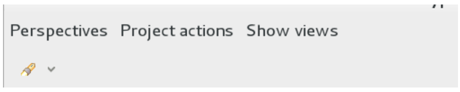


Figure 4.1: Main menuBar

The main menubar (see figure **??**) of the TERMINAE platform presents 3 menus which are associated to specific actions or submenus[1]:

- `Perspectives` menu allows to open other perspectives: you simply have to click on the name of the perspective you want to open in the perspective list that appears. 8 perspectives are accessible:
  - `Project perspective`, which is the default perspective opened when a project is loaded. It is presented in Chapter 5.
  - `Terminological level (step 1)` perspective (see Chapter 6).
  - `Terminological level (step 2)` perspective (see Chapter 7).
  - `Termino-Conceptual level` perspective (see Chapter 8).
  - `Neon toolkit Conceptual level (OWL)` perspective (see Chapter 9).
  - `Annotator perspective` (see Chapter 10).
  - `Collaboration perspective` (see Chapter 11).
  - `TMX perspective` This perspective is used only for Legilocal project. It permets to work with the Temis term extractor.

  The 1,2,3,4 perspectives make up Terminae. OWL perpective belongs to Neon ToolKit 2.4. Annotator perspective marks the occurrences of given terms in a text with concepts and individuals of an ontology. Collaboration perspective allows to compare two Terminae project.

- The action menu gives access to the specific functionalities accessible at the Terminae level where you are currently working. The name of the action menu depends on the current perspective: `Project actions`, `Linguistic actions`, `Terminological actions`, `TerminoConceptual actions` and `Terminae links`.

[1] This main menu slightly differe from on exploitation system to another.

- An additional `Terminae` menu is proposed on MacOS systems. It gives access to the standard application main operations: information (`About Terminae`), `Preferences`, `Hide Terminae`, `Quit Terminae`.

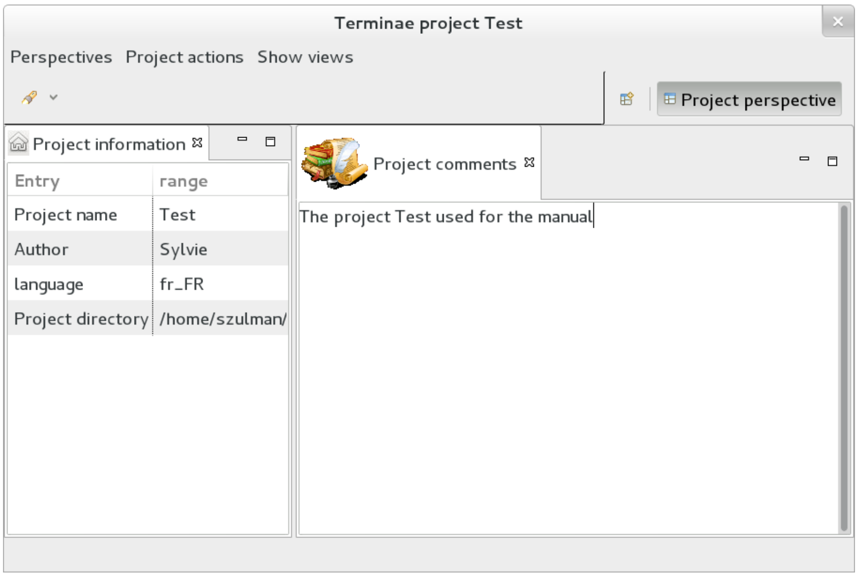
- `Show View` menu is proposed in all perspectives: Each perspective has many views and a main view which is on the left side of the perspective. A click on an item in the main view change values in other views. These views may be closed by the user or he/she may want to see a view of another perspective which is not in the used perspective (only one perspective could be selected).
  This menu is used to reopen a view that has previously been closed. Click on the single item (`Other...`) to find TERMINAE views. Select the view you want to reopen or to see - and be aware that the view may be dependant of one or the other perspective.

In the Terminae toolbar, you can run searches on files.
Below the Terminae menubar,see all your opened perspectives and click on the one you want see.

Chapter 5

Project management perspective

TERMINAE starts with the project management perspective. This perspective has 2 views (Fig. 5.1):



Entry	range
Project name	Test
Author	Sylvie
language	fr_FR
Project directory	/home/szulman/

The project Test used for the manual

Figure 5.1: Project management perspective

- The left view presents the project information if a project has been already defined: project, corpus, thesaurus and author(s)' names.

- The right view is a text editor where the user may write comments. To save the comments, you have to click on the right click of the mouse **`ctrl+s`**.

5.1 `Project actions` menu

A project consists of all data used or created by TERMINAE when building a specific termino-ontological resource from a given corpus (see Section 3.2.1 for a description of the project structure).
The corpus is in a txt file (it is advised to use utf-8 encoding). See section 6.2 to have the description of the used files.

You have either:

Project management submenu

- Create a new project (`Create Terminae project`) if you start to build a specific termino-ontological resource from a given corpus. You have to specify:
  - The name of your project.
  - The name of the directory where you want to locate your project. A default directory is proposed but click on the `cancel` button and navigate through the file system if you want to choose another directory.
  - If your project involves a corpus.
  - The language of the result semantic resource(s) for which the project is created. You have to choose between 5 languages: fr_FR (French), de (German), en (English), es (Spanish), it (Italian).
- Switch from one project to another (`Load Terminae project`, note that only one project can be opened at the same time). You are first offered to navigate through the file system to select the directory containing the concerned project directory.
- Export the current project (`Export project`). A zipped file is created in which all the required directories and files are included. If you have created a Neon project, its directory is also included in the zipped file.
- Import an existing project (`Import project`). The project to be imported is represented as a zipped file containing the project directory with all the required subdirectories and files. You do not have to unzip the file but you have to specify:
  - The zipped file to load.
  - The name of the directory where you want the project to be imported.

Project information management

* `Modify author` allows to modify the project's author.

* `Modify corpus language` defines the corpus language (french(fr_FR), German(de), English (en), Spanish (es), Italian (it)). By default, it is initialized by the platform used language.

* `Modify thesaurus name` allows to modify the name of the projet's master thesaurus. But if a thesaurus is already loaded, the modifying of the name will not be accepted.

- `Create corpus from many documents` allows to create a corpus from many documents. This functionnality is used before opening the terminological perspective (step 1). Each document has to be in a txt file and has to be processed through TreeTagger tool.
  The corpus involves all the *.txt* files selected by the user in the corpora directory. It is defined in a *.txt* file. A tagged file involving all the tagged files corresponding to the *.txt* files is created. If the used term extractor is TermoStat, a file involving all the results of TermoStat on each document is created.

- `Add document names` allows to give the names of the several documents. The user gives names separated by semi-colon (;) in the **same order as documents** in the corpus.
- `Remove document names` allows to remove the names of documents. For modifying a name, you have to remove all names and to add all names.
- `Cluster projects (corpora extraction)` allows to cluster corpora extraction with the same extraction tool.

Private

- `TMX result cluster` allows to cluster result extractions with TMX tool (This tool is not open access).

Chapter 6

Terminological level (step 1) perspective

The Terminological level allows to browse and modify the list of domain specific lexical units that have been extracted from the source corpus using term extraction and named entity recognition tools such as YaTeA[1] or the web service for TermoStat [2] and ANNIE[3].
You may also use a list of terms (see 6.2.4) if you have another term extractor.

6.1 Term extractor uses

TERMINAE assumes that the acquisition corpus has been processed by the term extractor beforehand and possibly ANNIE beforehand.

6.1.1 TermoStat web service

Termostat Web is usable after login. The software is still usable for free for research purposes, you only need to create an account. You have to upload an utf-8 txt file involving a document. You download a part of the results by clicking on a disk icon. The result is given in a txt file named "termostat_res.txt".

Put this file in the *repExtractTerm* directory of your project. The acquisition corpus has to be also processed by TreeTagger (Use the script for UTF-8).
Put the treetagger file and the corpus file in the *corpora* directory of your project.
If your corpus involves many documents, each document has to be processed by TermoStat tool and TreeTagger tool. You can use 5.1 item menu to build the corpus from all its documents.

6.1.2 YaTeA tool

TERMINAE assumes that the acquisition corpus has been processed by TreeTagger. YaTeA takes as input:

- The corpus file,
- a tagged corpus (required),
- a list of terms extracted from it as input (required, see Section 6.2.2),

[1] http://search.cpan.org/%7Ethhamon/Lingua-YaTeA-0.621/

[2] http://olst.ling.umontreal.ca/∼drouinp/termostat_web/

[3] http://gate.ac.uk/ie/annie.html

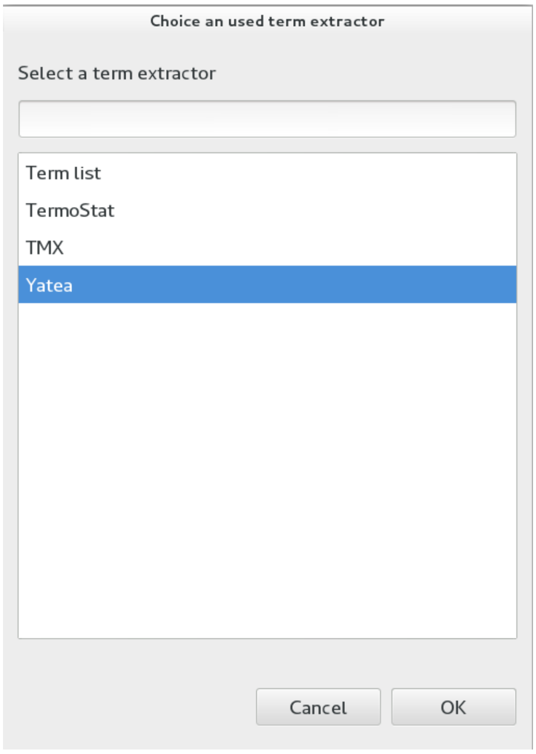


Figure 6.1: Term extractor used

6.2 Data: Terminological files

When you open the `Terminological level perspective`, you have to specify the term extractor used (see figure 6.1).

you have three choices:

- Term list (see 6.2.4)
- TermoStat (see 6.2.1)
- TMX (for Legilocal project)
- Yatea (see 6.2.2)

You may also want to work with named entities. (see 6.2.3).

6.2.1 TermoStat Term files

First you have to specify the terminological data you want to start with (note that additional data can be loaded afterwards).

- Load a term list (`Load TermoStat file`), which is supposed to be located in the `repExtractTerm` subdirectory of your project.
- Select the tagged corpus from which the terms have been extracted (`.tt`, `.ttfr`) file. It is supposed to be located in the `corpora` subdirectory of your project.

- Select the corpus file (`.txt`). It is supposed to be located in the `corpora` subdirectory of your project.

- Speficy the corpus language: English (en) or French (fr_FR).

When the terminological data is loaded, TERMINAE creates one additional file in the `corpora` directory:

- `fTempCorpus2XML.xml` which is an xml version of the corpus.

If you have several documents (see 5.1), each one must be processed by TreeTagger and the results must be concatenated in a single file where the various intial documents are separated by a document tag as shown below:

`Text_n TAB Document TAB n` where TAB is the tabulation character and n varies between 0 and x-1 (x being the total number of documents).

6.2.2 Yatea Term files

First you have to specify the terminological data you want to start with (note that additional data can be loaded afterwards).

- Load a term list (`Load Yatea file`), which is supposed to be located in the `repExtractTerm` subdirectory of your project.

- Indicate how many documents your corpus encompasses. Note that documents are numbered starting from 1 if there are several of them but that a single document has number 0.

- Select the tagged corpus from which the terms have been extracted (`.tt`, `.ttfr`) file. It is supposed to be located in the `corpora` subdirectory of your project.

- Select the corpus file (*.txt*).

- Speficy the corpus language: English (en) or French (fr_FR).

When the terminological data is loaded, TERMINAE creates two additional files in the `corpora` directory:

- `fTempCorpus2XML.xml` which is an xml version of the corpus.

If you have several documents, each one must be processed by TreeTagger and the results must be concatenated in a single file where the various intial documents are separated by a document tag as shown below:

`Text_n TAB Document TAB n` where TAB is the tabulation character and n varies between 0 and x-1 (x being the total number of documents).

6.2.3 Named entity files

You may also want to work with named entities. In that case, you need two files that are output by the ANNIE named entity recognition tool (see Annex 12.9 for details on the file format) and which are expected to be located in the `corpora` subdirectory of your project:

- The first xml file indicates which named entity types you are interested in.

- The second xml file contains the list of named entities extracted by ANNIE.

To create such files, follow the procedure described in Annex 12.10.

6.2.4 Term list files

- Load a term list (`Load term file`), which is supposed to be located in the `repExtractTerm` subdirectory of your project. The format is a term by line.
- Select the tagged corpus from which the terms have been extracted (`.tt` file, `.ttfr`). It is supposed to be located in the `corpora` subdirectory of your project.
- Select the corpus file (`.txt`). It is supposed to be located in the `corpora` subdirectory of your project.
- Speficy the corpus language: English (en) or French (fr_FR).

