Dipartimento di Ingegneria e Scienza dell'Informazione

– KnowDive Group –

Fast Healthcare Interoperability Resources

Document Data:	Reference Persons:
- date -	Shaun McNaughton, Jacopo Mocellin, Sander Martins Gonçalves, Zuhairia Ibnat

KnowDive (internal) reports are for internal only use within the KnowDive Group. They describe preliminary or instrumental work which should not be disclosed outside the group. KnowDive reports cannot be mentioned or cited by documents which are not KnowDive reports. KnowDive reports are the result of the collaborative work of members of the KnowDive group. The people whose names are in this page cannot be taken to be the authors of this report, but only the people who can better provide detailed information about its contents. Official, citable material produced by the KnowDive group may take any of the official Academic forms, for instance: Master and PhD theses, DISI technical reports, papers in conferences and journals, or books.

Contents

1 **Knowledge Graph Codebook** 1
1.1 Knowledge Graph general description . . . 1
1.2 Data level . . . 1
1.2.1 Datasets general details . . . 1
1.2.2 Datasets metadata documentation . . . 1
1.3 Ontology level . . . 2
1.3.1 Ontology general details . . . 2
1.3.2 Ontology metadata documentation . . . 2
1.4 Knowledge Graph Evaluation . . . 3

2 **Knowledge Graph Development Process** 4
2.1 Scope Definition . . . 4
2.2 Personas . . . 5
2.3 Scenarios . . . 6
2.4 Inception . . . 7
2.4.1 CQs definition . . . 7
2.4.2 Initial Datasets description . . . 9
2.4.3 Datasets generation method . . . 10
2.4.4 Datasets metadata documentation . . . 11
2.4.5 Datasets collection process . . . 13
2.4.6 Inception level evaluation . . . 13
2.5 Informal Modeling . . . 14
2.5.1 Schema level . . . 14
2.5.2 Data level . . . 16
2.5.3 Informal Modeling Evaluation . . . 20
2.6 Formal Modelling . . . 21
2.6.1 Schema level . . . 21
2.6.2 Data level . . . 24
2.6.3 Formal Modeling Evaluation . . . 25
2.7 Data Integration . . . 26
2.7.1 Data integration operations and tools . . . 26
2.7.2 Variance respect Formal Modeling datasets . . . 33

Revision History:

Revision	Date	Author	Description of Changes
1.0	19/10/2020	Jacopo	Scope definition section written
1.1	21/10/2020	Jacopo	Inception section written
1.2	21/10/2020	Sander	Datasets generation method
1.3	21/10/2020	Shaun	General layout and editing
1.4	22/10/2020	Jacopo	Scope and inception Edits
1.5	22/10/2020	Zuhairia	Datasets Metadata Documentation written
1.6	22/10/2020	Shaun	Initial competency queries and scenarios added
1.7	23/10/2020	Sander	Metadata Synthea and Smart
1.8	23/10/2020	Zuhairia	Query edits
1.9	23/10/2020	Zuhairia	Synthea attribute table formation
2.1	05/11/2020	Jacopo	IM Schema level documentation draft
2.2	06/11/2020	Zuhairia	Metadata table formation
2.3	06/11/2020	Shaun	EER Diagram added
2.4	06/11/2020	Jacopo	IM Schema level documentation finalized+overall improvements
2.5	06/11/2020	Sander	Datasets management process
3.1	26/11/2020	Jacopo	Formal modelling schema section written
3.2	27/11/2020	Shaun	Added ontology visualisation, revised EER and lexical information
3.3	27/11/2020	Sander	Added Formal Modelling datasets management
3.4	27/11/2020	Sander	Added Variance respect Informal Modeling datasets
4.1	15/12/2020	Shaun	Added Formal Modelling Evaluation metrics
4.2	17/12/2020	Shaun	Filled in Knowledge Graph Codebook overview section
4.3	18/12/2020	Shaun, Jacopo,Sander	Finalised Project Report

1 Knowledge Graph Codebook

In this section we describe the Fast Health Interoperability Resources (FHIR) project, the aims of this work, what data was used and what the end result of the project was.

1.1 Knowledge Graph general 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 (which use different standards) in order to run a specific set of queries on them.

1.2 Data level

In order to run this we used three synthetically generated electronic health record datasets. These were generated according to some regional parameters in order to align it to the European context.

- EMRBOTS patient
- Synthea Patient
- SMART patient

These datasets were converted into a JSON format, with each dataset containing an extensive range of variables that one might associate with a health record. These are listed in the appropriate data section in the Informal Modelling Phase.

1.2.1 Datasets general details

The general details of metadata of the three datasets used can be found in our project folder recorded under the DCAT format.

1.2.2 Datasets metadata documentation

The process of generating the synthetic datasets was problematic

- EMRBOTS dataset (http://www.emrbots.org): The dataset was created by Uri Kartoun, PhD, in 2015. We are using the 100000-patients dataset, which comprises in total data about 100.000 patients, 361.760 admissions, and 107.535.387 lab observation. The dataset was generated according to the literature on patient population studies in order to create a realistic set of observations.

- Synthea patient generator DBs (https://github.com/synthetichealth/synthea): This tool was developed by the MITRE corporation and released freely under an Apache license. Data generated includes: Conditions, Allergies, Medications, Vaccinations, Observations/Vitals, Labs, Procedures, CarePlans ,Primary Care Encounters, Emergency Room Encounters, and Symptom-Driven Encounters.
- Smart on FHIR data generator (https://github.com/smart-on-fhir/sample-patients): This tools is provided as a test suite for SMART on FHIR applications by SMART itself. We are hoping to exploit it to generate structured data that fits the FHIR specifications about a population of patients.

1.3 Ontology level

In order to define the underlying Knowledge Graph ontology, we describe each step taken for each element at each level. Starting with the identification of competency queries to finally a specified ontology. This approach follows the iTelos methodology in order to create a high quality schema.

