Enabling complex queries to drug information sources through functional composition

Olivier Bodenreider
Lister Hill National Center
for Biomedical Communications
Bethesda, Maryland - USA

Lee Peters, Jonathan Mortensen, Thang Nguyen
Motivation

- Biomedical vocabularies are used as reference knowledge sources
- They can be used programmatically through Application Programming Interfaces (APIs)
  - Unified Medical Language Server (UMLS)
  - BioPortal
  - RxNorm
- Difficult to compose complex queries across multiple APIs
Example use case

- Find all the brand name products available for a given generic drug, whose identity is known by the FDA unique ingredient identifier code (UNII)

- Workflow

   1. Find FDA UNII (ingredient)
   2. Get RxNorm ID (ingredient) using `findRxcuiById`
   3. Get RxNorm ID (branded drug) using `getRelatedByType`
Objectives

◆ To enable an end-user to create complex queries to drug information sources through functional composition

◆ Application to two different web services to drug information sources
  ○ RxNorm API
  ○ NDF-RT API – National Drug File-Reference Terminology

◆ Complex query = workflow
  ○ Sequence of web service functions
  ○ Function interoperability driven by an ontology
Web service annotation

- Several web service annotation and composition frameworks
  - SAWSDL (Semantic Annotations for WSDL and XML Schema) – W3C
  - SSWAP (Simple Semantic Web Architecture and Protocol) – U. Arizona
  - SADI (Semantic Automated Discovery and Integration) – U. British Columbia
  - Taverna – U. Manchester

- Our approach
  - Difficult to retrofit our services to these frameworks
  - Use of an ontology in the background, not in the payload
Drug information sources

- **RxNorm**
  - Developed by NLM
  - Normalized names and codes for drug entities
  - Relations among drug entities form a graph

- **NDF-RT**
  - National Drug File-Reference Terminology
  - Developed by the Veterans Health Administration
  - Generic drugs described in terms of their active ingredients, mechanisms of action, physiologic effects, and therapeutic intent (indications and contraindications)
Methods

- Problem: Determine and represent interoperability among RxNorm and NDF-RT API functions based on their input and output
- Semantic representation of input and output
  - Captured in an OWL ontology
- Definition of interoperability between two functions
  - Modeled through rules
Ontology of functions

- Developed in OWL using Protégé
- Main classes
  - Service (e.g., RxNorm, NDF-RT)
  - Function: input and output parameters
  - Function parameter: semantics of identifiers
- Examples
  - RxNorm API has_function findRxcuiById
  - findRxcuiById has_input id_type
  - findRxcuiById has_input id
  - findRxcuiById has_output RxCUI
  - RxNorm API has_function getRelatedByType
  - getRelatedByType has_input RxCUI
Rules of interoperability

◆ Main rule – If the input of B matches the output of A, then A is potentially interoperable with B

◆ Additional rules
  ● Interoperability between generic IDs and specific IDs in the input or output of functions

◆ Example
  ● A: rxnorm:findRxcuiById has_output RxCUI
  ● B: rxnorm:getRelatedByType has_input RxCUI
  ● input of B matches the output of A
    => rxnorm:getRelatedByType interoperable_with rxnorm:findRxcuiById
Implementation

- The OWL ontology supports the interoperability rules
- Interoperability relations inferred from the rules support the determination of interoperability for a given pair of functions
- The ontology and the interoperability relations drive
  - The validation of workflows
  - The graphical interface for creating workflows
    - Only allowable functions are proposed in the menu
Simple example workflow: Finding brand names from clinical drug strings

- Original input: a clinical drug string
  - citalopram 20 mg oral tablet
- Expected output: brand names
  - Celexa

Workflow

- clinical drug string -> clinical drug RxCUI
  - findRxcuiByString()
- clinical drug RxCUI -> branded drug RxCUI (+ name)
  - getRelatedByType()
RxMix (beta version)

Create applications from RxNorm, RxTerms, and NDF-RT APIs

No Workflow Defined

OR

Select Function
No function select

http://mor.nlm.nih.gov/RxMix/
Add first function

WORKFLOW

No Workflow Defined

LOAD CONFIGURATION

OR

Select Function

findRxcuiByString

Optional Parameters

source_type:

allSourcesFlag: 0

searchType: 0

Add to Workflow
Add second function
Complete workflow
Add input and run

INPUT

name: citalopram 20 mg oral tablet

OUTPUT

Output Filter
- RXCUI
- UMLSCUI
- name
- term_type

Output Format
- TABLE
- XML
- JSON
- TEXT

EXECUTE

Clear  Run

Run the Workflow (Run)
<table>
<thead>
<tr>
<th>name</th>
<th>RXCUI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Celexa</td>
<td>215928</td>
</tr>
<tr>
<td>---</td>
<td></td>
</tr>
</tbody>
</table>
Examples of use cases

- Finding clinical drugs which may cause allergic reactions
- Finding interactions to RxNorm clinical drugs
- Finding NDF-RT ingredients from RxNorm clinical drugs
- Finding VA classes for RxNorm clinical dose forms
- Finding brand names from clinical drug strings

Library of common use cases integrated in RxMix
Summary

- Web service composition model for the RxNorm and NDF-RT APIs
- Enables an end-user to create complex queries to drug information sources ("workflows")
- Our model supports common use cases, including checking interactions for RxNorm drugs and deploying allergy lists defined in reference to drug properties in NDF-RT
Medical Ontology Research

Contact: olivier@nlm.nih.gov
Web: mor.nlm.nih.gov

Olivier Bodenreider
Lister Hill National Center for Biomedical Communications
Bethesda, Maryland - USA