When the terminological data is loaded, TERMINAE creates one additional file in the `corpora` directory:

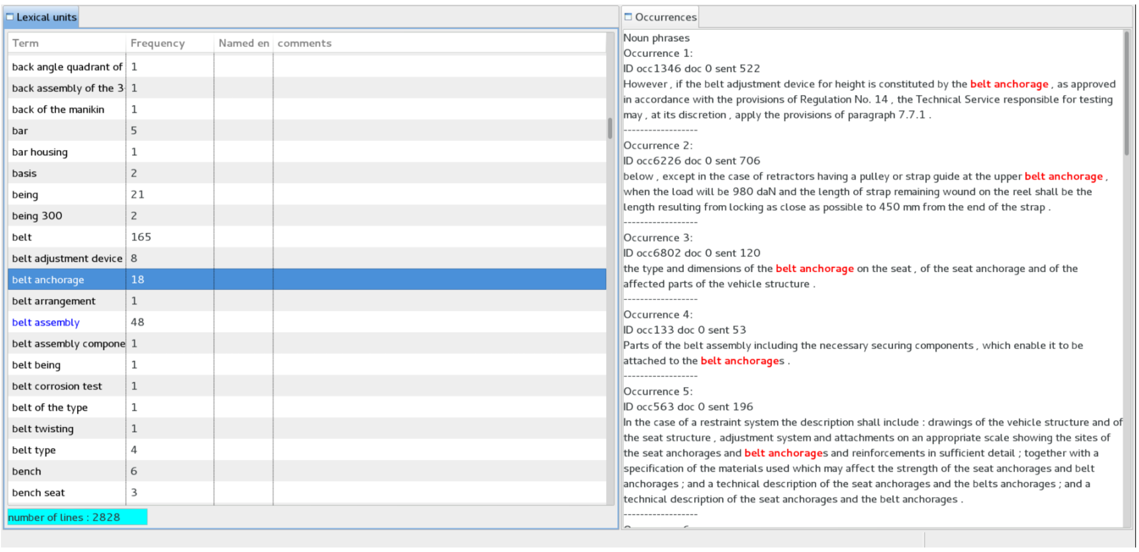
- `fTempCorpus2XML.xml` which is an xml version of the corpus.

If you have several documents, each one must be processed by TreeTagger and the results must be concatenated in a single file where the various intial documents are separated by a document tag as shown below:

`Text_n TAB Document TAB n` where TAB is the tabulation character and n varies between 0 and x-1 (x being the total number of documents).

6.3 Perspective overview

If everything works properly when loading the terminological data, the window of Figure 6.2 appears when the `Terminological level (step 1)` perspective is first opened.



Term	Frequency	Named en	comments
back angle quadrant of	1		
back assembly of the 3	1		
back of the manikin	1		
bar	5		
bar housing	1		
basis	2		
being	21		
being 300	2		
belt	165		
belt adjustment device	8		
belt anchorage	18		
belt arrangement	1		
belt assembly	48		
belt assembly compone	1		
belt being	1		
belt corrosion test	1		
belt of the type	1		
belt twisting	1		
belt type	4		
bench	6		
bench seat	3		

number of lines : 2828

```
Noun phrases
Occurrence 1:
ID occ1346 doc 0 sent 522
However , if the belt adjustment device for height is constituted by the belt anchorage , as approved
in accordance with the provisions of Regulation No. 14 , the Technical Service responsible for testing
may , at its discretion , apply the provisions of paragraph 7.7.1 .
------------------
Occurrence 2:
ID occ6226 doc 0 sent 706
below , except in the case of retractors having a pulley or strap guide at the upper belt anchorage ,
when the load will be 980 daN and the length of strap remaining wound on the reel shall be the
length resulting from locking as close as possible to 450 mm from the end of the strap .
------------------
Occurrence 3:
ID occ6802 doc 0 sent 120
the type and dimensions of the belt anchorage on the seat , of the seat anchorage and of the
affected parts of the vehicle structure .
------------------
Occurrence 4:
ID occ133 doc 0 sent 53
Parts of the belt assembly including the necessary securing components , which enable it to be
attached to the belt anchorages .
------------------
Occurrence 5:
ID occ563 doc 0 sent 196
In the case of a restraint system the description shall include : drawings of the vehicle structure and of
the seat structure , adjustment system and attachments on an appropriate scale showing the sites of
the seat anchorages and belt anchorages and reinforcements in sufficient detail ; together with a
specification of the materials used which may affect the strength of the seat anchorages and belt
anchorages ; and a technical description of the seat anchorages and the belts anchorages ; and a
technical description of the seat anchorages and the belt anchorages .
------------------
```

Figure 6.2: Visualisation of term extractor results

The window is composed of two views: the `Lexical units` view on the left and the `Occurrences` view on the right.

The terminological units (either terms or named entities) are listed on the left view. By clicking on the heads of the columns, you can sort the list alphabetically (Term), by frequency (Frequency) or by type (terms vs. named entities) and named entity type (Named entity).

The last column of the Lexical units view allows to write comments: if you click on a cell comment, a text field appears and you can add a comment to the corresponding terminological unit. The comments are saved with the terminological results and can be reloaded upon request when the term extractor results are loaded.

The occurrences of the selected terminological unit in the working corpus appear on the right view.

6.4 Linguistic actions menu

The action menu associated with the Terminological level (step 1) perspective is the Linguistic action menu. It proposes 5 submenus, which are also contextually accessible from the right click of the mouse:

- File submenu
- Display submenu
- Cleaning submenu
- Term management submenu
- To terminological step submenu

Those submenus are presented in the following subsections.

6.4.1 File submenu

This menu allows to load and save terminological data. It proposes the following actions:

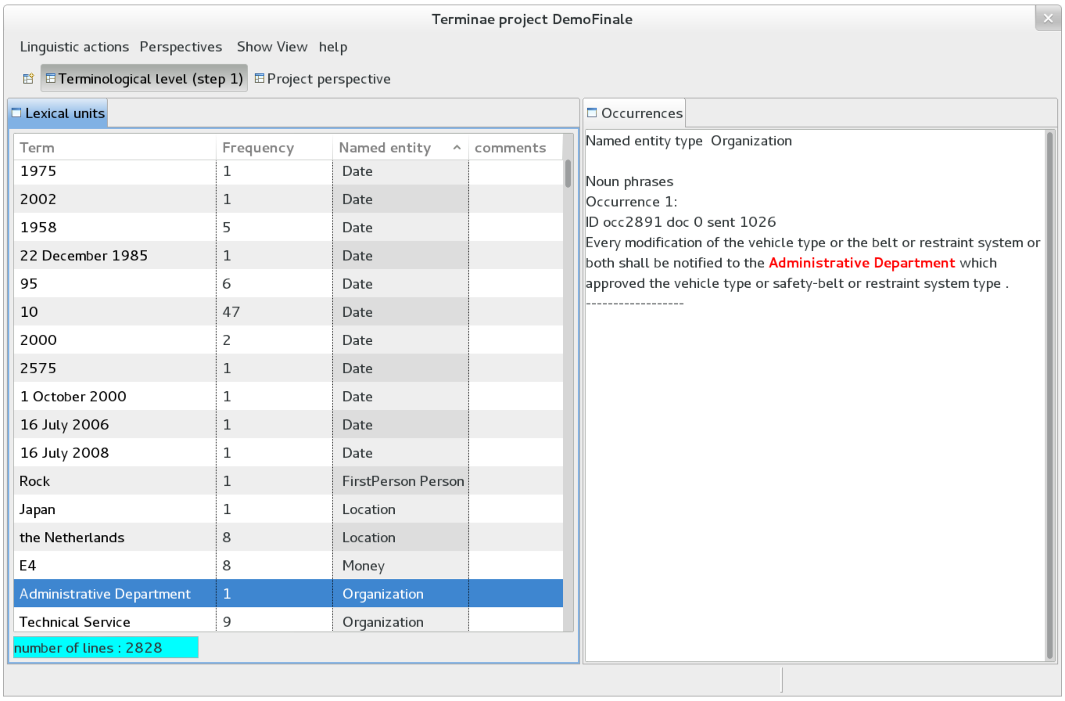
- Load term extractor results to load the terms initially extracted from your corpus by the term extractor or saved in a XML backup. The procedure is the same as that described in Section 6.2.
- Save term extractor results to make an XML backup (see Annex 12.1 for details on the file format).
- Load named entities from ANNIE results to load the named entities identified by the ANNIE named entity recognition tool (see Section 6.2.3):
  - First, you have to indicate the number of the document (0 if only one document) for which you have used Annie tool.
  - Secondly, a file dialog window opens, in which you have to indicate which named entity types you are interested in by selecting a named entity type XML file that should be located in the corpora subdirectory of your project.
  - A second file dialog window opens, in which you have to select another xml file containing the list of named entities extracted by ANNIE. This file should also be located in the corpora subdirectory of your project.

- `Save named entities` to make an XML backup (see Annex 12.2 for details on the file format).
- `Load named entities` to load the named entities from an XML backup.
- `Load all lexical units` to load the terms and named entities from a single XML backup.
- `Save all lexical units` to make an XML backup of all entities (terms and named entities) (see Annex 12.3 for details on the file format).
- `Load new term extractor result` to load a new term extractor result if you want to load a new version of a term extractor result or another term extractor result.
- `Compare lexical unit list with another` to compare two versions of lexical unit list. The first one is the list of the current project. The second one is the lexical unit list of another project. The lexical unit list is defined in the "ensLexUnit" file which is in the repExtractTerm directory of another project. This is the first functinality for collaboration project.
- `Partial saving`
  * `Save validated terms` allows to save only validated terms.
  * `Save invalidated terms` allows to save only invalidated terms.
  * `Save terms with comments` allows to save only terms which have a comment.

If everything works properly, when all types of terminological data are loaded, the window of Figure 6.3 appears.

6.4.2 `Display` submenu

- `Visualize all terms` to redisplay the list of terminological units after a search sequence.
- `Visualize validated terms` to visualise only validated terms.
- `Visualize non validated terms` to visualise only non validated terms (a non validated term is a candidate term with no terminological form, which is no invalidated).
- `Visualize non processed terms` to visualise only non processed terms (a non processed term is a candiadte term with on terminological form, with no comment which is no invalidated).
- `Vizualise invalidated terms` to visualise invalidated terms.
- `Sort on term length` to sort candidate terms by their length.



Term	Frequency	Named entity	comments
1975	1	Date	
2002	1	Date	
1958	5	Date	
22 December 1985	1	Date	
95	6	Date	
10	47	Date	
2000	2	Date	
2575	1	Date	
1 October 2000	1	Date	
16 July 2006	1	Date	
16 July 2008	1	Date	
Rock	1	FirstPerson Person	
Japan	1	Location	
the Netherlands	8	Location	
E4	8	Money	
Administrative Department	1	Organization	
Technical Service	9	Organization	

number of lines : 2828

Named entity type Organization

Noun phrases
Occurrence 1:
ID occ2891 doc 0 sent 1026
Every modification of the vehicle type or the belt or restraint system or both shall be notified to the Administrative Department which approved the vehicle type or safety-belt or restraint system type .

Figure 6.3: Visualisation of terms and named entities

6.4.3 `Cleaning` submenu

This menu allows to clean up the list of terminological units by removing a certain category of terms or named entities. Various options are proposed:

- `Remove terms listed in a file` allows to suppress all the terminological units that are listed in a given file. You have to give the name of that file, in which the stop words are listed, one at each line.
- `Remove terms involving given characters` allows to clean the list of terminological units on a character basis. You have to type in the list of forbidden characters.
- `Removing numbers` allows to suppress the terms that are numbers.
- `Removing term involving numbers` allows to suppress the terms which involve numbers.
- `Removing adjectives` allows to suppress the terms that are tagged as adjectives.
- `Remove single-character terms` allows to suppress the single-character terms from the list of terminological units.
- `Removing adverbs` allows to suppress the terms that are tagged as adverbs.
- `Removing terms from its frequency` allows to suppress the terms for which its frequency is less than a number (for example 0).

6.4.4 Term Management submenu

This menu allows to manage terminological data, *i.e.* to visualise the list of terminological units and edit it by clustering, removing or adding some of them. For all removing actions, the lemmas of removed lexical units are written in “blacklist file” which is in repExtractTerm directory. This list may be visualized. To open the view, go to `schow view` menu on the top of the window, then click on the `other`, then click on `BlackList view`. A dialog window open where you have to indicate the “blacklist file” to visualize.
The `Term Management` menu proposes 9 different actions:

- `Find a term` to search for a specific unit, on the basis of its beginning characters (CTRL+F).

