KDI : Knowledge and Data Integration

FHIR - Fast Healthcare Interoperability Resources

KDI Final Presentation

Contributors

- Data Scientist: Sander Martins Gonçalves
- Data Scientist: Zuhairia Ibnat
- Domain Expert: Jacopo Mocellin
- Knowledge Engineer: Jacopo Mocellin
- Knowledge Engineer: Shaun McNaughton
- Project Manager: Shaun McNaughton
- Tutor1: Simone Boccca
- Tutor2: Mauro Dragoni

Table of Contents

Table of Contents

Project description

FHIR is a standard describing data formats and elements for exchanging electronic health records (EHRs).

EHRs are systems that allow to store all medical data concerning a patient in a digital format. While innovation could allow increased portability across different health organizations, current EHR implementations offer limited transferability to other systems because of several layers of obstacles. This problem is formalized as a problem of data interoperability.

After looking at available health records, our goal was to integrate three sets of synthetically generated health records of patients from different countries and healthcare providers in order to run a specific set of queries on them.

Table of Contents

Resources

In order to do the integration, data had to be collected, maintained, operated on using a suite of different tools, both suggested in the course and out of the course.

- Knowledge resources
- Data resources
- Metadata

In particular, a lot of resources were spent in generating the synthetic EHRs. Some of these will be discussed in further detail in the problems and solutions section.

Knowledge Resources

FHIR reference ontology

- good for understanding the the logic of the FHIR standard, but extremely vast and complex, beyond what is needed for our project

yED

- A tool to visualize and communicate the informal modelling of an ontology, in a *potentially* collaborative way

Protégé

- Effective and scalable ontology editor. Note: we used the desktop version.

Data Resources

Synthea

- Generation: Python
- Filtering: Python (KNIME and RapidMiner didn't work properly)

EMRBots

- Generation: It's already done
- Filtering: KNIME

Smart

- Generation: Python
- Filtering: Python, KNIME and RapidMiner didn't work properly

Table of Contents

Problems and Solutions

During the course of this project, many issues were encountered and there were many challenges around working as a team over great distances. While these aren't all of the issues that we encountered over the course of the period. These are some of the most crucial.

Issues - What data to use?

Accessing data

- Restricted access
- Real Electronic Health Records cannot easily be obtained, so they had to be synthesised.
- Limitation in the data generation process

EMRBots and **SMART** - No possibility to change data
Synthea - Some settings regarding to the region wasn't working

Issues - Time consumption

Synthea

- Initially: all information per patient file
- Merge all files (Rapidminer, KNIME, Python) - thought it would easier for Karmalinker - didn't work
- Later: all patients divided per section (Allergy, Immunization..)

EMRBots

- Huge file (KNIME) - time spent trying to merge into one file - thought that would have gain in terms of integration later

Smart

- Messier dataset
- One file solution didn't work

Issues- Reference schema

Size of FHIR reference

- FHIR ontology is extremely large.
- We adapted certain of its classes to be attributes in order to make things work more easily.
- Etypes inferred starting from the available datasets, in order to only consider concepts actually needed

Issues - tools

KOS

- Importing our ontology into KOS was troublesome
- Access from outside Italy
- no debugging solution when uploads fail - Tutors are the only resources
- bugs in the platform prevented correct recognition of defined classes and properties

Protégé and yed

- promised a web-based interface to allow cooperation, are in fact sub optimal solutions

