# NAT-term – an authoring tool for terminologies

... Removed to anonymize

**Abstract**

It is well known that terminologies are a crucial resource for many areas such as NLP, knowledge representation and translation.

In this article we discuss NAT-term – a language to author terminologies, and a set of tools to validate, transform and produce a variety of outputs. Initially built for didactic intentions, this tool has also been used in other contexts.

**Keywords**

Terminologies, Dictionaries, Natural Language Processing

## 1. Introduction

In translation studies, terminology tasks (understand the concepts, definition, creation and use) is simultaneously (1) a crucial step, (2) something that takes some time to master, (3) often taken not so seriously as it should.

According to our experience, after learning the main concepts, students need to be involved in terminology projects.

In very compact way, we met projects reporting the use of several tools and formats. Following we mention some of them:

- Multiterm (terminologies, Trados, multiterm-xml);
- glossaries of CAT tools like SmartCAT [1];
- Text Encoding Iniciative (TEI), namely TEI dictionaries [2, 3];
- TBX (term base exchange format) [4];
- XDXF (dictionary exchange format) [5];
- stardict, stardict-editor [**?** ], using several formats;
- goldendict [6] - a dictionary lookup program;
- LaTeX + specific dictionary style – example: (lua)naterm.sty;
- several other non free tools (that we had some difficulties to study and even more, to share with the students);
- some academic tools (in many cases that we could not yet obtain);

We experiment with our students several approaches.

- We start with terminology created on Word (but the often good-looking results have very poor structure and use).
- We moved on to SmartCAT's glossaries[1] (Web based tool), where it is possible to define fields (standard or user-specific) – but it was difficult to obtain some advantages or output results, and the entries insertion process is painful.

*MDTT 2023*

- We tried MultiTerm[7] (some concepts are clear and elegant, but lots of problems with the tool (installation, Java incompatibilities, impossibility of using it outside Windows)[1]

In the end, we decide to build a specific terminology tool – NAT-term – to be used in teaching sceneries, to create, discuss terminologies, dictionaries and glossaries.

According to the indented scope and focus, we often end up hiding some NAT-term features and notation.

## 2. Design goals and features

NAT-term was build with the following features:

- Concept-based entries
- Textual input (it is possible to create a nat-term terminology with just a text editor)
- A set of tools to translate the terminology to a variety of output formats:
  - (1) PDF dictionary;
  - (2) xdxf [5] (XML format for dictionaries) – to be used with goldendict[6] (or similar) tools;
  - (3) (multi-file) HTML site;
  - (4) other formats planned [4].
- Transformation (nat-term terminology → output formats ) available through a command line script (for programmers) and through a very simple web interface (to be used with no installation effort and dependencies)
- Rich micro structure type of attributes: (1) concept attributes; (2) conceptual relations – it is possible to define relation properties (ex: InverseOf) to get inference and linking; (3) attributes for terms; (4) term / concept attribute values – multimedia attributes, text, numbers, etc.
- syntax for macro-structure definition (multi-level domain trees; maximum level: 3)
- advanced features to (optional): (1) create entries based on tables; (2) external terminologies; (3) directives to control output style: `ignore` (hide a field), `rename` (change the output name of a field), `inline` (compact / reduce size).

## 3. NAT-term by example

### 3.1. NATerm format

Consider the following simple example, with just 2 entries (concepts):

```
%title Dictionary EN - PT                    # area of directives
%author JJoão                                #   metadata
%lang EN PT ES                               #     languages
%inv hpr hyp                                 #   relation properties

PT: gato                                     # Portugues term
+G: m                                        #   term atribute (gender)
EN: cat                                      # English term
hpr: animal                                  # conceptual ralation
```

[1] We had not yet the opportunity of working with the most recent versions of Multiterm.

```
PT: cão
+G: m
EN: dog
ES: perro
def: ....                                   # concetual textual atribute
!img: dog.jpg                               # multimedia atribute
hpr: animal
```

In order to create a NATerm terminology, it is only necessary a text editor:

- Textual format;
- follows a similar approach to *markdown* or *wikis*;
- `%...` - directives, metadata;
- entries (concepts) are separated by empty-lines.

In order to build different output formats, we can use:

- Command line (needs installation of naterm, and optionally LaTex, GoldenDict)
  `naterm -html f.naterm`
  ... creates "f.html"
- or simple web interface (no installation needed)
  ... uploads "f.naterm" and downloads "t.pdf", "f.xdxf", etc

### 3.2. Micro structure (fields, attributes)

One very important concept is the structure of each entry (concept) – a very simple atribute tree:

- Concept oriented
- 2 level field:
  - Language term ( EN: cat ):
    - sub-fields of term ( +Gender: masculine )
  - Concept relations ( HPR: animal )
  - data properties ( def: is the second month of the year):
    - sub-fields (+src: Wikipedia)
    - attribute values – text, number, multimedia,
  - Images (!img: ....)
  - Sound (!snd: ....)
- relations and properties (ex: InverseOf)

### 3.3. Directives and Metadata

:

- Metadata:
  - title, author, date
- Introduction, Appendices, (`%pre  %pos`)
- Visual configure (`%inline, %ignore, %rename`)
- Language (`%lang  %rellang`)
- Relation properties (`%inv`)
- several terms obtained by inference (transportes, TGV, ...)