- `Cluster terms` to cluster several lexical units. You first have to select the various units you want to cluster, then click on the `Cluster terms` action and choose the canonical form you want to keep. The alternative forms are removed from the term list and all their occurrences are attached to the canonical form, which frequency count is updated. For each alternative form, it is proposed to add it as variant of the canonical form. If it is a variant, you have to choose its type (abbreviation, acronym or lexical variant) and its use (allowed, forbidden, hidden, recommended (hidden is used to save the variant as skos:hidden)).

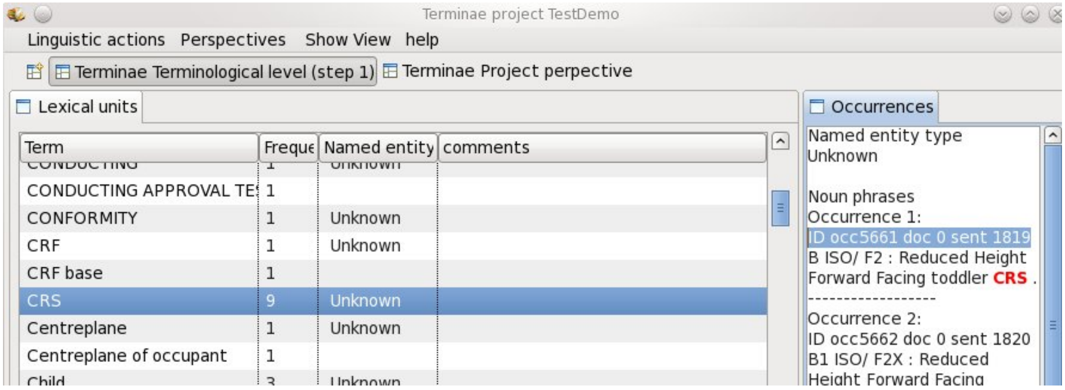
- `Add a term` to add a new term to the term list.

- `Remove a term (Ctrl+R)` to remove the selected term from the list.

- `Undo remove` to undo the last remove action. This may also undo a cleaning action (see Section 6.4.3).

- `View occurrence context` to visualise the surrounding sentences of an occurrence. You have to select the occurrence identifier (see Figure 6.4) and to set the size of the expected context (expressed as a number of sentences).

- `Add occurrence for a term` to enter a new occurrence for a term. You have to select a term and fill the form (see Figure 6.5).


Terminae project TestDemo
Linguistic actions
Perspectives
Show View
help
Terminae Terminological level (step 1)
Terminae Project perpective
Lexical units
Occurrences

Term	Freque	Named entity	comments
CONDUCTING	1	Unknown	
CONDUCTING APPROVAL TE	1		
CONFORMITY	1	Unknown	
CRF	1	Unknown	
CRF base	1		
CRS	9	Unknown	
Centreplane	1	Unknown	
Centreplane of occupant	1		
Child	3	Unknown	

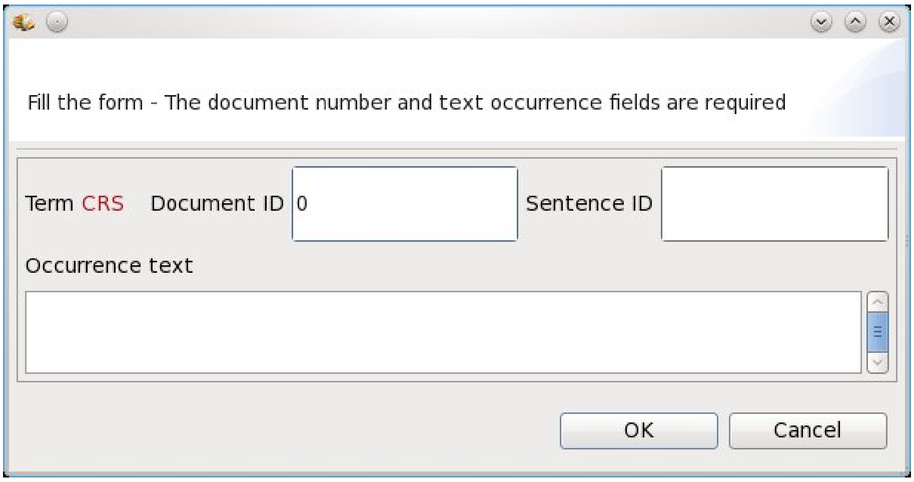
Named entity type
Unknown

Noun phrases
Occurrence 1:
ID occ5661 doc 0 sent 1819
B ISO/ F2 : Reduced Height Forward Facing toddler **CRS** .

Occurrence 2:
ID occ5662 doc 0 sent 1820
B1 ISO/ F2X : Reduced Height Forward Facing

Figure 6.4: Select an occurrence identifier

- `Remove occurrence(s) for a term` to remove occurrence(s) for a term. You have to select the identifier of occurrences you want to remove.



Fill the form - The document number and text occurrence fields are required

Term CRS Document ID 0 Sentence ID

Occurrence text

Figure 6.5: Add occurrence for a term

- `Select terms by document(s)` This action is used when the corpus has several documents. You search candidate terms which are presents in one or many documents. A dialog window opens in which you have to define the number identifying documents.
- `Add as variant` to add the selected lemma as a variant of a term already defined by a terminological form.
- `Invalidated term (Ctrl+I)` to invalidate a term.
- `Re-enable term` to re-enable a term which has been invalidated.

6.4.5 To terminological step submenu

This menu is used to define terminological forms described in next chapter.

- `New terminological form(s) (CTRL+T)` allows to create terminological form(s) for selected term(s). Once terminological form(s) is/are created, new form(s) can be visualized on the `Terminological level (step 2)` perspective, which is automatically opened, and lexical unit(s) which form has/have been created is/are displayed in blue character in the `Lexical units` view (`Terminological level (step 1)` perspective. If the number of occurrences is greater than 100, a window dialog opens to ask if the occurrences have to be all kept. If the response is no, a window dialog opens to define the number of occurrences to keep.
- `To terminological form` allows to visualise the terminological form of the selected terminological unit if it has one. This action automatically switches from the `Terminological level (step 1)` perspective to the `Terminological level (step 2)` perspective.

6.5 Occurrence view - Popup menu

A popup menu is associated to the occurrence view. The actions are accessible from the right click of the mouse.

- `Add occurrence for a term` to enter a new occurrence for a term. You have to select a term and fill the form (see Figure 6.5).

- `View occurrence context` to visualise the surrounding sentences of an occurrence. You have to select the occurrence identifier (see Figure 6.4) and to set the size of the expected context (expressed as a number of sentences).

- `Remove occurrence(s) for a term` to remove occurrence(s) for a term. You have to select the identifier of occurrences you want to remove.

- `Find`(CTRL+F) to search some group of words in the occurrence view. If it exists, it may appear in blue. If there are many group of the same words, they appear in blue.

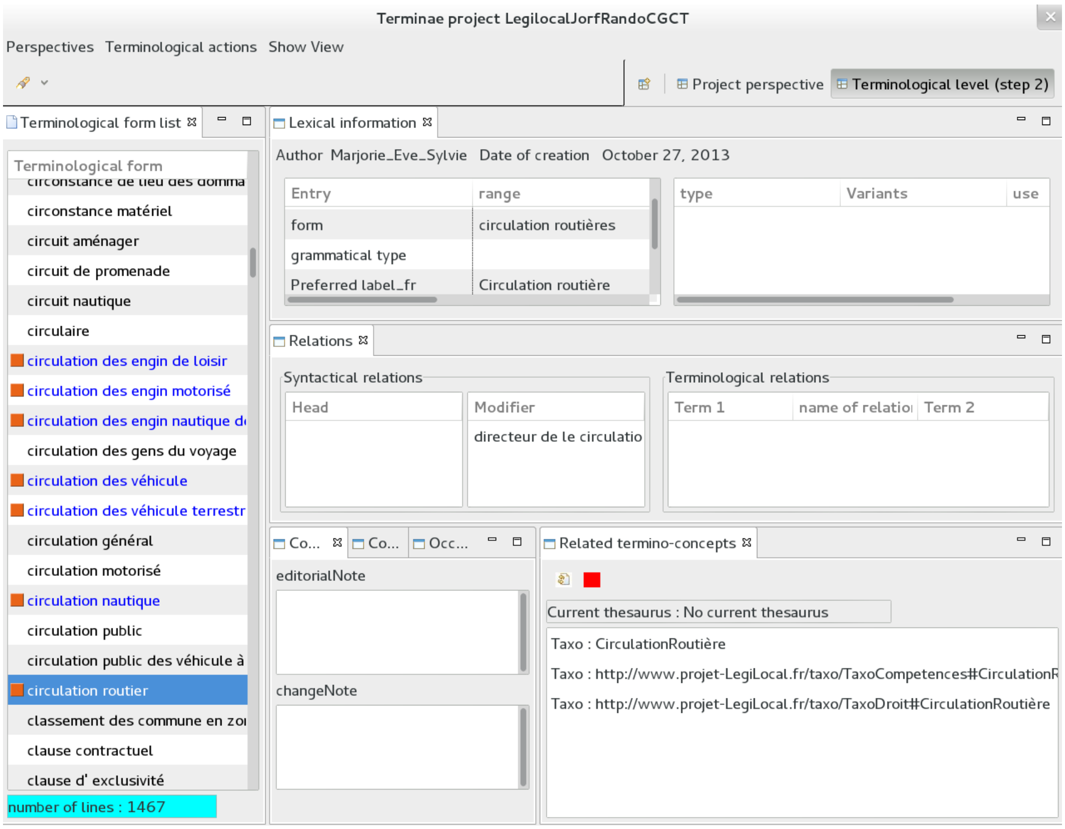
Chapter 7

Terminological level (step 2) perspective

This perspective can be opened either by creating a terminological form or from the main Perspective menu (`Terminological level (step 2)`).

7.1 Perspective overview

The `Terminological level (step 2)` perspective is composed of two main parts, with a global view on the left and a set of more detailed and dependant views on the right (see Figure 7.1):



Terminological form

- circonstance de lieu des domma
- circonstance matériel
- circuit aménager
- circuit de promenade
- circuit nautique
- circulaire
- 🟧 circulation des engin de loisir
- 🟧 circulation des engin motorisé
- 🟧 circulation des engin nautique d
- circulation des gens du voyage
- 🟧 circulation des véhicule
- 🟧 circulation des véhicule terrestr
- circulation général
- circulation motorisé
- 🟧 circulation nautique
- circulation public
- circulation public des véhicule à
- 🟧 circulation routier
- classement des commune en zo
- clause contractuel
- clause d' exclusivité

number of lines : 1467

Author Marjorie_Eve_Sylvie Date of creation October 27, 2013

Entry	range
form	circulation routières
grammatical type	
Preferred label_fr	Circulation routière

type	Variants	use

Head	Modifier
	directeur de le circulatio

Term 1	name of relatio	Term 2

Current thesaurus : No current thesaurus

Taxo : CirculationRoutière
Taxo : http://www.projet-LegiLocal.fr/taxo/TaxoCompetences#CirculationR
Taxo : http://www.projet-LegiLocal.fr/taxo/TaxoDroit#CirculationRoutière

Figure 7.1: `Terminological level (step 2)` perspective

- The `Terminological form list` view is by default presented on the left part of the perspective. It gives the lists of all the canonical terminological units for which a terminological form has been created (the form can be `In progress` (red color), `ToDo`(black color) or `Done` (blue color)). A brown square indicates that the terminological form is associated with a terminoConcept.

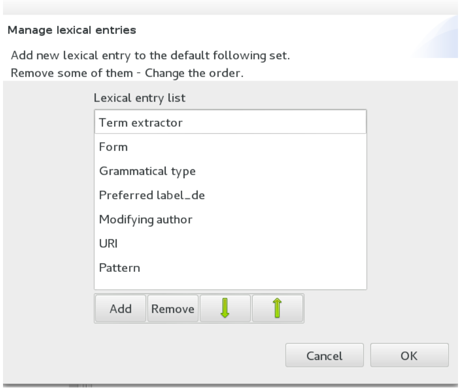
- The other views form the terminological form of the unit that has been selected in the `Terminological form list` (see Section 7.2).

7.2 Data: Terminological forms

An example of terminological form is displayed on the right part of Figure 7.1. A terminological form gathers all the lexical and terminological information that has been collected or manually added for a given term or named entity. It is usually composed of the following views:

- The `Lexical information` view is a form in which you can freely create, modify or suppress some fields. At the first time the terminological perspective is open, a configuration dialog window opens (see Figure 7.2). You can choose the lexical fields. By default seven lexical fields are defined:
  - `Term extractor` , which range is X if the terminological unit has been extracted by term extractor named X,
  - `form`, which gives its form,
  - `grammatical type`, which gives its grammatical category;
  - `Preferred label_fr`, which gives the preferred label in french;
  - `Preferred label_en`, which gives the preferred label in english;
  - `URI`, which gives the URI of the form;
  - `Pattern` a regular expression to find variants of the preferred label.