1.3.1 Ontology general details

We initially tried to adopt the FHIR ontology, but difficulties in trying to match only the entities that we were concerned with, meant we rebuilt a section of the ontology. This was done using data generated under the FHIR schema, meaning even though it was recreated, it was highly related to the FHIR such that things could be mapped easily.
From these datasets, we defined 8 classes representing the data we wished to integrate

- EMRBOTS patient
- Synthea Patient
- SMART patient

1.3.2 Ontology metadata documentation

The core and contextual entities were defined to be:

- Patient:[core] This entity defines the concept of a human patient. In our schema, it is the concept for which all competency queries are drawn through.
- Immunization: [core] An entity that defines an vaccination instance. This is a core concept for queries that are focused on retrieving a patient's past medical history.
- Condition: [core] defines a current or past condition. This is a core concepts for queries that are focused on retrieving a patient's past medical history.
- Allergies:[core] Defines current allergy or intolerance.This is a core concepts for queries that are focused on retrieving a patient's past medical history.
- MedicationOrder: [core] Defines a drug prescription administered by a physician. This is a core concept for queries involving past recommendations made by health professionals.

- Observation: [core] Defines lab observations and subsequent results and their associated units of measurements from testing.
- Encounter: [contextual] Defines an encounter with a health professional. The encounter may represent something like a consultation which may have an associated observation (e.g. a blood test).

The relationship between these entities can be seen in the EER diagram found in the Informal Modelling section. Broadly speaking, all the core entities are attributes of the Patient entity.

1.4 Knowledge Graph Evaluation

In order to evaluate our knowledge graph, we use a set of metrics to determine how much we have added to the overall schema.

The reference schema (FHIR) has 852 Classes and 3 Data Properties. Our schema has 8 Classes and 49 Data properties. With an almost complete overlap (where most of our data properties are classes in FHIR) we can calculate the following metrics:

Metric	**Value**
Coverage	0.06
Flexibility	0.004
Extensiveness	0.003
Sparsity	0.94

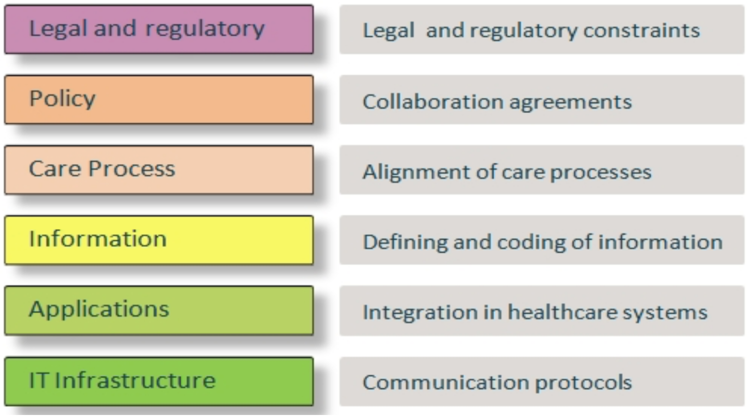
2 Knowledge Graph Development Process

The second chapter of this document aims to describe, in a detailed way, the KG development process. The sections below describe each phase of the KG building project, reporting for each phase, the description of the datasets and their evolution respect the previous phases, the schema construction which will generate the KG ontology in the end, as well as the description of the procedures adopted to manage the data and finally achieve those results. Moreover for each phase is reported an evaluation section, which aims to evaluate the quality of the results achieved at the end of each phase.

2.1 Scope Definition

In this part of the project report, we provide the foundational specifications for the further development of our data integration project. In particular, we are going to outline what is the problem our solution is going to address, why we chose that problem, what is the context in which the problem exists in, and why the iTelos methodology is the most suitable technical approach to solving the problem.

EHRs (Electronic Health Records) 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.



Legal and regulatory	Legal and regulatory constraints
Policy	Collaboration agreements
Care Process	Alignment of care processes
Information	Defining and coding of information
Applications	Integration in healthcare systems
IT Infrastructure	Communication protocols

Figure 1: (from eHN's Refined eHealth European Interoperability Framework)

While several layers of complexity need to be addressed in order to achieve complete interoperability, we are going to focus our efforts to implementing a solution that aligns semantic differences across different EHR systems, and therefore our intervention can be ascribed to the "Information" layer of this scheme. This layer is comprised of all aspects of the data model, including coding terminology and the formatting of the medium for transportation of the information.

The problem of incompatible data representations is particularly taxing in Europe, where, on one hand, free circulation of EU citizens across the 27 member states is fostered by a friendly regulatory scenario, but, on the other hand, seamless circulation of citizen's health data has been hindered by the lack of complete interoperability.

In the current European medical context, the eHDSI (e-Health Digital Service Infrastructure) connects eHealth national contact points allowing them to exchange two sets of health data: patient summaries and ePrescriptions. This system serves as reference in the field of interoperable EHRs.

Recently, the European Commission produced and adopted a recommendation on a framework for the further development of a European EHR exchange format that integrates what already accomplished by the eHDSI project. This consists of a set of common technical specifications and principles that will enable citizens to securely access and exchange their health data across borders in the EU.

In the annex to the Commission Recommendation on a European Electronic Health Record exchange format, a total of five types of data are defined as target of a European cross-border exchange solution:

- Patient Summaries
- ePrescriptions/eDispensations
- Laboratory reports
- Medical images and reports
- Hospital discharge reports

In line with this proposed framework, while selecting the types of data that our data integration system should be able to interpret, we opted for restricting our focus to those mentioned above in order to adhere with the requirements of a realistic EHR solution existing in the current European context described.

In this problem space, the iTelos methodology proves to be the most appropriate because it allows the alignment of information that in our case, coming from different source datasets, is scattered and represented in a diversified manner.

2.2 Personas

As a top level view, a solution that could help foster data interoperability could benefit several stakeholders in the healthcare panorama:

- Healthcare Facilities: could reduce costs and inefficiencies by reducing the amount of testing needed
- Healthcare professionals: could save time by not having to enter information by hand in a system. Could become able to interpret information in different languages. Could see the history of a patient: reaction to medications, previous operations, allergies, and other intolerances
- Patients: enables to consult a specialist or receive emergency treatment in another EU country
- Researchers: Could enable to recruit a diverse population for clinical studies while being able to access accurate retrospective reporting information about the patients.

Among these, we chose to focus on developing a solution that could fit primarily the needs of patients and healthcare professionals taken as a unique dyad of subjects cooperating to foster better outcomes in the therapeutic process

We will now illustrate some proposed personas and relative hypothetical use cases.

