

ANTLR - translator CIM2OWL

Wprowadzenie

Przedmiotem zadania będzie translacja formatu [CIM](#) do [OWL](#) z wykorzystaniem generatora parserów [ANTLR](#).

CIM, czyli Common Information Model, jest zorientowanym obiektowo modelem zarządzanych obiektów. Te zarządzane obiekty są odzwierciedleniem dowolnych zasobów systemowych. CIM jest wykorzystywany do określania sposobu wymiany danych pomiędzy aplikacjami zarządzającymi (np. [WBEM](#)). CIM Schema jest formalizacją tego modelu i może być definiowany przy pomocy Managed Object Format (MOF). W ramach tego laboratorium pod uwagę będzie brany [podzbiór MOF](#).

Z kolei OWL, tj. Web Ontology Language, jest językiem ze składnią opartą na XML, a semantyką opartą na tzw. description logics. Służy do reprezentacji i przetwarzania danych w sieci WWW. OWL umożliwia opisywanie danych w postaci ontologii i jest językiem stworzonym do celów opisu Semantic Web.

Więcej informacji można znaleźć pod następującymi adresami:

- Common Information Model (CIM) Standards
<http://www.dmtf.org/standards/cim/>
 - [CIM Schema 2.10](#)
 - [Managed Object Format \(MOF\)](#)
 - [Windows Management Instrumentation](#), zakładki Common Information Model, Managed Object Format
- OWL Web Ontology Language <http://www.w3.org/TR/2004/REC-owl-features-20040210/>
 - [Web Ontology Language on WIKI](#)
 - [Przegląd języków i metod do opisu ontologii](#)
- ANTLR (ANother Tool for Language Recognition) parser generator
<http://www.antlr.org/>
 - [Introduction to ANTLR](#)
 - [ANTLR grammars](#)
 - [The cim22Grammar v0.2 Package](#). "A CIM parser by Pedro Assis in Oporto. A very nice write-up too. Based on Antlr 2.7.1 and it targets CIM based MOF models. This is full compliant with the CIM present specs (2.2) and it's working for all DMTF models. Includes the grammar file, the java main file, and some ms-dos batch file. Also I've included a mini-howto addressing CIM, ANTLR and the package itself."

Opis zadania

Poniżej opisany jest minimalny podzbiór MOF do obsłużenia w zadaniu. Poniżej opisany jest minimalny podzbiór MOF do obsłużenia w zadaniu. Przyjmujemy, że:

- obsługujemy klasy
- obsługujemy dziedziczenie
- obsługujemy atrybuty proste (datatype)
- nie obsługujemy wartości domyślnych
- nie obsługujemy wektorów
- nie obsługujemy metod
- obsługujemy qualifier Description w najprostszej postaci, tzn. bez ValueMap, Values
- obsługujemy qualifier Association, Aggregation, Composition

Translator CIM2OWL tworzony będzie w dwóch etapach:

- **cz. 1** to zapoznanie z narzędziami i formatami meta opisu oraz przedstawienie własnych pomysłów na tłumaczenie w formie krótkiej dokumentacji. Minimalnie wymagany jest opis podzbiór składni MOF, który został wybrany do tłumaczenia i proponowany sposób tłumaczenia, tj. odpowiadający mu zbiór konstrukcji OWL. Ponadto w dokumentacja mogą się znaleźć np. przyjęte założenia, napotkane problemy, sposoby ich rozwiązania lub trudności z ich rozwiązaniem, znalezione ciekawe materiały. Zaliczenie cz.1 wymaga przesłania wstępnego projektu w formacie ASCII lub pdf.
- **cz. 2** obejmuje implementację translatora w Javie z wykorzystaniem ANTLR. W dołączonym do kodu źródłowego readme powinien się znajdować opis wykonanego narzędzia. Translator będzie testowany przy pomocy zbioru plików MOF należących do [Core Schema](#). Zaliczenie cz. 2 wymaga przesłania kodu translatora wraz z krótkim readme zawierającym opis wykonanego narzędzia, minimum jest to: lista przysłanych plików, dokładny opis sposobu kompilacji i uruchomienia, liste zmian w stosunku do cz. 1. Proszę o sprawdzenie poprawności wygenerowanej ontologii przy wykorzystaniu edytora ontologii, np. Protege <http://protege.stanford.edu/>, SWOOP <http://www.mindswap.org/2004/SWOOP/> lub reasonera.