  The first three fields are automatically filled in by information provided by term extractor if a term extractor has been used.



Manage lexical entries

Add new lexical entry to the default following set.
Remove some of them - Change the order.

Lexical entry list

Term extractor
Form
Grammatical type
Preferred label_de
Modifying author
URI
Pattern

Figure 7.2: Lexical entry dialog window

- The `Variants` view lists all the lexical forms that are associated as variants to the canonical form. They can be found in the corpus and automatically added if a cluster has been created beforehand or manually added. A variant has 2 attributes :

  - its type : abbreviation, acronym or lexical variant
  - its use : allowed, forbidden, hidden, recommended (hidden is used to save the variant as skos:hidden)

- The `Relations` view presents the relations that the terminological unit has:
  - The `Syntactical relations` list shows the phrases to which it belongs either as a head or as a modifier. The syntactical information is provided by YaTeA analysis of the corpus.
  - The `Terminological relations` list shows what are its terminological relationships. In the current version of the TERMINAE platform, the terminological relations have to be filled manually.

- The `Comment for administrator` view to indicate comments for the administrator. It contains two fields (editorialNote (see skos:editorialNote), changeNote (see skos: changeNote)). You have to save the contents by clicking on save (right click of the mouse).

- The `Comment for user` view to indicate comments for the user. It contains three fields (definition (see skos:definition), historyNote (see skos: historyNote), scopeNote (see skos:scopeNote)). You have to save the contents by clicking on save (right click of the mouse).

- The `Occurrences` view lists all the occurrences of the terminological unit that have been identified. They can be occurrences of the canonical form or of any of its alternative (variant) form.

- The `Related termino-concepts` view shows to which termino-concepts the terminological unit is related. A term may be bound to many termino-concepts in different thesauri.

Each terminological form is saved in an XML file in the `terminoFormDir` directory. The list of terminological forms is saved in the file `tableTermeFiches.xml` in `terminoFormDir` directory.

7.3 `Terminological actions` menu

The action menu associated with the `Terminological level (step 2)` perspective is the `Terminological action` menu. It proposes 3 submenus which are presented in the following subsections:

- `File` submenu
- `Display` submenu
- `Collaboration` submenu
- `Form management` submenu
- `Feature management` submenu

- `To terminological level (step 1)` action
- `Termino-concept management` submenu

The corresponding actions are also contextually accessible from the right click of the mouse.

7.3.1 `File management` submenu

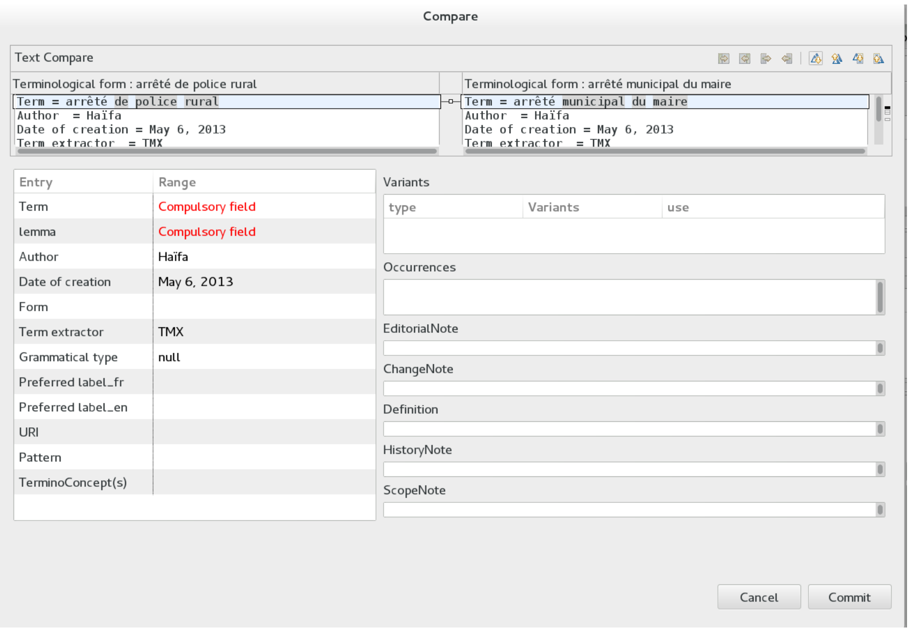
- `Export (skos) form set` to export in skos the set of forms. For each form, the term is described by the skos:prefLabel, the variants are described by skos:altLabel, the use of each variant is described by skos:note as "altLabel *variant* (use)", except for the "hiddenLabel" type which is described by skos:hiddenLabel. The comment is described by skos:editorialNote, terminoConcepts are described by skos:related.(see DTD 12.5).
- `cvsToForm`: not described

7.3.2 `Display` submenu

- `Visualize all forms` to redisplay the list of all forms after a search sequence.
- `Refresh` to trigger a refresh of the window.

7.3.3 `Collaboration` submenu

- `Load terminological form from another project` to load a terminological form from another project. The terminological form file is copied into the terminological form directory and the terminological form is visible in the list of terminological form list.
- `Compare terminological form list with another` to compare a list of terminological forms with another list of terminological forms. This functinality is useful when there are many users on the same project. A dialog window opens to define the "tableTermeFiches" file to open. This file is in the "terminoFormDir" directory of a project.
- `Add a set of terminological forms from another project` to add a list of terminological forms from another project. A dialog window opens to define the "tableTermeFiches" file to open. This file is in the "terminoFormDir" directory of a project. All the forms defined in the "tableTermeFiches" are added.
- `Merging 2 forms` to merge 2 forms. These forms have to be selected before this item is used. A window open (see fig 7.3). The upper part of the window is a form comparing window. Each form field is presented and is compared to the equivalent field of the other form. Upper right buttons may be used to see all differences. The lower part of the window is divided into two parts:
  - The left part window contains the fields of the merging form. These fields have to be filled by fields from initial forms using copy/paste action.
  - The right part window contains three fields:
    - Variants: this field is filled by identical variants from the two forms. The user may add other variant from each form using copy/paste action.



Terminological form : arrêté de police rural

```
Term = arrêté de police rural
Author  = Haïfa
Date of creation = May 6, 2013
Term extractor  = TMX
```

Terminological form : arrêté municipal du maire

```
Term = arrêté municipal du maire
Author  = Haïfa
Date of creation = May 6, 2013
Term extractor  = TMX
```

Entry	Range
Term	Compulsory field
lemma	Compulsory field
Author	Haïfa
Date of creation	May 6, 2013
Form	
Term extractor	TMX
Grammatical type	null
Preferred label_fr	
Preferred label_en	
URI	
Pattern	
TerminoConcept(s)	

type	Variants	use

Figure 7.3: Merging 2 forms

- ∗ Occurrences: this field is filled by identical occurrences from the two forms. The user may add other occurrence from each form using copy/paste action. He/she may add all occurrences using “paste all occurrences” item from contextual menu.
- ∗ Comments: these fields are filled by the user.

If all fields are filled, the user has to click on the commit button. The resulting form is added to the list of forms. A dialog window opens in which the user has to decide if old forms have to be removed.
If some field is not defined, the click on the commit button will bring about an error. Only the cancel button may be used to finish the merge action if the merging has errors.

7.3.4 `Form management` submenu

This submenu proposes two actions related to terminological forms:

- `Find form` to search for a form, on the basis of its beginning characters (CTRL+F).
- `Modify author` to modify the author of the selected terminological form. By default, the author is the project author.
- `Modify terminological form status`: this action is used to note that the work on this terminological form is developed or is completed. It acts as a comment aimed at the user.
- `Remove a terminological form` to remove the selected terminological form.
- `New terminological form` to create a terminological form from scratch. A dialog window opens to define the corresponding term for which a terminological form is created.

- Modify term to modify the term which identifies the terminological form.

- Select from status to select all forms having same state. A dialog window appears to select the wished state.

7.3.5 Feature management submenu

This submenu proposes various actions related to the detailed information provided for a given terminological unit and recorded in its terminological form. It proposes one item and five submenus which are presented in the following subsections.

- Lexical entry configuration is used to modify the set of lexical entries. A configuration dialog window opens (see Figure 7.2). You can choose the lexical fields. By default seven lexical fields are defined:

  - Term extractor, which range is X if the terminological unit has been extracted by term extractor named X,
  - form, which gives its form,
  - grammatical type, which gives its grammatical category;
  - Preferred label_fr, which gives the preferred label in french;
  - Preferred label_en, which gives the preferred label in english;
  - URI, which gives the URI of the form;
  - Pattern a regular expression to find variants of the preferred label.

  The first three fields are automatically filled in by information provided by term extractor if a term extractor has been used.

- Lexical entry management submenu

- Variant submenu

- Syntactical relation management submenu

- Terminological relation management submenu

- Occurrence management submenu

- Create a terminological form for a syntactical relation to create a terminological form for a term selected in syntactical relation view.

Lexical entry management submenu

- Add a lexical entry to add a lexical entry for the selected term. You have to type in the entry name and its value separated by two points.

- Modify value lexical entry to modify the value of the lexical entry.

- Modify lexical entry to modify the lexical entry.

- Remove a lexical entry to remove a lexical entry.

Variant submenu

- `Add a variant` to add a lexical variant of the selected term.
- `Remove a variant` to remove a lexical variant of the selected term.
- `Modify value variant` to modify the value of the variant.
- `Modify type variant` to modify the type of the variant.
- `Modify use variant` to modify the use of the variant.

Syntactical relation management submenu

- `Add a syntactical relation-head` to add a phrase where the selected term is the head.
- `Remove a syntactical relation-head` to remove the selected relation.
- `Add a syntactical relation-modifier` to add a phrase with the selected term as a modifier.
- `Remove a syntactical relation-modifier` to remove the selected relation.

Terminological relation management submenu

- `Add a terminological relation` to add a terminological relation where the selected term is `term1` or `term2`,
- `Remove a terminological relation` to remove a terminological relation.

Occurrence management submenu

- `Add an occurrence` to add an occurrence to the selected term. You have to specify the document identifier and to type in the text of the occurrence.
- `Remove occurrence(s)` to remove an occurrence to the selected term. Select the relevant occurrence(s) to indicate which occurrence has(ve) to be removed.
- `Find occurrences for a term in corpus` to find occurrences in a corpus for a term. This functionality is useful when the forms are created and the user wishes to find occurrences for a term from a specified corpus (the corpus is defined by two files: a “txt” file and the result of the “txt” file by treetagger).
- `Find occurrences for all terms` to find all occurrences in a corpus. This functionality is useful when the forms are created and the user wishes to find occurrences for all terms from a specified corpus (the corpus is defined by two files: a “txt” file and the result of the “txt” file by treetagger).

7.3.6 To Terminological level (step 1) action

`To terminological level (step 1)` to go to previous pespective and to select the lemma corresponding to the terminological form.

7.3.7 `Termino-concept management` submenu

- `Create termino-concept(s) (ALT+G)` to create termino-concept(s) linked to selected terminological unit(s). The termino-concept(s) is/are added to the current (master) thesaurus. If the terminological unit is a named entity, the type of the named entity may also give bearth to a termino-concept and a kindOf link is created between the two termino-concepts.

- `Remove a termino-concept` to remove a termino-concept from the current thesaurus.

- `Rename termino-concept` to rename a termino-concept.

- `Link termino-concept` to link to a termino-concept that already exists. The term is defined as a synonym of the term which is linked to the same termino-concept. A dialog window opens to select the thesaurus involving the termino-concept. The thesaurus must be open. Next, another dialog window opens to choice the termino-concept.

- `To TerminoConceptual level` to switch from the `Terminae Terminological level (step 2)` perspective to the `Terminae TerminoConceptual level` perspective.

- `Remove all terminoConcepts` to remove all created terminoConcepts.

- `Remove link between ter and termino-concept not exists` to remove link between a term and a termino-concept which does not exist because it has been removed from its thesaurus.