- Giannantonio, aged 35, is an Italian sales representative for an important clothing brand who is asked to travel frequently from one country to the other for work. His agenda is made up by appointments fixed with very little advance, which doesn't allow him to have much time to prepare luggage for his business trips or to bring extra paperwork along with him. He has good understanding of English, but not in a medical domain.
- Rita is an Italian woman, aged 73, who lives in a rural town near Caltanissetta. Once a year she travels to the Netherlands to visit her son and his family who emigrated there. She has no knowledge of foreign languages.
- Pedro is a Spanish man aged 45 who was recently diagnosed with Parkinson disease. He is travelling to Germany constantly in order to receive life-saving treatments. However, his clinical situation is complex since he suffers several co-occurring chronic conditions.
- Jean is a 52 year old, General Practitioner from France. He practices in a small clinic near Rennes, but is also working as a Forensic Medical Officer for the French police. While he loves both of his jobs, the additional stress from COVID-19 has been pressuring him to focus on a single job.

2.3 Scenarios

As the exchange of health information is an inherently complex and has widely varying needs, we use the personas and expand on them by creating a hypothetical scenarios to which having integrated health systems could be used.

Giannantonio: During one of his frequent trips to Greece, he is struck down by a heart attack. Not having any form of medical documentation with him, the medical team helping him needs to retrieve information elsewhere in order to know which treatments he might not respond well to. [Access to patient history]

Rita: During one of her visits to the Netherlands, Rita falls to the ground and breaks an arm. Even though her condition is not severe, the physician recommends close supervision by a colleague upon hospital discharge in order to avoid incorrect calcification of her fragile bones. Once back home, Rita is aware of the need to consult with her trusted physician in order to avoid complications, but would not be able to report to him any of the indications given by the Dutch doctor. On the other hand, her trusted doctor is unable to understand languages other than Italian. [Access to discharge reports data]

Pedro: Due to his complex medical conditions, the different medical teams who treat him in the two different countries need to be constantly updated on prior medications administered in order to inform their future clinical decisions. [Access to patient history]

Jean:Before seeing a new patient,Jean spends time reviewing that patient's medical history. He does this to better understand the patient at both a physical and emotional level. This in turn, allows him to mentally prepare for the consultation and to act in a professional manner. In doing so, he wishes to review at least any basic health information on the patient, for example age, blood type, and past medical history before seeing them. Now, he wishes to access medical records from an Italian hospital to help diagnose a patient who has recently arrived in France and has a history of hospitalisations. [Access to patient history]

2.4 Inception

This section is dedicated to the Inception phase description. Here are reported the initial definitions for CQs (Competency Queries), initial datasets collected and the relative metadata. For each of those elements the procedures and the tools adopted to achieve the results, have to be reported in the sections below.

2.4.1 CQs definition

In order to create the competency queries, we use the scenarios and persona descriptions provided in the previous section, to come up with possible queries that a person could want from the system.

The scenarios provide a set of actions for which a competency query can be modelled from. These actions and the mapping to the relevant competence query for each scenario is given below.
In future iterations, we might consider shrinking our focus to a narrower set of pathologies (and therefore queries) since our current dataset provide a data about patients with a broad and diverse set of conditions that could prove to be too extensive to represent in the final DKG.

Scenario	Question	Action	Description
Giannantonio	1.1	Access to known allergies	A list of known allergies and intolerances is returned
Giannantonio	1.2	Access to history of circulatory diseases	A list of current and previous conditions is returned.
Giannantonio	1.3	Access to lab reports	Results from latest blood analysis are returned
Giannantonio	1.4	Access to lab reports	Complete results from all blood analyses are returned
Giannantonio	1.5	Access to specific lab data	Latest measurement of MCHC RBC Auto-mCn is returned
Rita	2.1	Access to patient name	Name and last name of the patient are returned
Rita	2.2	Access to patient language	Desired spoken language is displayed
Rita	2.4	Access to known conditions	The system returns whether the patient suffers osteoporosis
Rita	2.5	Access to conditions history, other language	Previous Complete list of conditions is shown, in Dutch
Pedro	3.1	Access to patient allergies	The system shows whether the patient is allergic to sulfonamides
Pedro	3.2	Access to patient allergies	The system returns the complete list of recorded allergies
Pedro	3.3	Patient allergies, in different language	The system returns a complete list of allergies, in Spanish
Pedro	3.4	List of encounters	The system returns [illegible] by date and time

Scenario	Question	Action	Description
Pedro	3.4	Access to patient allergies	The system shows whether the patient is allergic to sulfonamides
Pedro	3.5	Access to lab reports metadata	Date and time of the latest blood exams is shown
Pedro	3.6	Data source	The database name from in which patient data is sourced is displayed
Jean	4.1	Access to demographics	The system returns the place of birth of the patient.
Jean	4.2	Immunizations history	The complete list of immunizations is shown.
Jean	4.3	Specific immunization	The system returns whether the patient has been vaccinated for Pneumococcal (PCV).
Jean	4.4	Specific immunization	The system returns whether the patient has been vaccinated for Hepatitis A (HepA).

2.4.2 Initial Datasets description

In this section are reported the metadata at datasets level involved in the Inception phase, so those metadata regarding the sources, the authors, the collection methods, and other meaningful information.

For our data integration process, we had to rely completely on synthetic data. The collection of genuine, real-world data is almost impossible because of the special privacy measures that are in place to protect this kind of personal data. This means that real-world data is often protected and needs some kind of authentication to be accessed. Nonetheless, we were able to find satisfying patient data generators by looking into the solutions developed to test other existing EHR systems. Specifically, we are going to provide a list of found datasets and their metadata here.

- EMRBOTS dataset (http://www.emrbots.org): The dataset was created by Uri Kartoun, PhD, in 2015. We are using the 100000-patients dataset, which comprises in total data about 100.000 patients, 361.760 admissions, and 107.535.387 lab observation. The dataset was generated according to the literature on patient population studies in order to create a realistic set of observations. More details about the data can be found on table 1 bellow.

- Synthea patient generator DBs (https://github.com/synthetichealth/synthea): This tool was developed by the MITRE corporation and released freely under an Apache license. The datasets generated with this tool can include data about over 90 different conditions thanks to its modular structure. Data generated includes: Conditions, Allergies, Medications, Vaccinations, Observations/Vitals, Labs, Procedures, CarePlans ,Primary Care Encounters, Emergency Room Encounters, and Symptom-Driven Encounters. This data is matched with Configuration-based statistics and demographics. The data can be formatted in unstructured formats or structured (JSON) ones. The latter also allows for formatting according to HL7 standards. The generator can be queried thanks to a Java-based interface.
- Smart on FHIR data generator (https://github.com/smart-on-fhir/sample-patients): This tools is provided as a test suite for SMART on FHIR applications by SMART itself. We are going to exploit it to generate structured data that fits the FHIR specifications about a population of patients.

