UNIVERSITE DE TECHNOLOGIE DE COMPIÈGNE
Département de Génie Informatique

MOSS v7 - Multilinguism

Jean-Paul Barthès

BP 349 COMPIÈGNE
Tel +33 3 44 23 44 66
Fax +33 3 44 23 44 77

Email: barthes@utc.fr

N234
December 2008

Warning

This document describes the use of multilingual data for defining ontologies in the MOSS system.

Keywords

MOSS, ontology, multilinguism

Revisions

Version	Date	Author	Remarks
1.0	December 08	Barthès	Draft

Contents

1 Introduction

When developing the SOL representation for building the Terregov ontology, we had a requirement to accommodate several languages, namely, French, English, Italian and Polish. Because MOSS had been developed in the context of a single language (English), it has to be modified to support other languages.

Thus a new datatype was introduced, the multilingual name abbreviated as mln. The idea was to be able to use a single piece of data for representing the same text in different languages. At the same time a global variable *language* would specify which language to choose from. In addition we felt the need to represent synonyms or equivalent expressions in a given language, a possibility that we added to the representation.

A library of elementary functions was developed and used to integrate the new feature to MOSS.

The resulting possibilities have been used in the Terregov project but also in the OMAS multi-agent platforms, where we can have Personal Assistant agents speaking different languages, although using the services of the same multilingual Service agents.

2 Multilingual Names

A multilingual name has been defined as a list containing language tags, formally:

```
<mln> ::=  ({:name} {<language tag> <terms>}+)
<terms> ::= "<term> {; <term>}"
```

where curly brackets indicate that the term is optional and + indicates one or more occurrence of the expression and term represents any combination of words.

Example

The following expressions are multilingual names:

```
(:name :en "Person")
(:name :en "Person; Fellow")
(:name :en "Person; Fellow" :fr "Personne; Quidam")
(:en "Person" :fr "Personne" :it "Persona")
(:fr "Personne; Quidam")
(:en "Alice in Wonderland" :fr "Alice au pays des merveilles")
```

Person and Fellow are here considered as English synonyms, Personne and Quidam as French synonyms.

3 Function Library

Several functions have been created for dealing with multilingual names (Table 1) and synonyms (Table 2).

4 Using Multilingual Names

Using multilingual names is sometimes tricky. Some ontologies do not have entries for all languages for a number of reasons. Another difficult point is for creating or using objects in a multilingual context. This is detailed in the next two sections.

Table 1: MLN functions

%mln?(expr)	type check
%mln-add-value(mln value language-tag)	add a language tag and value
%mln-equal(mln1 mln2 &key language)	test for equality
%mln-extract (mln &key (language *language*) canonical)	extract language string
%mln-extract-all-synonyms(mln)	gets all synonyms
%mln-extract-mln-from-ref (ref language &key no-throw-on-error (default :en))	builds a language mln
%mln-filter-language (mln language-tag &key always)	extract synonym string
%mln-get-canonical-name (mln)	extract a canonical name
%mln-get-first-name (name-string)	extract first synonym from a string
%mln-member (input-string tag mln)	check for membership
%mln-merge (&rest mln)	merge several mlns
%mln-norm (expr)	remove the :name tag
%mln-print-string (mln &optional (language-tag :all))	print an mln nicely
%mln-remove-language (mln language-tag)	remove language entry
%mln-remove-value (mln value language-tag)	remove a value
%mln-set-value (mln language-tag syn-string)	set language tag and value
%multilingual-name? (expr)	same as %mln?

Table 2: Synonym functions

%synonym-add(syn-string value)	adding a synonym
%synonym-explode (text)	extract all synonyms
%synonym-make (&rest item-list)	makes a synonym string
%synonym-member (value text)	membership test
%synonym-merge-strings (&rest names)	merge synonyms
%synonym-remove (value text)	removes a synonym

4.1 Inputting Multilingual Data

Two cases occur when inputting data into MOSS: (i) all data use the same language; of (ii) data are in different language. In the latter case two sub-cases can occur: (i) data are tagged with the corresponding language: or (ii) data are mixed and not tagged.

4.1.1 Inputting in the Same Language

Whenever the data uses the same language, it is not necessary to define mlns in the definitional macros of MOSS, although it is correct.

For example: writing

```
(defconcept (:name :en "Person")
    (:att (:en "name; family name") (:entry))
    (:att (:en "first name"))
    )
```

or writing

```
(defconcept "Person"
    (:att "name; family name" (:entry))
    (:att "first name")
    )
```

is equivalent, provided that in the second case the global *language* variable be set to `:EN`.

Note that the value of the global *language* variable will shadow any other language specified in the definition.

4.1.2 Inputting in Different Languages

If we want to specify several languages in the input data, then the global *language* variable should be set to the value `:ALL`. In that case we must tag the different strings. E.g.:

```
(defconcept (:name :en "Person" :fr "Personne")
    (:att (:en "name; family name") (:entry))
    (:att (:en "first name" :fr "prénom"))
    )
```

Note that the system will accept the data even if some tags are missing like in the example where the name attribute has no French value. However, it is an error to give simple strings.

The :unknown tag: There are some cases however when the data is multilingual but cannot be tagged because it comes from an external source. If we want to include the new data in a multilingual environment, then we must add a tag to the language. The special tag `:unknown` can be declared as the value of the global *language* variable.

4.2 Accessing Data

Accessing data is done with regard to the value of the global *language* variable.

The solution we adopted is the following: if a name is requested in a specific language, then we try to obtain it using the specific language tag. However, if it is not present, then we try to get the English name for it. If the latter is not present, we try to find an `:unknown` tag. If we cannot find it we take a value choosing randomly from the languages that are recorded.

4.3 Legal Languages

As mentioned previously languages are specified by the use of (standard) tags. The list of languages recognized by MOSS is given by the value of the global *language-tags* variable.

5 Miscellaneous

Several points should be considered when using multilingual data:

- Most tools cannot accommodate languages other than those that can be expressed in ASCII.
- It is recommended to encode all strings using UNICODE (e.g. UTF-8). Now, if this option is selected, UTF-8 encoded files should be loaded using the external-formal option.
- One should be careful when specifying the input language when creating MOSS concepts.