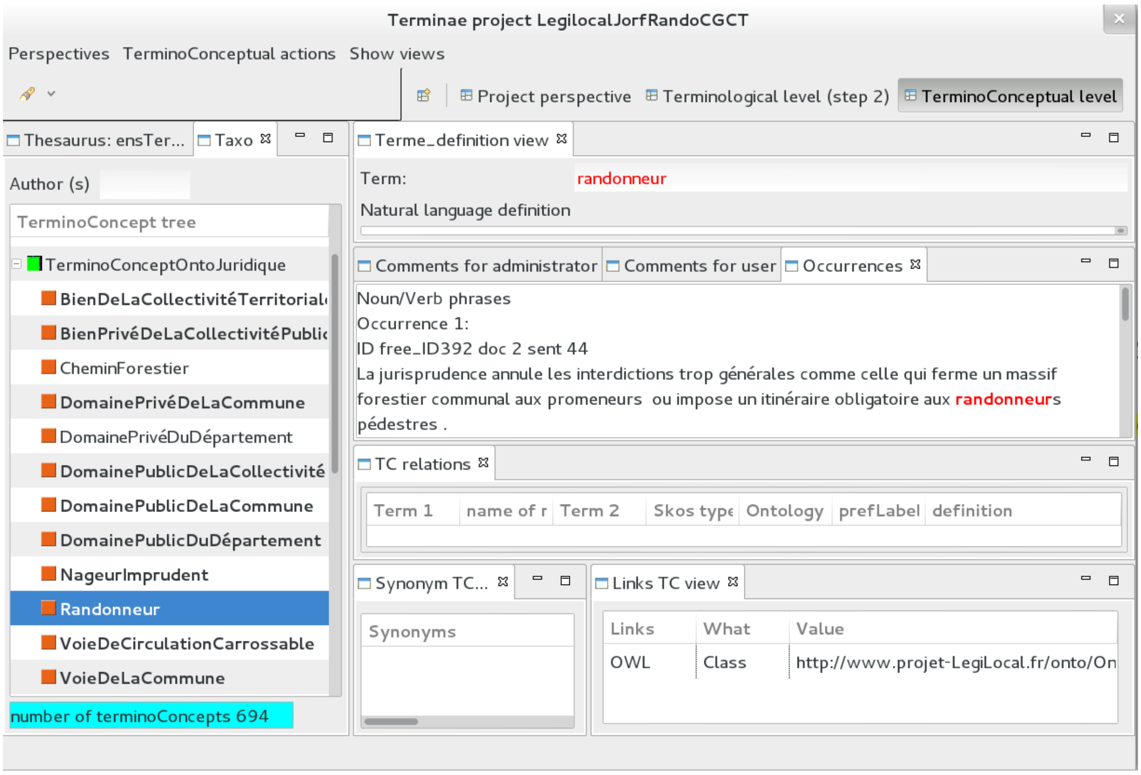
Chapter 8

TerminoConceptual level perspective

This perspective must be opened from the `Perspective` submenu in the main menu by selecting the `Terminae TerminoConceptual level`.

8.1 Perspective overview

The `Terminae TerminoConceptual level` perspective presentation is similar to that of the `Terminological level (step 2)` perspective. It is composed of two main parts, with a set of tabs on the left and a set of more detailed and dependant views on the right (see Figure 8.1):


Terminae project LegilocalJorfRandoCGCT
Perspectives
TerminoConceptual actions
Show views
Project perspective
Terminological level (step 2)
TerminoConceptual level
Thesaurus: ensTer...
Taxo
Author (s)
TerminoConcept tree
TerminoConceptOntoJuridique
BienDeLaCollectivitéTerritoriale
BienPrivéDeLaCollectivitéPublic
CheminForestier
DomainePrivéDeLaCommune
DomainePrivéDuDépartement
DomainePublicDeLaCollectivité
DomainePublicDeLaCommune
DomainePublicDuDépartement
NageurImprudent
Randonneur
VoieDeCirculationCarrossable
VoieDeLaCommune
number of terminoConcepts 694
Terme_definition view
Comments for administrator
Comments for user
Occurrences
TC relations
Synonym TC...
Synonyms
Links TC view

Term: randonneur

Natural language definition

Noun/Verb phrases
Occurrence 1:
ID free_ID392 doc 2 sent 44
La jurisprudence annule les interdictions trop générales comme celle qui ferme un massif forestier communal aux promeneurs ou impose un itinéraire obligatoire aux randonneurs pédestres .

Term 1	name of r	Term 2	Skos type	Ontology	prefLabel	definition

Links	What	Value
OWL	Class	http://www.projet-LegiLocal.fr/onto/On

Figure 8.1: `Terminae TerminoConceptual level` perspective

- Each tab visualizes a thesaurus presented on the left part of the perspective. It shows the hierarchy of all the termino-concepts that belong to the thesaurus. A thesaurus is identified by an URI which is also its name.

- The other views form the termino-conceptual form of the termino-concept that has been selected in the `TerminoConcept tree` (see Section 8.2).

8.2 Data: Termino-conceptual forms

The termino-conceptual level is a bridge between the terminological level and the conceptual level (the ontology). It is made of a set of termino-concepts which are themselves described by termino-conceptual forms gathering the relevant information that has been collected or defined for those termino-concepts. Each termino-concept has a type terminological (T) or structuring (S). A terminological termino-concept is bound to a terminological form. It is display with brown square. A structuring termino-concept is used to structure the thesaurus. It is display with a green square.

A termino-conceptual form is usually composed of the following views:

- The `ID/Term` view shows the ID of the selected termino-concept if the termino-concept display is preferred label otherwise the term or preferred label is displayed.

- The `NL definition` view allows to enter a natural language definition for the selected termino-concept.

- The `Comment for administrator` view to indicate comments for the administrator. It contains two fields (editorialNote (see skos:editorialNote), changeNote (see skos: changeNote)). You have to save the contents by clicking on save (right click of the mouse).

- The `Comment for user` view to indicate comments for the user. It contains three fields (definition (see skos:definition), historyNote (see skos: historyNote), scopeNote (see skos:scopeNote)). You have to save the contents by clicking on save (right click of the mouse).

- The `Occurrences` view presents the occurrences in the corpus of the lexical units to which the termino-concept is linked.

- The `TC relations` view presents the termino-conceptual relations in which the termino-concept is domain or range.

- The `Synonym TC` view presents the synonyms of the termino-concept.

- The `Links` view holds for termino-concepts related to named entities for which type information can be collected. Typical links are `brother`, `father` links. It is also used for describing links to ontologies or links skos. There are three area:
  - Links: to describe type of link as OWL, Skos, unknown
  - What: to describe the link as Class, individual, hiddenLabel@fr, hiddenLabel@en
  - Value: to describe the value

Note that the meaning of a termino-concept is not formally defined. It is mainly described by its related occurrences.

8.3 TerminoConceptual actions menu

The action menu associated with the Terminae TerminoConceptual level perspective is the TerminoConceptual action menu. It proposes 4 submenus which are presented in the following subsections:

- File submenu
- Display submenu
- Thesaurus entity manangement submenu
- Feature management submenu
- Modify author action
- Neon ontology submenu

The corresponding actions are also contextually accessible from the right click of the mouse.

8.3.1 File submenu

This menu allows to load and save termino-conceptual data. It proposes the following actions:

- Load XML format to load a thesaurus in XML format (see DTD in Annex 12.6).
- Save XML format to save a thesaurus in XML format.
- Import SKOS to load an existing thesaurus in Skos format.
- Export SKOS to export a thesaurus in Skos format. A dialog window opens, in which you have to define an URI (added to the name of skos concepts to guarantee they are uniquely identified ; for instance http://www.lipn.univ-paris13.fr/terminae). Note that, in the current version of the TERMINAE platform, the termino-conceptual relations are defined as in the exported file but only its value and its type.
- Export SKOS RDF/XML format to export a thesaurus in RDF/XML format. A dialog window opens, in which you have to define an URI as for the skos format. The termino-conceptual relations are not defined in the exported file.
- New thesaurus to create a new thesaurus. A dialog window opens in which the user gives the name of the thesaurus.
- Remove thesaurus to remove the thesaurus in the active window.
- Private not described.

8.3.2 `Display` submenu

- `Visualize all terminoConcepts` to visualize all terminoConcepts. It is used after a specific select on termino-concepts.
- `Refresh` to refresh the wiew.
- `Termino-concept display` to display termino-concepts with their ID, or their preferred label or their URI. The URI of a termino-concept is formed with the URI of the thesaurus and its ID.
- `Expand tree` to expand all the branches of the tree.
- `Close tree` to reduce the tree to its roots.
- `New window` to open a tab to visualize aotner thesaurus.

8.3.3 `Thesaurus entity management` submenu

It involves four submenus:

- `Termino-concept management` submenu
- `Termino-conceptual relation management` submenu
- `To terminological step` submenu
- `Management between thesauri` submenu

Termino-concept management submenu

- `Find termino-concept` to select a terminoConcept with a part of its name (CTRL+F).
- `Select on type or status termino-concept` to select the termino-concepts on their type or their status which may be (todo, in progress, done).
- `Create termino-concept` to create a new termino-concept. You have to type in the name of the termino-concept if it is not created directly from a terminological unit.
- `Remove termino-concept` to remove the selected termino-concept. You have to confirm the removal.
- `Rename termino-concept` to change the name of the selected termino-concept.
- `Add kindOf link` to give a father to the selected termino-concept. A dialog window opens, in which you have to give the name of the father termino-concept. **This can be done by drag and drop command**:Drag the son node upon the father node.
- `Remove kindOf link` to remove a father of the selected termino-concept.
- `Modify type` to modify the type of a termino-concept (terminological(T) or structuring(S)).
- `Modify status` to modify the status of a termino-concept(todo, in progress, done).

- `Add occurrence` to add an occurrence to the selected termino-concept.

- `Remove occurrence` to remove an occurrence of the selecteed termino-concept. You have to select the identifier of the occurrence to be removed.

Termino-conceptual relation management submenu

- `Add a RTC` to add a termino-concept relation for the selected termino-concept:
  - A first dialog window opens, in which you have to give the name of the relation.
  - A second dialog window opens, in which you have to click on `ok` if the selected termino-concept is the domain and on `cancel` if not.
  - A third dialog window opens, in which you have to give the name of the range or domain (depending on the previous answer). That termino-concept must pre-exist.
  - A choice dialog window then opens, in which you have to select the skos type of the relation.

- `Modify a RTC` to modify a field of a RTC. A first dialog window opens in which the user chooses the field. Asecond dialog window opens in which the user defines the new value.

- `Remove a RTC` to remove the selected termino-conceptual relation.

To terminological step submenu

- `To terminological form` to go to the corresponding terminological form.

- `Create a terminological form` to create a terminological form from a termino-concept. This functionality is useful when you want to add terminological information and occurrences to an existing thesaurus. You start from an existing termino-concept and create a terminological form using a defined corpus.

- `Create all terminological forms` to create all terminological forms from a preexisting thesaurus. This functionality is useful when you want to add terminological information and occurrences to an existing thesaurus. You start from an existing thesaurus and create a terminological form for each termino-concept using a defined corpus.

Management between thesauri submenu

- `Copy TC` to copy a termino-concept from a thesaurus to another one. It is associated with `Paste TC`. You have to copy a TC in a view of the first thesaurus and to paste the TC in the view of other thesaurus.

- `Paste TC` to finish the copy.

- `Copy tree` to copy a tree of thesaurus from a thesaurus to another one. It is associated with `Paste tree`. You have to copy a tree in a view of the first thesaurus and to paste the tree in the view of other thesaurus.

- `Paste tree` to finish the copy.

8.3.4 `Feature management` submenu

This submenu proposes various actions related to the detailed information provided for a given termino-concept and recorded in its termino-conceptual form:

- `Add a synonym` to add a synonym to the selected termino-concept. A dialog window opens for capturing the new synonym. If the corresponding terminological unit has been found by YaTeA or ANNIE, its occurrences are automatically clustered with that of the current termino-concept.

- `Remove a synonym` to remove a synonym. You have to confirm if you want also to remove the related occurrences.

- `Add a link` to add a type of link and its link and its value.

- `Remove a link` to remove a type of link and its link and its value.

- `Modify link type` to modify link type.

- `Modify link` to modify link.

- `Modify link value` to modify value link.

- `Permute term and synonym` to permute the term and the selected synonym.

- `Modify synonym` to modify a synonym.

8.3.5 `Modify author`

action It allows to modify the author of the thesaurus. You may also modify the editing author field.

8.3.6 `Neon ontology` submenu

This menu is used to link TERMINAE and Neon ToolKit. It supports the creation of the conceptual level and many actions to connect it to the termino-conceptual one. It is divided into 2 submenus:

- Neon ontology management

- Entity management

Neon ontology management

- `Create a Neon project` is used to create a Neon toolkit project. If you want to work at the conceptual level, you have to create a Neon project and to specify its name. It is recommended to use different names for the TERMINAE and Neon projects.

- `Create Neon Toolkit ontology` is used to create an ontology. This ontology is part of the newly created Neon project.

- `Link to Neon project` is used when one wants to exploit an existing Neon toolkit project.

- `Link to Neon ontology` is used when one wants to exploit an existing ontology in a specified project.
- `Synchronisation thesaurus and ontology` is used when an ontology has lost its annotations described in the thesaurus. It supposed that the ontology has a good URI.