Ultimately, our goal is to coherently aggregate this data in order to form the population of patients that is going to be represented in our DKG.

The possibilities were either to keep the datasets separate or to build a single dataset by merging the ones mentioned above, assigning to single patients from the EMRBOTS dataset (or a subset of it) data from the other datasets. We are aware that these datasets currently do not refer to the same population, this we are going to focus on retaining validity in the case of performing this task. In this case, the final result is going to be a single dataset containing very diverse and overlapping data on a set of patients. This data is going to be in different formats, therefore constituting the ground for usage of a DKG to align diversity.

2.4.3 Datasets generation method

EMRBOTS

EMRBOTS provided the method used for the creation of the dataset. The process starts by setting the desired configuration of the population. This means that is possible to set the percentage of gender, marital status, major language, ethnicity, date of birth and income level.
Once this pre configuration is made the next step is the generation of the individuals. For each generated patient, will be created a record that will be its lifetime electronic medical record (EMR). This record is created by randomly assigning the number of admissions, the length of each admission, the result of laboratory exams and the code for the main symptoms of the patient according to the list of International Classification of Diseases (ICD-10-CM). Laboratory results means a list with the measurements of levels of sodium, creatinine, or platelet counts, for example.

Synthea

Synthea is based on the PADARSER framework. This framework assumes that access to EHR real data is not available and therefore relies on public data to compose the population to be used for the generation of synthetic EHR. These public data are composed by aggregation of health incident statistics, clinical practice guidelines (used to develop caremaps) and medical coding dictionaries. These public will then feed the information and knowledge model in order to generate synthetic data that is composed by patient profiles, diseases and their prevalences, caremaps and coded narratives. Synthea uses Health Level-7 (HL7) the main codification system.

Variable and Category	**Patients (n=100,000)**
Mean age as of 1/1/2015, years (SD)	57.8 (17.3)
Gender (%)	
Female	52.0
Ethnicity (%)	
White	49.0
Asian	23.0
African American	15.0
Unknown	13.0
Mean number of admissions per patient, days (SD)	3.6 (1.5)
Mean length of stay (SD)	11.0 (5.2)
% Population with length of follow-up (years)	
0 - 9	13.1
10 - 15	9.3
> 15	77.6
Population below poverty (%)	21.6
Comorbidities; Prevalence (%)	
Malignant neoplasm	41.4
Rheumatoid arthritis	25.6
Diabetes (type I or II)	24.4
Renal complications	17.0
Coronary artery disease	7.0
Laboratory values (Mean; SD)	
Blood urea nitrogen (mg/dL)	17.5; 7.2
Platelets (k/cumm)	284.9; 95.3
Creatinine (mg/dL)	0.9; 0.2
Albumin (gm/dL)	4.2; 1.0
Lymphocytes (%)	25; 5.8

Table 1: Data setting - size 100.000 - EMRBOTS

2.4.4 Datasets metadata documentation

In this section are reported the new metadata at attribute level, describing the types, meanings, value-set, validity, and so on, of each variable within the initial dataset collected.
The dataset created by EMRBOTS has the following attributes:

- Patient ID: A unique number representing a particular patient
- Gender: The sex of a patient; either male or female
- Birth: The date of the birth of the patient
- Race: This describes the ethnicity of the patient. The categories are white, Asian African-American, unknown
- Marital Status: The categories of marital status are married, single, divorced, separated, widowed, and unknown

- Language: The preferable language of the patients are mentioned here
- Population percentage below poverty: This category is represented in percentage
- Admission ID: An admission ID for the patient
- Admission start date: The date the patient was admitted
- Admission end date: The date the patient was released
- Primary diagnosis code: This is an ICD10 code for the primary disease diagnosis
- Primary diagnosis description: The description of the diagnosis is mentioned
- Lab name: This category includes the tests of the patients. The CBC tests are: White blood cell count, Red blood cell count, Haemoglobin, Hematocrit, Mean corposcular volume, MCH, MCHC, RDW,Platelet count, Absolute Neutrophils, Absolute lymphocytes, Neutrophils, Lymphocytes, Monocytes, Eosonophils, Basophils. The metabolic tests are: Sodium, Potassium, Chloride, Carbon dioxide, Anion gap, glucose, BUN, creatinine, total protein, Albumin, Calcium, Bili total, AST/SGOT, ALT/SGPT, ALK PHOS.
  The urinalysis tests include: pH, specific gravity, red blood cells, white blood cells
- Lab value: This represents the numeric value of the tests
- Lab units: This attribute shows the unit of the lab tests
- Lab date: The date of the lab tests is shown in this attribute

The dataset created by Synthea has the following attributes: Information related to the patient:

- Patient ID: A unique number representing a particular patient
- Name: Corresponding name of the patient
- Gender: The sex of a patient; either male or female
- Birth: The date of the birth of the patient
- Race: This describes the ethnicity of the patient. The categories are white, Asian African-American, unknown
- Marital Status: The categories of marital status are married, single, divorced, separated, widowed, and unknown
- Address: Information related to the residence of the person. It is composed by Country and City
- Language: The primary language spoken by the patient
- Education level: Categorical variable indicating the education level of the patient
- Identification: Documents of the patient, could be passport, driver's id or other document
- Allergy: Information if the patient has any kind of allergy of food or medication
- Diagnostic: Information related to the identified disease with its related code (SNOMED CT format)

- Treatment: The treatment given to the patient for treating its condition
- Dosage: Information of the quantity of medicine was prescribed
- Period: Information related to the beginning and the ending of the admissions
- Immunizations: The code of the vaccine applied or NA in case it was not prescribed

Other variables related to the diagnostics or other information can be found in the repository's project folder DCAT specification.

The dataset created by Smart has the following attributes:

- Patient ID: A unique number representing a particular patient
- Name: Corresponding name of the patient
- Birth: The date of the birth of the patient
- Diagnostic: Information related to the identified disease
- Treatment: The treatment given to the patient for treating its condition
- Period: Information related to the beginning and the ending of the admissions

2.4.5 Datasets collection process

The collection process section aims to describe the collection methods adopted to obtain the initial datasets in this phase, as well as the tools used.