Table of Contents

Outcomes

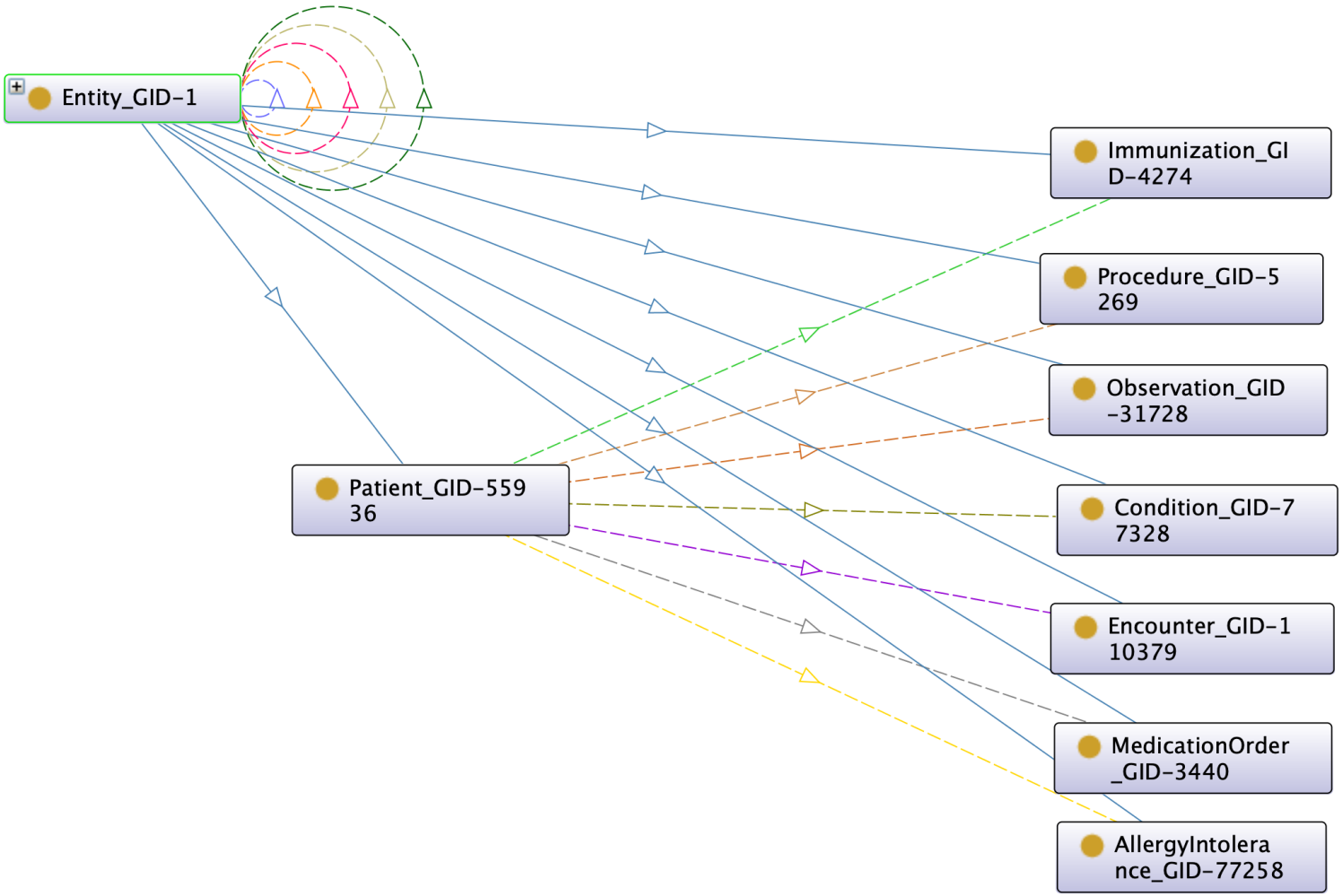

Figure: Final version of our ontology

Outcomes

An example from the DKG

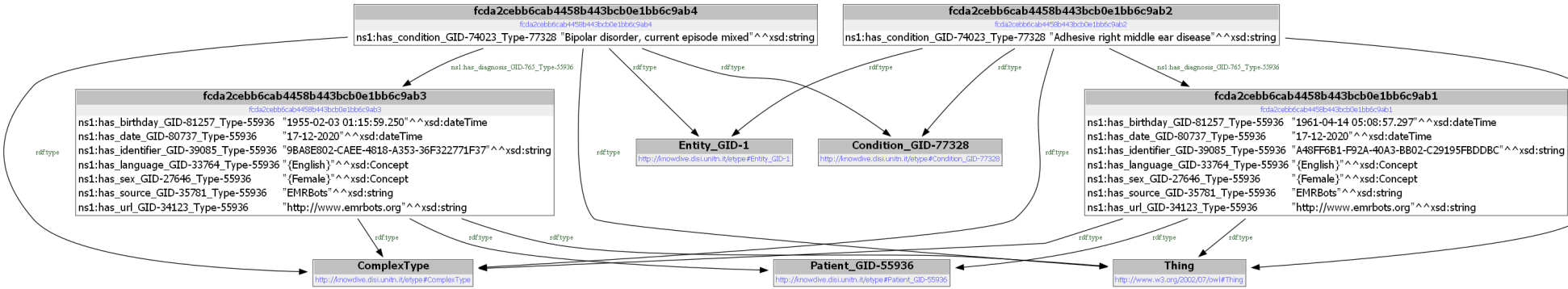

Figure: Example DKG using EMRBots information

Outcomes

The results from our integration can be seen as follows:

Metrics

- Coverage 0.06
- Flexibility 0.004
- Extensiveness 0.003
- Sparsity 0.94

Outcomes

- Unfortunately, it has been impossible to connect valid IRIs to the data integration already performed through Karmalinker in the short time we've become aware of the problem.
- SPARQL queries could therefore not be run through GraphDB

Table of Contents

Open Issues & Future works

- Future Work: Integration with projects of other domains: While this was one of the goals of the course, the various teams have been forced to disregard it in order to keep up with the pace of the development of their single projects.
- Open Issue: testing with more data if the defined ontology suits the representational needs

KDI : Knowledge and Data Integration

FHIR - Fast Healthcare Interoperability Resources

KDI Final Presentation