Entity management

- `Create a class (ALT+C)` is used to create a class in the previous ontology and from the selected termino-concept. A dialog window opens, in which you have to give a name to the class and select a class father in the existing ontology. The class can be visualized in the `Neon toolkit Conceptual level (OWL)` perspective (see Figure 9.1). Note that the class is created with an annotation property in which the link to the source termino-concept and its identifier is saved. Once it has been linked to a class at the conceptual level, the termino-concept is displayed in blue color in the `TerminoConcept tree`.
- `Link to a class` is used to link a termino-concept to an existing class.
- `Create an ObjectProperty (ALT+O)` is used to create an objectProerty from a termino-conceptual relation. A dialog window opens and you have to enter the name of the property, the father object property, its domain and range. The objectProperty is created with an annotation property in which the name and type of the source termino-conceptual relation are saved.
- `Link a RTC and an ObjectProperty` is used to link a termino-conceptual relation to an existing objectProperty.
- `Link a RTC and a class` is used to link a termino-conceptual relation to a an existing class.
- `Create classes and TCs` is used to derive a set of classes from a set of selected termino-concepts. If these termino-concepts have termino-conceptual relations, objectProperties are created and linked to these source relations.
- `Create classes and TCs without dialog` offers the same functionality as above but there without dialog. The default values are systematically kept:
  - name of class = name of terminoconcept,
  - name of objectproperty = name of the RTC,
  - if termino-concepts are linked by a `isKindOf` link, the corresponding classes are in the same hierarchical order.
- `Create an individual` is used to create an individual. You have to enter the individual name and select the class from which it belongs thanks to dialog windows.
- `Link to an individual` is used to link a termino-concept to an individual. You have to enter the individual name and select the class from which it belongs thanks to dialog windows.

- `To ontology level` is used to switch from the termino-conceptual perspective to the OWL one. This action opens the OWL perspective and shows the class corresponding to the selected termino-concept.

Chapter 9

Neon toolkit Conceptual level (OWL) perspective

The conceptual perpective is a Neon toolkit plugin (version 2.4) to which a specific menu has been added for the TERMINAE platform to link the conceptual and termino-conceptual levels. When using Neon toolkit conceptual level perspective, you need to create or to import a Neon toolkit project (which is different from the Terminae project), and to create or import an ontology in this project.

This can be done either from theNeon ontology submenu of the Terminae TerminoConceptual perspective (Create a Neon project and Create Neon Toolkit ontology items) , or create the project and the ontology from the menu of the navigator view in Neon toolkit conceptual level perspective (click right).

In the Neon toolkit conceptual level perspective, you can also import an existing project. In this case, you have to refresh the view to display the imported project and to link it to the terminoConceptual perspective (see the following section). You can also import an ontology, use import item from the menu of the navigator view of Neon toolkit conceptual level perspective.

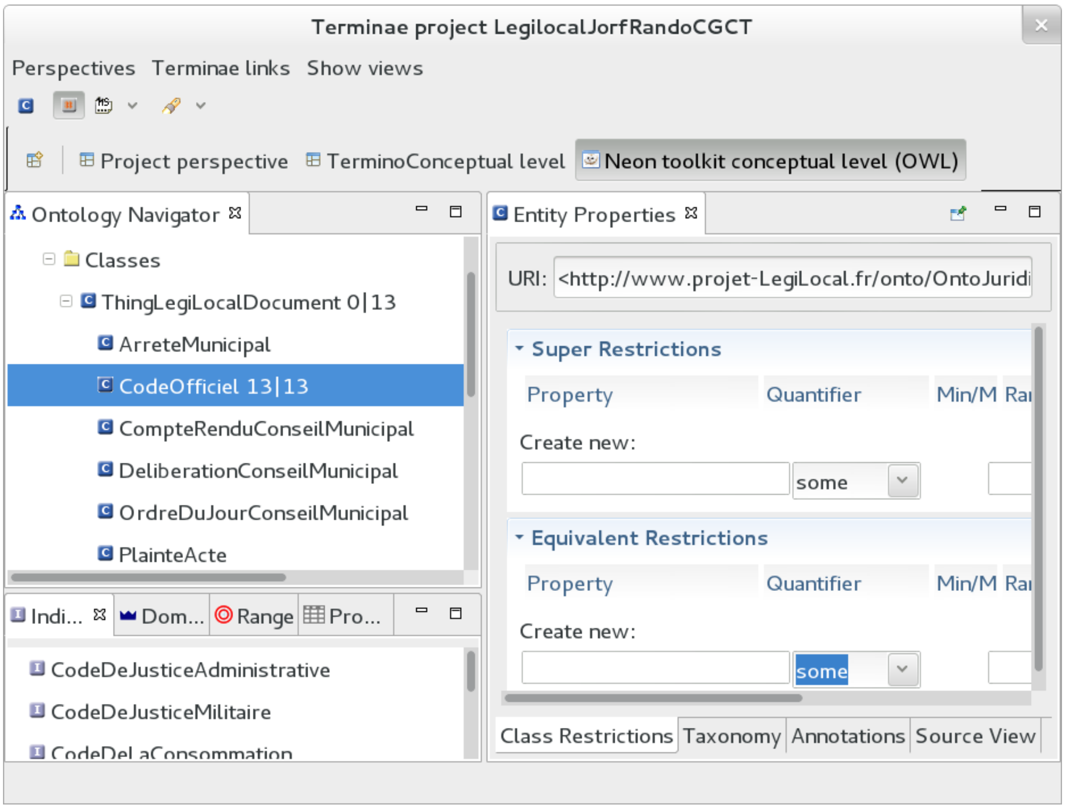
9.1 Perspective overview

The Neon toolkit Conceptual level (OWL) perspective presentation is very similar to that of the Terminae TerminoConceputal level perspective. It is composed of two main parts, with a global view on the left and a set of more detailed and dependant views on the right (see Figure 9.1). See the documentation (http://www.neon-toolkit.org/wiki/Documentation_and_Support). **When adding a link between an individual and a terminoConcept, Neon does not keep the modification. You have to export the ontology.**

9.2 Terminae links menu

Terminae links menu has been added to the Neon Toolkit perspective to link the conceptual and the termino-conceptual levels of Neon and TERMINAE projects and of the resulting termino-conceptual resources:

- To terminoConceptual level is used to switch from the Neon toolkit Conceptual level (OWL) perspective to the Terminae TerminoConceputal level perspective. Clicking on this action item (re-)opens the termino-conceptual perspective and



URI: <http://www.projet-LegiLocal.fr/onto/OntoJuridi

Figure 9.1: `Neon toolkit conceptual level (OWL)` perspective

selects the termino-concept associated with the class initially selected in the conceptual perspective. It is not yet implemented for objectProperties.

- `Create a termino-concept` is used to create a termino-concept and link it to the selected entity. This functionality is useful when you want to add thesaurus information to an existing ontology. You start from an existing class and create a termino-concept in the thesaurus of the TERMINAE project.

- `To link a class to a TC` is used to link a class to an existing termino-concept in the thesaurus of the TERMINAE project.

- `Extract thesaurus from lexicalized ontology` is used to create a thesaurus from a lexicalized ontology. A lexicalized ontology includes for many entities (class/objectProperty) skos annotations as:

  - skos:prefLabel
  - skos:altLabel
  - skos:definition
  - skos:hiddenLabel

From these annotations, a terminoconcept network (as thesaurus) is created at the terminoConceptual level. Click on terminoConceptual perpective to visualizee it. Each terminoconcept is linked to its corresponding class. Each terminoConceptual relation is linked to its corresponding objectProperty.

- `Create a lexicalized ontology` is used to create a lexicalized ontology as explaining above. Note that for links between terminoConceptual relations and ObjectProperties, the objectProperty must have a defined domain and a defined range. The ontology which is open is modified. You may export it for saving.

- `Extract thesaurus from Owl entities` is used to create a thesaurus from an Owl ontology. Each class become a terminoconcept named by the normalized name of the class.

- `Lexicalisation` is used to find a lexicalisation of OWL entities. Each class label is broken down into words. Each word is searched in the set of candidate terms or in the set of terms. The results are written in "txt" files. This functionality will be more developed in next version.

Chapter 10

Annotator perspective

This chapter and the tool have been written by F. Lévy, A. Guissé, S. Szulman.

The LIPN Annotator marks the occurrences of given terms in a text with concepts and individuals of an ontology. It outputs a project which can be directly opened by SemEx, the LIPN semantic explorer[1], to explore the annotations, mark and transform rules, etc. The user can alternatively choose to produce plain result files and to work them with her/his own programs. The output format is textual (.html and .txt) and self explaining.

The output format is language independent, as are the algorithms, so the application can in principle be used for any language where its input makes sense – namely where lemmatizing and POS tagging are possible and not too ambiguous.

The Annotator is included as a plugin in SemEx and in Terminae and can be used from them if preferred. Only the installation differs.

In Terminae, the Annotator may be used through the Annotaor perspective.

Linux specific: Eclipse's browser calls native browsing libraries to do its work. Under Linux, you may have to install specific ones: the present version of the annotator relies on Eclipse 3.7, which browser needs a proper installation of one of Mozilla 1.4 GTK2 - 1.7.x GTK2, XULRunner 1.8.x - 1.9.x and 3.6.x (but not 2.x), WebKitGTK+ 1.2.x and newer. If your installed browser is either too old, or too recent, you can install also XULRunner (the autonomous heart of Mozilla, Firefox and Thunderbird), to enable Eclipse browser. In this case, you have to specify where XULRunner is: modify the annotator.ini file in the executable's directory, to initialize org.eclipse.swt.browser.XULRunnerPath, e.g.
-Dorg.eclipse.swt.browser.XULRunnerPath=/home/szulman/outils/xulrunner-sdk/bin
(Of course, you must replace /home/szulman/outils/xulrunner-sdk/bin with your own location);

10.1 Input files

To annotate a document, you need 4 inputs :

- The document itself, in a single text (.txt) file ;
- The output of a morphological analyzer and POS tagger, in three tab-separated columns (word, POS, lemma) ;

[1] The Annotator and SemEx can be found from http://www-lipn.univ-paris13.fr/~szulman/Annotator/annotator.html or http://www-lipn.univ-paris13.fr/fr/rcln or from http://www-lipn.univ-paris13.fr/fr/rcln-logiciels

- A lexicalization file following the SKOS standard, such as provided by TERMINAE when it builds an ontology. This file can also be created or modified with a plain text editor ; Its DTD is defined in the annex part see 12.8.

- One or several ontologies in OWL format.

10.2 How to proceed

The ontologies and their lexicalization can generally be reused for several documents. The POS file is of course document dependent, and must be generated before annotating.

When the Annotator perspective is open, it supposes that the directory of your project is the defined workspace and that the file's encoding is UTF-8.

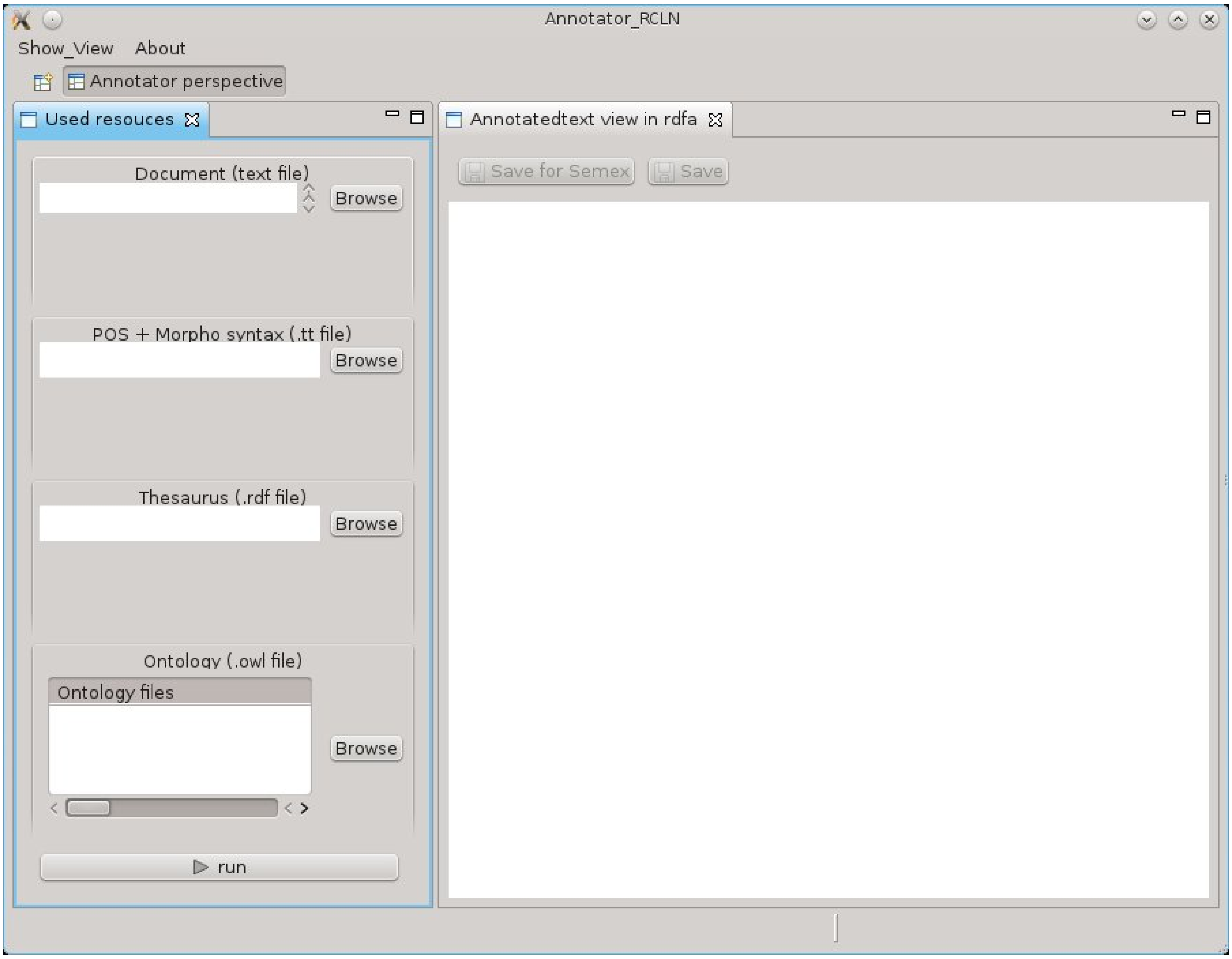


Figure 10.1: The Annotator window

Then a window opens (see fig. 10.1) with four fields in the left pane ("Used resources") and with a blank right pane entitled "Annotated text view". Browse in the four left pane fields for the files which have been prepared. Then run by clicking on the button with a triangle down this pane.