The collection has proven to be challenging for reasons mentioned earlier: Health data constitutes sensitive information and is therefore always protected with special provisions. Access to real-world health data most of the times is granted only via an accreditation process (As is the case for the MIMIC dataset, including data about real-world intensive care unit stays). After clashing with this bureaucratic obstacles, we started looking into free-to-use, open solutions available on the internet. We were finally able to find resources given by communities active in both academia and industry. The community behind the development of current EMR solutions has implemented testing datasets or dataset-generation tools that we eventually opted for using in our solution.

2.4.6 Inception level evaluation

We initially struggled with the initial concept behind what we were required, first to find data sources that were relevant and secondly to create a set of personas for which we could then create queries from. However, we can evaluate our performance of this stage of this phase by considering the following specific KPIs:

1. Number of datasets found: The current number of data sources amounts to 3, which is in line with the given indications.
2. Diversity of data collected: The number of datasets creates sufficient diversity in the data gathered. In fact, data collected is diverse in format (structured vs unstructured) and in representations (e.g. measurement units)

3. Coherence of data with purpose: Given the found data sources represent information in the medical domain, we are reassured that it fits coherently with the purpose of integrating medical information.

2.5 Informal Modeling

This section is dedicated to the Informal Modeling phase description. The Section is divided in Schema and Data level in order to report the details of the elements involved in the generation of the schema, as well as the description of the datasets evolution in this phase. Moreover each specific section, one for each level, reports the difference between the elements defined in this phase and the definitions in the previous phase, analyzing in this way the variance in the different phases.

2.5.1 Schema level

The schema level in this phase reports the first informal definition of the ETypes and of the Enhanced Entity Relationship (EER) model constructed using them. We further refine the set of competence queries in order to guide the construction of the EER and subsequent data filtering.

2.5.1.1 ETypes and EER Model definition

This section reports an informal definition of the ETypes involved in the datasets collected in the previous phase. This section includes a list of metadata/attributes associated to each of the elements generated and how they are related to each other. At this stage we are focussed on the relationships between our core entities required for our previously identified competency queries.

In this first, informal version of our EER model, we have taken care of including the fundamental entities that could represent the variables in our selection of the data and to address our competency queries.

Particularly, our initial core and contextual entities are:

- Patient:[core] This entity defines the concept of a human patient. In our schema, it is the concept for which all competency queries are drawn through.
- Immunization: [core] An entity that defines an vaccination instance. This is a core concept for queries that are focused on retrieving a patient's past medical history.
- Condition: [core] defines a current or past condition. This is a core concepts for queries that are focused on retrieving a patient's past medical history.
- Allergies:[core] Defines current allergy or intolerance.This is a core concepts for queries that are focused on retrieving a patient's past medical history.
- MedicationOrder: [core] Defines a drug prescription administered by a physician. This is a core concept for queries involving past recommendations made by health professionals.
- Observation: [core] Defines lab observations and subsequent results and their associated units of measurements from testing.
- Encounter: [contextual] Defines an encounter with a health professional. The encounter may represent something like a consultation which may have an associated observation (e.g. a blood test).

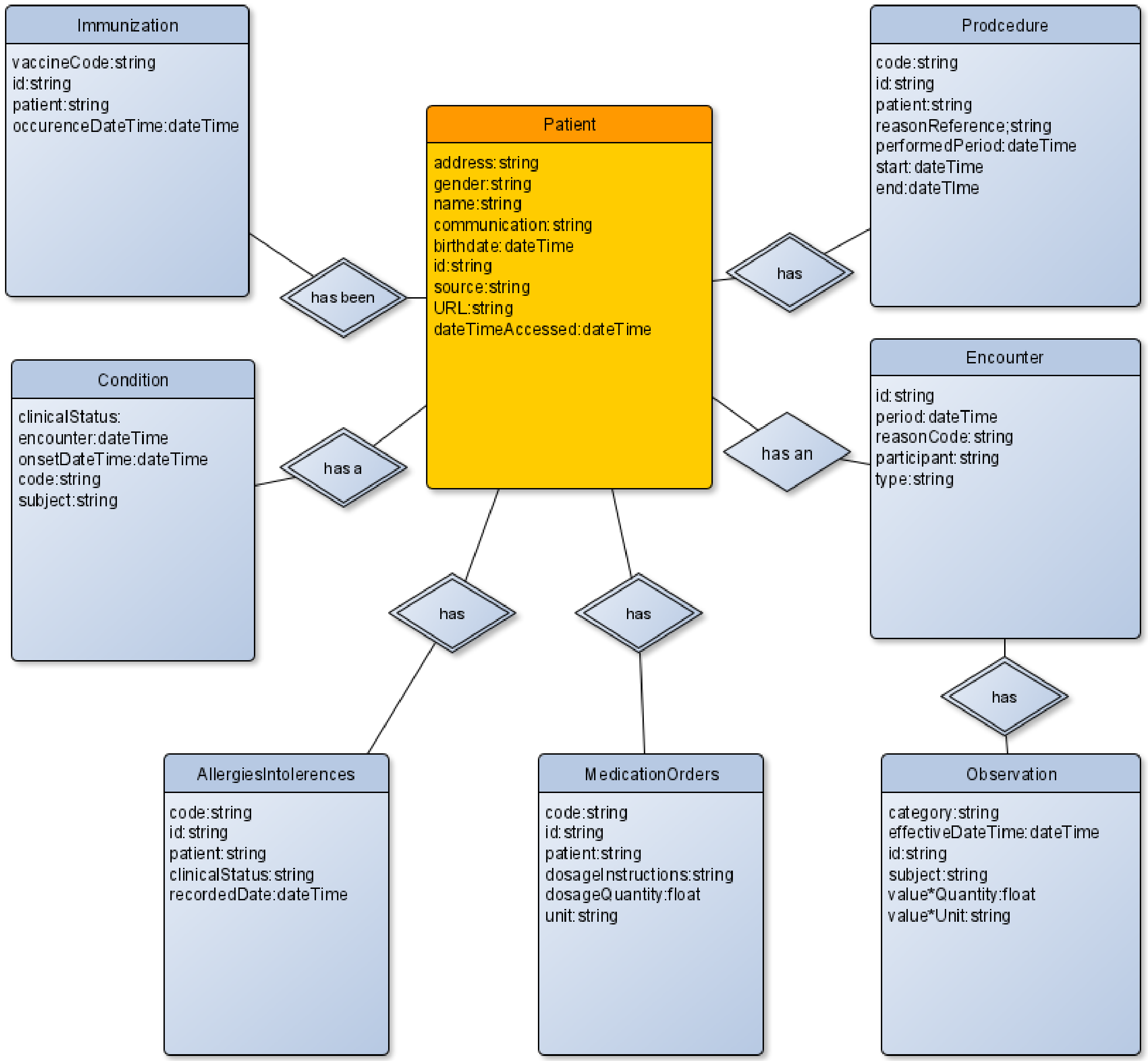


Figure 2: (FHIR Extended Entity Relationship Model)

Furthermore, we defined three more contextual entities in order to eventually represent the metadata about our sources of data within the EER. In this initial phase, we only limited ourselves to defining tentatively these entities and not the attributes contained.

