Experiences in Reusing Knowledge sources using Protégé and Prompt. An application in a medical domain

María Taboada
University of Santiago de Compostela. Spain.

Introduction

• Goal:
  – Modelling a Knowledge Base (KB) in a clinical domain
  
  Conjunctivitis

• Guideline American Academy of Ophthalmology (AAO)
  • http://www.aao.org/aao/education/library/ppp.cfm

Development of the domain ontology

• We have chosen PROMPT for several reasons:
  – It allows us to merge source ontologies into a resulting ontology
  – It analyses source concepts, properties (including restriction on value properties) and relationships
  – It is interactive with the user, allowing us to accept or reject the suggestions

Development of the domain ontology

• Three types of conflicts identified by PROMPT:
  – Multiple paths from a same concept
  – Not defined concepts exist
  – Different defined types for a same concept
Experiences and Difficulties using Prompt

- PROMPT automatically identifies identical concepts from different sources for merging.

- However, if the concepts are not identical, it is necessary to manually identify them.

- So, the reuse time could be reduced if PROMPT uses a synonym dictionary.

- PROMPT identifies dangling references and, for each of them, it makes a suggestion:
  - Copying a concept that does not exist
  - If you accept the suggestion, PROMPT automatically copies the concept
  - But if you reject it, PROMPT does not facilitate any operation
  - We identified two possible operations:
    - Removing the property that made the dangling reference
    - Changing the restriction on the property values

- We have detected that we made some operations repeatedly.
  - Example
    - Each time PROMPT identified the conflict ‘Multiple paths from a concept’
      - We removed all parents relative to the most external nodes in the hierarchy
  - An improved alternative:
    - Configuring operations when a conflict is detected
    - Take into account the particular heuristic of the each case.

Specialisation of the core ontology
Experiences and Difficulties using Protégé

- Meta-classes provides:
  - We are able to classify concepts from different sources in the same medical domain hierarchy
  - At the same time, we conserve the origin of the information.
  - We can easily distinguish between UMLS terminological information and essential properties of a class or instance
  - We do not overwhelm the user with excessive information
  - The maintaining of the KB is simplified

Experiences and Difficulties using UMLS

- We searched concepts using
  - The web based interaction tools provided by the UMLS Knowledge Source Server (UMLSKS)
  - The UMLS plug-in
- With regard to the search process
  - This way of displaying the information is more clear using the UMLS plug-in
  - The relevant information is located more quickly
  - The way of displaying the ordered list of search results and the narrow tree of a concept is more clear

Experiences and Difficulties using UMLS

- With regard to the import process
  - In Protégé it is quick
    - Selecting the concept to be imported
    - Selecting the place (class or instance) in the KB where the information will be imported
    - The plug-in automatically creates all slots and fills them with the search results.
    - You can directly import not only the searched concept, but also any concept in the narrow tree.

Experiences and Difficulties using UMLS

- With regard to the import process
  - The UMLS tab prevent us from importing the information directly into our UMLS Mapping class
    - The information can be imported as a class or instance
    - But, the definition of the imported slot is predefined
    - The place (class or instance) where the information will be imported is also predefined
    - This definition can be inadequate for all applications.

Experiences and Difficulties using UMLS

- For example, the cui slot is a single string and the slot semantic type is a multiple string.
- We decided to model the string cui separately from the class representing the UMLS information
  - The cui slot is an instance of the CUI subclass of the Unique Identifier class
  - We can verify whether an identifier is unique or not, by defining a single PAL constraint
- We represent the slot semantic type as an instance of the Semantic Type class.
- In our KB, Metasaurus concepts and Semantic types are related to each other.

Experiences and Difficulties using UMLS

- There is no means of selecting the information to be imported
  - It could be of interest to import only some slots.
THANK YOU
VERY MUCH!!!