The annotated text appears in the right pane. You can check it and, if satisfied, save it : two buttons up the right pane allow to save either a project which SemEx can use, or only two files describing the annotations according two different formats. Then, if you continue annotating some more files for SemEx, you can store the new results in the same project or create a fresh one.

10.3 Some caveats

The document must be in text format, so pdf and other elaborated files have to be converted. It is required to use the same encoding in the three files where non-ascii characters may appear (text, POS and SKOS). UTF-8 is proposed by default, but other encodings can work too. Due to OS and source files diversity, encoding may need some care. When debugging anomalies, the text and POS file being non homogeneous results in scope errors and misses of the annotations. The SKOS and POS file being non homogeneous results in misses.

Sentence splitting and word splitting are provided by the POS tagger. Depending on it, sentence boarders may happen to be internally incorrect, e.g. because titles have no end point. But the output exactly preserves the appearance of the input (white space, line length, blank lines). Some typography may be ambiguous w.r.t. word splitting (e.g; "the upper/middle class"), and we have had a version of a POS tagger which blows out ÿ - with some poor effects on the annotation.

The lexicalization of the ontology described in the SKOS file associates several lexical forms to a single labeling entity. Each lexical form stores the lemmatized form of words (don't forget it if you create your own SKOS). As this form is also computed by the morphosyntactic parser, lexicalizations are recognized independently of morphological variants. Note that the technique is a bit over-productive, due to ambiguity of lemmas. We plan to improve it by using the POS category. On the other hand, before annotating according to the SKOS file, the labeling entity is checked against the ontology ; if it is not present there, the annotation is skipped. Discrepancies between SKOS and OWL files are logged in annotator.log in the result directory, and it can be wise to check the content of this file;

Chapter 11

Collaboration perspective

This perspective is under building. The perspective objectives are to give functionnalities to allow to compare and to merge two Terminae projects. At the moment, you may only compare two Terminae projects (step 1 and step 2).

`Comparing` menu

`Open comparing view (step 1)` to compare two versions of lexical unit list. The first one is the list of the current project. The second one is the lexical unit list of another project. The lexical unit list is defined in the "ensLexUnit" file which is in the repExtractTerm directory of another project. This is the first functinality for collaboration project. `Open comparing view (step 2)` to compare two versions of terminological form table. This file is in terminoFormDir directory and is named *tableTermeFiches.xml*.

Chapter 12

Annex

This annex lists the DTD used by Teminae.

12.1 XML backup DTD for terms

The DTD of the XML file which contains terms and their occurrences which is visualized in `Terminological level (step 1)` perspective.

```
<!ELEMENT DOC ( #PCDATA ) >
<!ELEMENT END_POSITION ( #PCDATA ) >
<!ELEMENT FORM ( #PCDATA ) >
<!ELEMENT ID ( #PCDATA ) >
<!ELEMENT LEMMA ( #PCDATA ) >
<!ELEMENT LIST_OCCURRENCES ( OCCURRENCE+ ) >
<!ELEMENT LIST_TERM_CANDIDATES ( TERM_CANDIDATE+ ) >
<!ELEMENT List_Variants ( Variant* ) >
<!ELEMENT MORPHOSYNTACTIC_FEATURES ( SYNTACTIC_CATEGORY ) >
<!ELEMENT NUMBER_OCCURRENCES ( #PCDATA ) >
<!ELEMENT OCCURRENCE ( ID, DOC, SENTENCE, START_POSITION,
                                  END_POSITION, Texte ) >
<!ELEMENT SENTENCE ( #PCDATA ) >
<!ELEMENT START_POSITION ( #PCDATA ) >
<!ELEMENT SYNTACTIC_CATEGORY ( #PCDATA ) >
<!ELEMENT TERM_CANDIDATE ( ID, LEMMA, FORM, List_Variants,
NUMBER_OCCURRENCES, LIST_OCCURRENCES, MORPHOSYNTACTIC_FEATURES, comment )
<!ELEMENT TERM_EXTRACTION_RESULTS ( LIST_TERM_CANDIDATES ) >
<!ELEMENT Texte ( #PCDATA ) >
<!ELEMENT Variant EMPTY  >
<!ATTLIST Variant type CDATA #REQUIRED word CDATA #REQUIRED use CDATA #RE
<!ELEMENT comment (#PCDATA) >
```

12.2 XML backup DTD for ENs

The DTD of the XML file which contains named entities and their occurrences which is visualized in `Terminological level (step 1)` perspective.

```
<!ELEMENT DOC ( #PCDATA ) >
<!ELEMENT END_POSITION ( #PCDATA ) >
<!ELEMENT FORM EMPTY >
<!ELEMENT ID ( #PCDATA ) >
<!ELEMENT LEMMA ( #PCDATA ) >
<!ELEMENT LIST_EN ( NAMED_ENTITY+ ) >
<!ELEMENT LIST_OCCURRENCES ( OCCURRENCE* ) >
<!ELEMENT LIST_SENT ( SENT* ) >
<!ELEMENT List_Lemme EMPTY >
<!ELEMENT List_Variants EMPTY >
<!ELEMENT NAMED_ENTITY ( ID, LEMMA, FORM, List_Variants,
         Types, NUMBER_OCCURRENCES, LIST_OCCURRENCES, LIST_SENT ) >
<!ELEMENT NUMBER_OCCURRENCES ( #PCDATA ) >
<!ELEMENT OCCURRENCE ( ID, DOC, SENTENCE, START_POSITION,
                                        END_POSITION, Texte ) >
<!ELEMENT SENT ( ID, offset, phrase, List_Lemme ) >
<!ELEMENT SENTENCE ( #PCDATA ) >
<!ELEMENT START_POSITION ( #PCDATA ) >
<!ELEMENT Texte ( #PCDATA ) >
<!ELEMENT Types ( type+ ) >
<!ELEMENT offset ( #PCDATA ) >
<!ELEMENT phrase ( #PCDATA ) >
<!ELEMENT type ( #PCDATA ) >
```

12.3 EnsLexUnit DTD

The DTD of the XML file which contains terms, named entities and their occurrences which is visualized in `Terminological level (step 1)` perspective.

```
<!ELEMENT DOC ( #PCDATA ) >
<!ELEMENT END_POSITION ( #PCDATA ) >
<!ELEMENT FORM ( #PCDATA ) >
<!ELEMENT ID ( #PCDATA ) >
<!ELEMENT LEMMA ( #PCDATA ) >
<!ELEMENT LIST_EN ( NAMED_ENTITY+ ) >
<!ATTLIST LIST_EN numeroDocument CDATA #REQUIRED>
<!ELEMENT LIST_OCCURRENCES ( OCCURRENCE* ) >
<!ELEMENT LIST_SENT ( SENT* ) >
<!ELEMENT LIST_TERM_CANDIDATES ( TERM_CANDIDATE+ ) >
<!ELEMENT List_Variants ( Variant* ) >
<!ELEMENT MORPHOSYNTACTIC_FEATURES ( SYNTACTIC_CATEGORY ) >
<!ELEMENT NAMED_ENTITY ( Ens_Variants | ID | LEMMA |
   LIST_OCCURRENCES |  LIST_SENT | NUMBER_OCCURRENCES | Types )* >
<!ELEMENT NUMBER_OCCURRENCES ( #PCDATA ) >
<!ELEMENT OCCURRENCE ( ID, DOC, SENTENCE, START_POSITION,
                                  END_POSITION, Texte ) >
<!ELEMENT SENT EMPTY >
<!ATTLIST SENT ID CDATA #REQUIRED >
```

```
<!ELEMENT SENTENCE ( #PCDATA ) >
<!ELEMENT START_POSITION ( #PCDATA ) >
<!ELEMENT SYNTACTIC_CATEGORY ( #PCDATA ) >
<!ELEMENT TERM_CANDIDATE ( ID, LEMMA, NUMBER_OCCURRENCES,
 LIST_OCCURRENCES, FORM, MORPHOSYNTACTIC_FEATURES, List_Variants,
            NAMED_ENTITY?, comment ) >
<!ELEMENT TERM_EXTRACTION_RESULTS ( LIST_TERM_CANDIDATES, LIST_EN ) >
<!ELEMENT Texte ( #PCDATA ) >
<!ELEMENT Types ( type+ ) >
<!ELEMENT Variant EMPTY  >
<!ATTLIST Variant type CDATA #REQUIRED word CDATA #REQUIRED use CDATA #RE
<!ELEMENT type ( #PCDATA ) >

<!ELEMENT comment (#PCDATA) >
```

12.4 Terminological from DTD

```
 <!ELEMENT Fiche_terminologique ( Term, TERM_CANDIDATE, Lexical_informati
<!ELEMENT Term ( #PCDATA ) >
<!ELEMENT TERM_CANDIDATE ( ID, LEMMA, FORM, List_Variants, NUMBER_OCCURRE
<!ELEMENT ID ( #PCDATA ) >
<!ELEMENT LEMMA ( #PCDATA ) >
<!ELEMENT FORM ( #PCDATA ) >
<!ELEMENT List_Variants ( Variant+ ) >
<!ELEMENT NUMBER_OCCURRENCES ( #PCDATA ) >
<!ELEMENT LIST_OCCURRENCES ( OCCURRENCE+ ) >

<!ELEMENT Lexical_information ( Entry | Range )* >
<!ELEMENT Entry ( #PCDATA ) >
<!ELEMENT Range ( #PCDATA ) >

<!ELEMENT Author ( #PCDATA ) >
<!ELEMENT Create_date ( #PCDATA ) >
<!ELEMENT Validating ( #PCDATA ) >

<!ELEMENT Terminological_concept ( Thesaurus, ID ) >

<!ELEMENT OCCURRENCE ( ID, DOC, SENTENCE, START_POSITION, END_POSITION, T
<!ELEMENT DOC ( #PCDATA ) >
<!ELEMENT SENTENCE ( #PCDATA ) >
<!ELEMENT START_POSITION ( #PCDATA ) >
<!ELEMENT END_POSITION ( #PCDATA ) >
<!ELEMENT Texte ( #PCDATA ) >

<!ELEMENT SyntacticalRelationsHead ( ARelSynt ) >
<!ELEMENT SyntacticalRelationsModifier ( ARelSynt ) >
<!ELEMENT ARelSynt ( #PCDATA ) >
```

```
<!ELEMENT TerminologicalRelations ( name, domain, range, Occurrence+ ) >
<!ELEMENT name ( #PCDATA ) >
<!ELEMENT domain ( #PCDATA ) >
<!ELEMENT range ( #PCDATA ) >
<!ELEMENT Occurrence ( #PCDATA ) >

<!ELEMENT Thesaurus EMPTY >
<!ATTLIST Thesaurus name NMTOKEN #REQUIRED >

<!ELEMENT Variant EMPTY >
<!ATTLIST Variant type CDATA #REQUIRED >
<!ATTLIST Variant use CDATA #REQUIRED >
<!ATTLIST Variant word CDATA #REQUIRED >

<!ELEMENT comment ( EditorialNote, ChangeNote, Definition, HistoryNote, S
<!ELEMENT EditorialNote ( #PCDATA ) >
<!ELEMENT ChangeNote ( #PCDATA ) >
<!ELEMENT Definition ( #PCDATA ) >
<!ELEMENT HistoryNote ( #PCDATA ) >
<!ELEMENT ScopeNote ( #PCDATA ) >
```