- EMRBOTS patient
- Synthea Patient
- SMART patient

The contextual entities defined here are those that are required specifically for our particular competency queries. On the other hand, core Etypes are those collected from our reference ontology, that is the FHIR standard specification.

Along with these, we sourced complemented our EER with Etypes from general-purpose reference ontologies in order to define our common entities and therefore ensure maximum compatibility with other schemas and ontologies. These entities have also been temporarily defined in order to account for the integration with other projects from the KDI course and will be expanded on demand.

- Doctor: defined as a Person entity sourced from schema.org (https://schema.org/Person)
- Hospital: as sourced from https://schema.org/Hospital. This entity inherits several properties from the entity Place.

Furthermore, the final version of our schema is going to include rdf label tags for each attribute and entity in order to represent a concept in a different languages. This is fundamental in order to satisfy the requirements from our competency queries.

2.5.1.2 Variance respect CQs definition

This section aims to define the variance between the schema elements produced in this phase, and the definition of the CQs reported in the previous phase. This a way to define the quality of the outcomes for the current phase as well as the alignment of the overall project development process.

In order to keep our work focused and manageable, we opted for restricting the entities under consideration to those related to the categories of Laboratory Reports and Patient Summaries that should be comprised in a EHR system. We did not initially consider this option, since our goal was to define entities also regarding ePrescriptions or eDispensations, Medical images and reports and Hospital discharge reports.

For this reason, a great improvement on the previously defined CQs was needed, since we had to delete those not regarding strictly to the Patient Summaries and Laboratory Reports. At the same time, the total number of queries has more than tripled with respect to the previous phase, since a clearer understanding of the potential of the data collected has been gained.

2.5.2 Data level

The data level section in this phase reports the evolution of the datasets collected previously, reporting the metadata information for each new data, or new version of data, obtained.

2.5.2.1 Datasets management process

During the Informal Modeling phase the datasets collected in the previous phase are filtered and managed in order to obtain more suitable sets of data. In this section are described the procedures adopted to obtain that result.

After checking the variables contained on the datasets, was analysed the possibility of removing the ones that would not benefit the scope of our project and would not integrate with the initial set of entities defined. For each dataset will be presented the table correspondent to the variables that will be kept for our project.

Table 2: Filtered EMRBOTS

Dataset	Fields to be removed
EMRBOTS	Admission Id, Admission start date, Admission end date

Table 3: Filtered SMART

Dataset	Fields to be kept
SMART	Patient, Condition, Immunization, List, Observation

Table 4: Filtered SYNTHEA

Dataset	Keep only
SYNTHEA	“Resource type”: Patient, Allergy Intolerance, Immunization, Patient, Condition, Procedure, Diagnostic Report, Observation

2.5.2.2 Datasets metadata documentation

In this section is reported a list of new metadata in order to describe the modification performed on each datasets and attribute, to achieve the new version of the datasets. The metadata from the EMRBOTS, Synthea and SMART on FHIR datasets are described in the following tables:

Field Name	Description
Title	Health records, Diagnosis, Testing, Test results
Description	All the health records were created to analyze realistic health related data. The data was generated to project primary diagnosis, required medical tests, lab observation of the patients in both macro and individual scale. The data focuses on the aforementioned attributes of the patient health record.
Category	Electronic Health Record
Keywords	Electronic Medical Records, Patient Simulation, FHIR, Electronic Health Record
Source of dataset	EMRBOTS
Link of the website	http://www.emrbots.org/terms.html
Total number of observations	100000
Spatial coverage	United States of America
Privacy Level	No privacy restriction
Authority level	View, use, copy or modify for non-commercial use
Last date of modification	06/11/20

Table: Metadata description of EMRBOTS dataset

The metadata description for the Synthea dataset is given below:

Field Name	Description
Title	Health records, Prevalent conditions, Diagnosis, Procedure, Testing, Test results, Observation
Description	All the health records were created to analyze realistic health related data. The data was generated to project primary diagnosis, required medical tests, procedures, lab observation, immunization of the patients in both macro and individual scale.
Category	Electronic Health Record
Keywords	Electronic Medical Records, Patient Simulation, FHIR, Electronic Health Record
Source of dataset	Synthea
Link of the website	https://github.com/synthetichealth/synthea
Total number of observations	2000
Spatial coverage	United Kingdom, Finland
Privacy Level	No privacy restriction
Authority level	View, use, copy or modify under the terms and conditions of Apache license 2.0
Last date of modification	06/11/20

Table: Metadata description of Synthea dataset

The metadata description for the SMART on FHIR dataset is given below:

Field Name	Description
Title	Health records, Prevalent conditions, Diagnosis, Procedure, Observation
Description	All the health records were created to analyze realistic health related data. The data was generated to project primary diagnosis, procedures, lab observation, immunization of the patients in both macro and individual scale.
Category	Electronic Health Record
Keywords	Electronic Medical Records, Patient Simulation, FHIR, Electronic Health Record
Source of dataset	SMART
Link of the website	https://smarthealthit.org/
Total number of observations	67
Spatial coverage	United States of America
Privacy Level	No privacy restriction
Authority level	View or modify the data using the SMART apps only
Last date of modification	06/11/20

Table: Metadata description of SMART on FHIR dataset

2.5.2.3 Variance respect Inception datasets

This section aims to define the variance between the data elements (datasets and attributes within them) produced in this phase, and the initial datasets collected in the previous phase. This a way to define the quality of the outcomes for the current phase as well as the alignment of the overall project development process.

The data collection ended in the inception phase, therefore no new dataset has been added. Our already rich set of variables and medical concepts connected to the data suggested that adding more diversity would not have been required. At the same time, given each one of our datasets consists of a separate set of patients, collecting more information about one of these sets of patients would have been impossible given the constraints behind the collection of EHR data. For all these reasons we opted to work on the data already collected during the inception phase.