### 3.4. Macro-structure Domain trees

Naterm provides notation for Domain Tree (navigation, ...) – see Fig.1, making it possible navigate both top-down and bottom-up:

- top-down:
  - TOP *subdom* animal
  - animal *subdom* mammal
  - mammal *subdom* feline
  - feline *voc* {lion, cat, ...}
- bottom-up:
  - {lion, cat, ...} *dom* feline
  - feline *supdom* mammal
  - mammal *supdom* animal
  - animal *supdom* TOP

InverseOf:

- subdom $\longleftrightarrow$ supdom
- dom $\longleftrightarrow$ voc

Domains may have 3 different levels "==", "===" and "====".

### 3.5. Tables based entry definition (CSV and similar)

Consider the following example:

```
PT: $1
EN: $2
hpn: $3
*tab{
árvore :: tree :: bananeira | palmeira | carvalho
fruto :: fruit :: maça | banana | uvas
bebida :: drink :: vinho | água | chá | café
flor :: flower :: rosa | girassol | lírio
}
```

Notes:

- For each line in a csv-table :
  - split by '::'
  - define a new concept-entry
  - field 1 — `$1`
- tables:
  - external: `* tab(food.csv)`
  - inline: `* tab{ line1 ....}`

Tables are often good to describe repetitive homogeneous structures.

## 4. Conclusions

We would like to emphasize the importance of (1) Building real projects helps with terminology concepts comprehension. (2) NAT-term authoring syntax (or subsets of it) helps clarifying ideas

```
%tit Polish : Engl
%author Sylwia Foryńska
%lang EN PL PT
%rellang EN
%inv dom voc
%inline G

==  animals

===  birds
====  bird of prey

EN: eagle

===  mammals
====  canines
====  felines

EN: lion

===  amphibians
===  reptiles
===  insects

EN: ant

===  crustaceans
===  mollusca
===  fish
```

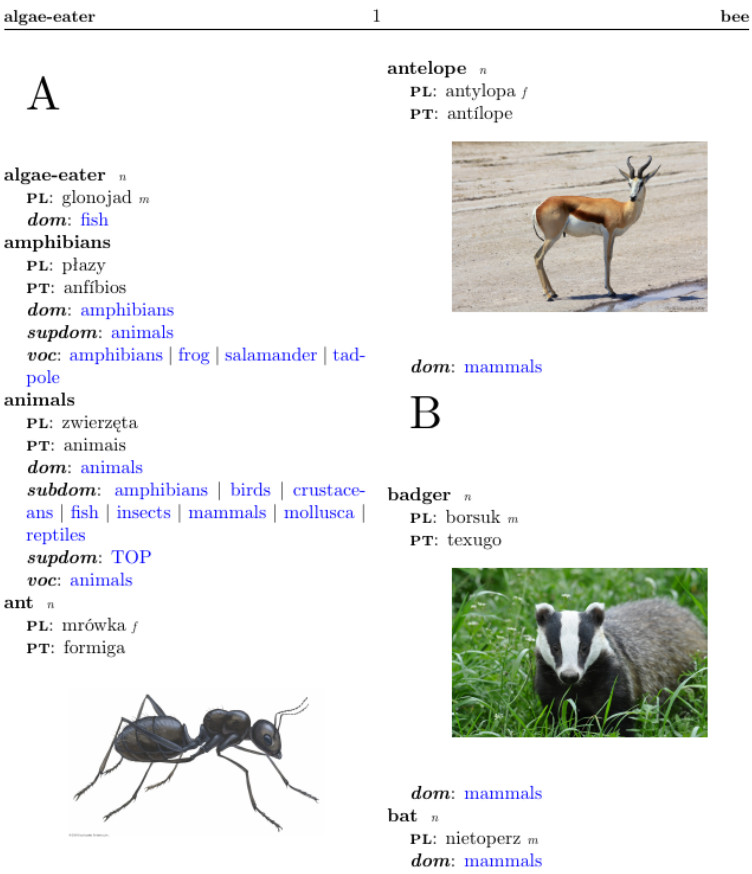


**Figure 1:** Naterm domain tree elements(left), and an excerpt of the generated PDF (right).

(ex: inverseOf completions achieves good results but fails miserably if the concepts are not well organized). (3) Good-looking output encourages the students. (4) textual syntax approach.

### Future work

- Improve HTML output:
  - options for:
    - multi page, multi lingual, ... site
- webservice
- better error messages, and documentation

## References


[1] smartcat, SmartCat Glossary support tool, CAT tool terminological module, smartcat, 2017.

[2] E. Vanhoutte, An Introduction to the TEI and the TEI Consortium,

Literary and Linguistic Computing 19 (2004) 9–16. URL: http://llc.oxfordjournals.org/cgi/content/abstract/19/1/9. doi:10.1093/llc/19.1.9. arXiv:http://llc.oxfordjournals.org/cgi/reprint/19/1/9.pdf.

[3] TEI Consortium (Ed.), TEI P5: Guidelines for Electronic Text Encoding and Interchange, Version 2.0.1 ed., TEI Consortium. http://www.tei-c.org/release/doc/tei-p5-doc/en/html/DI.html (January 2012), 2011.

[4] termbase, Introduction to TermBase eXchange (TBX), XML-based format, LISA, 2019. ISO 30042, https://www.tbxinfo.net/, definition available on TTT.org.

[5] S. Singov, L. Soshinskiy, XDXF dicionary exchange format, XML-based format definition, wikipedia, 2022.

[6] goldendict, GoldenDict a dictionary lookup program, tool, 2015. http://goldendict.org.

[7] multiterm, MultiTerm technical data sheet, tool, Trados Studio, RWS, 2022. https://www.trados.com/products/multiterm-desktop.