12.5 DTD export set of forms to skos

```
 <!ELEMENT rdf:Description ( rdf:type | skos:altLabel |
 skos:editorialNote | skos:hiddenLabel | skos:note |
 skos:prefLabel | skos:related )* >
<!ATTLIST rdf:Description rdf:about CDATA #REQUIRED >

<!ELEMENT rdf:RDF ( rdf:Description+ ) >
<!ATTLIST rdf:RDF xmlns:rdf CDATA #REQUIRED >
<!ATTLIST rdf:RDF xmlns:skos CDATA #REQUIRED >

<!ELEMENT rdf:type EMPTY >
<!ATTLIST rdf:type rdf:resource CDATA #REQUIRED >

<!ELEMENT skos:altLabel ( #PCDATA ) >

<!ELEMENT skos:editorialNote ( #PCDATA ) >

<!ELEMENT skos:hiddenLabel ( #PCDATA ) >

<!ELEMENT skos:note ( #PCDATA ) >

<!ELEMENT skos:prefLabel ( #PCDATA ) >
<!ATTLIST skos:prefLabel xml:lang ( en | fr ) #REQUIRED >

<!ELEMENT skos:related EMPTY >
```

```
<!ATTLIST skos:related rdf:resource CDATA #REQUIRED >
```

12.6 Thesaurus DTD

The DTD of the XML file which contains a thesaurus which is visualized in Terminae TerminoConceptual level perspective. A thesaurus contains a collection of terminoconcepts. Each terminoconcept is described by an ID, a natural language definition, corpus occurrences, a prefLabel, a set of “see_also”, a set of synonyms (altLabel), a set of children and its father.

```
<!ELEMENT DOC ( #PCDATA ) >
<!ELEMENT END_POSITION ( #PCDATA ) >
<!ELEMENT EnsTerminoConcepts ( name, TerminoConcept+ ) >
<!ELEMENT ID ( #PCDATA ) >
<!ELEMENT NL_Definition ( #PCDATA ) >
<!ELEMENT OCCURRENCE ( ID, DOC, SENTENCE, START_POSITION,
                     END_POSITION, Texte)>
<!ELEMENT PrefLabel ( #PCDATA ) >
<!ELEMENT RelationRTC ( name, domain, range, Skos_type ) >
<!ELEMENT SENTENCE ( #PCDATA ) >
<!ELEMENT START_POSITION ( #PCDATA ) >
<!ELEMENT See_also ( #PCDATA ) >
<!ELEMENT SetRTC ( RelationRTC? ) >
<!ELEMENT Skos_type ( #PCDATA ) >
<!ELEMENT Synonym ( #PCDATA ) >
<!ELEMENT TerminoConcept ( ID | NL_Definition | OCCURRENCE |
 PrefLabel | See_also | SetRTC | Synonym | children | fathers )* >
<!ELEMENT Texte ( #PCDATA ) >
<!ELEMENT child ( #PCDATA ) >
<!ELEMENT children ( child* ) >
<!ELEMENT domain ( #PCDATA ) >
<!ELEMENT father ( #PCDATA ) >
<!ELEMENT fathers ( father? ) >
<!ELEMENT name ( #PCDATA ) >
<!ELEMENT range ( #PCDATA ) >
```

12.7 TreeTagger English Tagset

```
CC Cooordinating conjunction
CD Cardinal number
DT Determiner
EX Existential there
FW Foreign word
IN Preposition or subordinating conjunction
JJ Adjective
JJR Adjective, comparative
JJS Adjective, superlative
LS list item marker
MD Modal
```

```
NN Noun, singular or mass
NNS Noun, plural
NP Proper noun, singular
NPS Proper noun, plural
PDT Predeterminer
POS Possessive ending
PP Personal pronoun
PP$ Possessive pronoun
RB Adverb
RBR Adverb, comparative
RBS Adverb, superlative
RP Particle
SYM Symbol
TO to
UH Interjection
VB Verb, base form
VBD Verb, past tense
VBG Verb, gerund or present participle
VBN Verb, past participle
VBP Verb, non-3rd person singular present
VBZ Verb, 3rd person singular present
WDT Wh-determiner
WP Wh-pronoun
WP$ Possesive wh-pronoun
WRB Wh-adverb
```

12.8 TreeTagger French Tagset

```
ABR abreviation
ADJ adjective
ADV adverb
DET:ART article
DET:POS possessive pronoun (ma, ta, ...)
INT interjection
KON conjunction
NAM proper name
NOM noun
NUM numeral
PRO pronoun
PRO:DEM demonstrative pronoun
PRO:IND indefinite pronoun
PRO:PER personal pronoun
PRO:POS possessive pronoun (mien, tien, ...)
PRO:REL relative pronoun
PRP preposition
PRP:det preposition plus article (au,du,aux,des)
PUN punctuation
PUN:cit punctuation citation
SENT sentence tag
SYM symbol
```

```
VER:cond verb conditional
VER:futu verb futur
VER:impe verb imperative
VER:impf verb imperfect
VER:infi verb infinitive
VER:pper verb past participle
VER:ppre verb present participle
VER:pres verb present
VER:simp verb simple past
VER:subi verb subjunctive imperfect
VER:subp verb subjunctive present
```

```
<!Element rdf:Description (skos:prefLabel, skos:altLabel*, rdf:type)>
<!ATTLIST rdf:Description rdf:about CDATA>
<!Element prefLabel (#PCDATA)>
<!Element altLabel (#PCDATA)>
<!Element rdf:type EMPTY>
<!ATTLIST rdf:type rdf:resource CDATA>
```

12.9 Use ANNIE to extract named entities

This annex describes the procedure to be followed to use ANNIE to extact named entities from a given document (only one document can be processed at a time).

Note that the following procedure is extracted from the Gate documentation for processing English corpora: http://gate.ac.uk/sale/tao/splitch3.html.

GATE enables you to extract named entities from plain texts and annotate your corpus with it. GATE is distributed with an IE system called ANNIE. ANNIE relies on finite state algorithms and the JAPE[1] language.

Take one large pile of text (documents, emails, etc.). Call this your corpus.

If you right-click on ”Language Resources” in the resources pane, select ”New” then ”GATE Document”, the window ”Parameters for the new GATE Document” will appear.

Once you indicate the corpus to work on it, you can call for ANNIE.

From the File menu, select ”Load ANNIE System”. To run it in its default state, choose ”with Defaults”. This will automatically load all the ANNIE resources, and create a corpus pipeline called ANNIE with the correct resources selected in the right order, and the default input and output annotation sets.

If ”without Defaults” is selected, the same processing resources will be loaded, but a popup window will appear for each resource, which enables the user to specify a name, location and other parameters for the resource. This is exactly the same procedure as for loading a processing resource individually, the difference being that the system automatically selects those resources contained within ANNIE. When the resources have been loaded, a corpus pipeline called ANNIE will be created as before.

The next step is to add a corpus, and select this corpus from the drop-down corpus menu in the Serial Application editor. Finally click on ”Run” from the Serial Application editor, or by right clicking on the application name in the resources pane and selecting ”Run”.

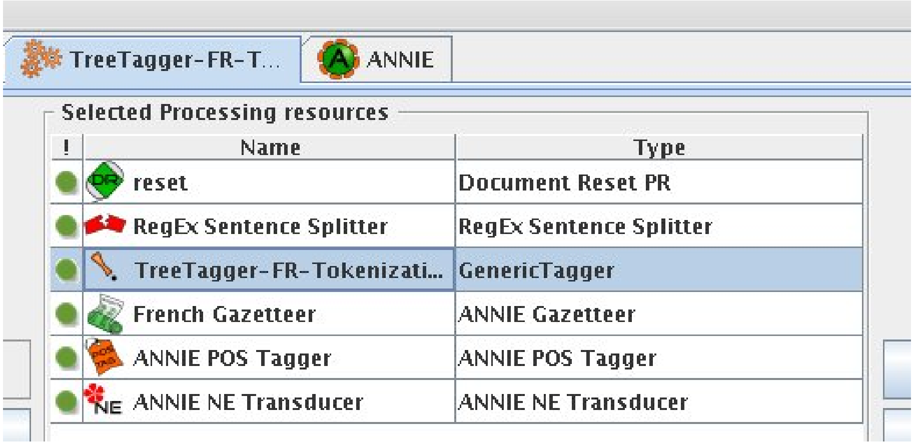
[1] JAPE is a Java Annotation Patterns Engine. It provides finite state transduction over annotations based on regular expressions. JAPE allows you to recognise regular expressions in annotations on documents.

To view the results, double click on one of the document contained in the corpus processed in the left hand tree view. No annotation sets nor annotations will be shown until annotations are selected in the annotation sets; the ”Default” set is indicated only with an unlabelled right-arrowhead which must be selected in order to make visible the available annotations. Open the default annotation set and select some of the annotations to see what the ANNIE application has done.

Having selected an annotation type in the annotation sets view, hovering over an annotation in the main resource viewer or right-clicking on it will bring up a popup box containing a list of the annotations associated with it, from which one can select an annotation to view in the annotation editor, or if there is only one, the annotation editor for that annotation.

Now to save your corpus annotated with ANNIE, right-click on a document in the resources tree and choose ”Save as XML”. In addition, all documents in a corpus can be saved as individual XML files into a directory by right-clicking on the corpus in the resources tree and choosing the option ”Save as XML”.

For French corpora, you have to install treetagger and load the Tagger_Framework plugin. In the resource directory, you find *TreeTagger-FR-Tokenization.gapp*. You load this application in Gate platform.You also load the Lang_French plugin and the french.gapp Gate application. The selected processing resources are defined in Figure 12.1 .



!	Name	Type
●	reset	Document Reset PR
●	RegEx Sentence Splitter	RegEx Sentence Splitter
●	TreeTagger-FR-Tokenizati...	GenericTagger
●	French Gazetteer	ANNIE Gazetteer
●	ANNIE POS Tagger	ANNIE POS Tagger
●	ANNIE NE Transducer	ANNIE NE Transducer

Figure 12.1: Selected processing resources

12.10 Gate named entity type file

The DTD of the XML file which contains named entity type file which is used when loading named entities (see 6.2.3).

```
<?xml version='1.0' encoding='UTF-8'?>
<ensTypeEn>
<typeEn>Organization</typeEn>
<typeEn>Date</typeEn>
<typeEn>Person</typeEn>
<typeEn>Percent</typeEn>
<typeEn>Location</typeEn>
<typeEn>Money</typeEn>
<typeEn>Title</typeEn>
```

```
<typeEn>Address</typeEn>
<typeEn>Unknown</typeEn>
<typeEn>Jobtitle</typeEn>
<typeEn>FirstPerson</typeEn>
<typeEn>Location</typeEn>
<typeEn>UrlPre</typeEn>
</ensTypeEn>
```

12.11 Key binding

```
 CTRL+I = invalidate term
 CTRL+R = remove term
 CTRL+T = new terminological form
 CTRL+X = exit
 CTRL+J = find a term
 CTRL+V = paste a lexical entry (merging form)
 CTRL+F5 = Refresh forms
 ALT+K = add a kindOf link
 ALT+O = create an objectProperty
 ALT+C = create a class
 ALT+G = new terminological form
 ALT+F = find in occurrences
 ALT+Q = select a form
```

12.12 DTD configuration file

```
 <!ELEMENT Application EMPTY >
<!ATTLIST Application nom NMTOKEN #REQUIRED >

<!ELEMENT Auteur EMPTY >
<!ATTLIST Auteur nom NMTOKEN #REQUIRED >

<!ELEMENT Configuration ( Application, Nom_corpus, Nom_thesaurus, Ens_Nom

<!ELEMENT Documents EMPTY >

<!ELEMENT Encodage ( #PCDATA ) >

<!ELEMENT Ens_Nom_thesaurus ( Nom_Un_Thesaurus+ ) >

<!ELEMENT Nom_Un_Thesaurus EMPTY >
<!ATTLIST Nom_Un_Thesaurus nom NMTOKEN #REQUIRED >

<!ELEMENT Nom_corpus EMPTY >
<!ATTLIST Nom_corpus langue NMTOKEN #REQUIRED >
<!ATTLIST Nom_corpus loaded NMTOKEN #REQUIRED >
<!ATTLIST Nom_corpus nom NMTOKEN #REQUIRED >
```

```
<!ELEMENT Nom_thesaurus EMPTY >
<!ATTLIST Nom_thesaurus nom NMTOKEN #REQUIRED >

<!ELEMENT RepertoireAppli ( #PCDATA ) >

<!ELEMENT TermExtractor EMPTY >
<!ATTLIST TermExtractor nom NMTOKEN #REQUIRED >

<!ELEMENT TerminaeNeonLinks EMPTY >
<!ATTLIST TerminaeNeonLinks NamespaceOntology CDATA #REQUIRED >
<!ATTLIST TerminaeNeonLinks OntologyURI CDATA #REQUIRED >
<!ATTLIST TerminaeNeonLinks ProjectName NMTOKEN #REQUIRED >

<!ELEMENT fichierValidationLexicalUnit ( #PCDATA ) >

<!ELEMENT version ( #PCDATA ) >
```