2.5.3 Informal Modeling Evaluation

While there were no metrics derived for evaluation of this phase, what could be said for now is that the EER drafted adheres greatly to the concepts (entities) defined in the FHIR standard, given this is already a far-reaching framework specification that effectively represents medical data.

2.6 Formal Modelling

In this section, we describe the creation of the ontology in Protege, preparation of the data for integration and lastly the creation of the metadata in the DCAT standard.

2.6.1 Schema level

The schema level section in the current phase, reports the detailed description of the ontology generation.

2.6.1.1 Ontology definition

This section reports in details how the ontology was generated, starting from the informal schema of the previous phase. In addition, which tools are used to do that, as well as usage of external ontology resources adopted to obtain the final KG ontology. Moreover a list of metadata is reported in this section, in order to describe all the elements of the ontology defined.

The informal EER was used as a starting point to define the entities constituting our final ontology, even though some changes (further detailed in the appropriate section of the report) had to be made with respect to the informally defined EER.

In order to define entities that would be in accordance with the FHIR standard (our reference ontology) we took advantage of the fhir.ttl RDF ontology officially released on this documentation page. We used this resource just to perform a check of our understanding of how the FHIR standard works, since the actual definitions of etypes and attributes included in this ontology are not compatible with the very simplified representation we have devised during the informal modelling. We considered the possibility of pruning the FHIR ontology, but the fact that the official FHIR ontology defines all concepts as entities (while we define only few key entities and then add data properties (attributes) on those).

Therefore, when it came to defining the actual ontology, we needed to work from scratch adding entities on top of a framework ontology provided by the UKC (iTelos repository). Through the provided KOS tool and related APIs we were able to first download this framework ontology, and then, after editing it, uploading it back to the UKC knowledge base.
KOS a knowdive group tool that allows to explore the knowledge contained in the UKC. In the context of our project, using this tool was fundamental to link the etypes defined in our ontology for our very specific purpose to the context provided by the network of interrelated concepts present in UKC. In this sense, KOS is at the core of the knowledge-data integration.
As a very first step, we checked in KOS whether the concepts we defined in our ontology were already present or if we needed to upload new concepts with relative definitions. By using the general search bar, we searched for etypes like ”Patient” and verified that the concept was already present in UKC with the sense corresponding to what we defined in our own ontology. No addtitional concepts needed to be deined and imported to the UKC since this already contained all the definitions we needed.

The editing phase of this process has been performed using Protégé. This tool allowed us to first define a set of entities and then to endow each of these of the desired attributes (Data properties). Below here follows a list of entities and relative attributes.

- Patient (GID "55936")
  - address: string
  - city:string
  - state: string
  - country: string
  - birthdate: dateTime
  - communication: string
  - gender: string
  - id: string
  - firstName: string
  - lastName: string
  - source: string
    - URL: string
    - dateTimeAccessed: dateTime
- Condition (GID "77328")
  - ClinicalStatus: string
  - encounter: dateTime
  - onsetDateTime: dateTime
  - code: string
  - subject: string
- Encounter (GID "110379")
  - id: string
  - period: dateTime
  - reasonCode: string
  - participant: string
  - type: string
- Observation (GID "31728")
  - category: string
  - effectiveDateTime: dateTime
  - id: string
  - subject: string
  - valueQuantity: float
    - unit: string
- Procedure (GID "5269")
  - code: string
  - id: string
  - patient: string
  - reasonReference: string
  - start: dateTime
  - end: dateTime
- Immunization (GID "4274")
  - vaccineCode: string
  - id: string
  - patient: string
  - occurrenceDateTime: dateTime
- AllergyIntolerance (GID "77258")
  - code: string
  - id: string
  - patient: string
  - clinicalStatus: string
  - recordedDate: dateTime
- MedicationOrder (GID "3440")
  - code: string
  - id: string
  - patient: string
  - dosageInstructions: string
  - dosageQuantity: float
    - unit: string

The following are basic information about the metadata regarding the SKG generated:

- Author: FHIR team
- Version: 1.0
- Date Created: 26/11/2020 17:00 CET
- Class count: 32
- Data property count: 74

The final version of our ontology can be seen in Figure 3. All classes we defined have the class entity as parent class, as shown in the picture. Furthermore, the class patient has outbound links to other the other defined classes. The object properties defined to formalize these links are the following (Domain is always patient, range is specified for each):

- has_diagnosis (to condition)
- has_attend (to encounter)
- has_analysis (to observation)
- medical_care (to procedure)
- has_immunity (to immunisation)
- has_risk (to allergyintollerance)
- has_ingesta (to medicationorder)

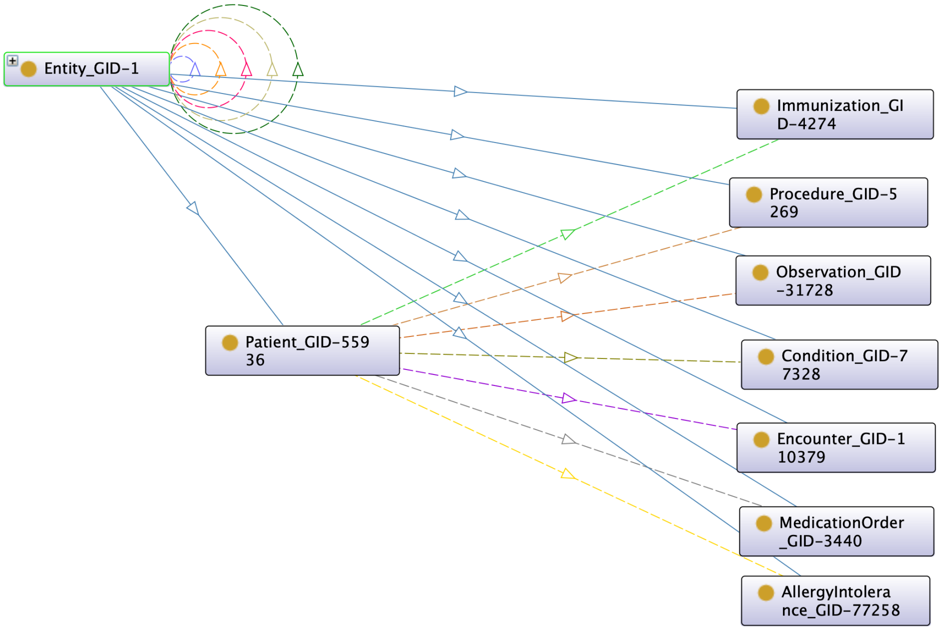


Figure 3: Final version of our ontology

2.6.1.2 Variance respect to the EER Model

The EER was revised and refined according to the new ontology, this included Implementing the indications we were given during the previous phase assessment, we needed to modify the total number of entities defined. The Patient entity has been unified, therefore no separate ”EMRBOTS”, ”SMART”, ”SYNTHEA” patient entities were created. Instead, the defined patient entity type encompasses attributes that encode the source and some metadata information about the provenance of the medical record the patient belongs to.
Furthermore, Doctor, Hospital and Laboratory entities were not created since we followed the advice to focus closely on our project’s scope.

2.6.1.3 Lexical information

As many of the entities have terms which can be ambiguous and our context requires interoperability between different countries with different languages, it is essential that the entities are clearly defined to a concept. We used concept definitions derived from the UKC (iTelos repository) to clearly identify the entities and it allows us to align our project with UKC concepts.

The following entities used the following definitions, where the format goes, concept, comment, and global ID;

- **Patient**: a person who requires medical care, 55936
- **Encounter**: come together, 110379
- **Observation**: facts learnt through observing, 31728
- **Condition**: the state of (good) health (especially in the phrases ’in condition’ or ’in shape’ or ’out of condition’ or ’out of shape’), 77328
- **Procedure**: a procedure employed by medical or dental practitioners, 5269
- **Medication**: the act of treating with medicines or remedies, 3440
- **Allergy**: hypersensitivity to a particular allergen, 77258

2.6.2 Data level

2.6.2.1 Formal Modelling datasets management

In order to have a better way to align the datasets with the ontology’s schema, was decided to unify the files for each data source. This unification wasn’t straightforward since the files have unique features, each dataset case will be described bellow.
EMRBots

- Tool used: KNIME
- Input type: CSV
- Output type: CSV
- ’LabsCorePopulatedTable.txt’ - is a huge file. Not having enough memory to load it on Python (even using lazy load). It was decided to just filter this file using KNIME and keep it separated (even using KNIME the process of join was taking too long)

- the other files were filtered and joined into one output

Synthea

- Tool used: Python
- Input type: JSON
- Output type: JSON
- The challenge in this case was to join the a large amount (1k) of JSON files into one while filtering. The result was one large JSON containing all the information.

SMART

- Tool used: Python
- Input type: XML
- Output type: JSON
- The XML file from this dataset was not formatted. Commonly used tools/libraries to convert it to JSON didn't work. It was necessary to search through all tags and elements, remove inconsistencies, filters the ones desired and add them to an empty JSON.

2.6.2.2 Datasets metadata documentation

In this section eventually new metadata information are added in order to describe the evolution of the datasets.

- The metadata generated during the SKG, LKG were explained by creating separate files. The link to the skg and dkg files is: (https://github.com/UNITN-KDI-2020/kdi-fhir/tree/master/dataset/Formal
- The JSON file for the metadata documentation was created using the DCAT format. (https://github.com/UNITN-KDI-2020/kdi-fhir/tree/master/dataset/Formal

2.6.2.3 Variance respect Informal Modeling datasets

During this phase, a refinement of the representation for the data regarding the address has been performed. Given that in our dataset the country, city, state and address line are in different data fields from one another, it hs been necessary to add one data property per each of these in the class patient. Note that in the informal modelling these were all grouped in the "address" attribute of the patient class, making it much harder to then map the data to our SKG.

2.6.3 Formal Modeling Evaluation

Evaluation of our schema relies on the calculation of several metrics. It is important to note that since our reference schema is extremely large and our addition very small these metrics are quite skewed. Furthermore, most of our ontology was based on the existing FHIR schema.
The set of metrics calculated are the following:

- Coverage: Reflects how much overlap exists between the two schema
- Flexibility: Represents how much does not overlap between our schema and the reference one

- Extensiveness: Is the proportional knowledge provided when considering the overall schema
- Sparsity: Details how much is not in the overlap between the reference and our schema

Furthermore, there are slight differences in how we have specified certain entities in our schema. For example, in the FHIR reference schema, Address is an entity in its own right. In our schema, Address is just a data property of Patient. This was done to simplify the ontology and fit it to our specific purposes, but represents the same concept. Because of this, we treat these as being equal even if strictly speaking they are not the same (one is a class, the other an attribute).
The result is that this leaves three attributes that exist in our schema but not in the reference schema. These are, in fact, our contextual entities:

- dateTimeAccessed
- URL
- source

The reference schema (FHIR) has 852 Classes and 3 Data Properties. Our schema has 8 Classes and 49 Data Properties. With an almost complete overlap (where most of our Data Properties are classes in FHIR) we can calculate the following metrics:

Metric	**Value**
Coverage	0.06
Flexibility	0.004
Extensiveness	0.003
Sparsity	0.94

2.7 Data Integration

In this last section we detail the operations required to import the data into KOS, build the DKG in Karmalinker and finally display and run SPARQL queries in GraphDB.

2.7.1 Data integration operations and tools

UKC-KOS importer
A large portion of this phase was spent importing the ontology created in the Formal Modelling phase through the provided APIs endpoints to interface with the KOS importer. The task proved challenging, due to several reasons. First of all, despite our great effort to check the correctedness of the classes and various properties names, we have initially encountered errors while trying to upload the Protege-defined ontology to KOS due to typos in their definitions. The problem required us to ask for support to the platform developers several times due to the absence of a debugging tool on the user side.
Another problem encountered in this phase was the definition of the ontology URIs, which in our case were not compliant with the standard required by KOS.
Solving these issues wasn't enough to obtain an output that could be then used in Karmalinker. In fact, intrinsic bugs in the platform were preventing a correct identification of the defined classes and properties. Once again, we had no other option than asking for support to our supervisors, who could solve the issue thanks to their knowledge of the inner workings of this tool.

Karmalinker

The process of data integration starts by loading the dataset and the schema into karmalinker. Once this process is successful, the next step is setting the semantic types for each column in order to align the dataset with the schema. Two parts that need to be kept in mind in this step, the changes that need to be done on the dataset (in terms of adaptability for karmalinker) and what class and properties should be set for each column. These two parts will be explained below.

At this point the dataset should be filtered and clean in order to be uploaded to karmalinker. However, the datasets Synthea and Smart were not recognized correctly by karmalinker. Given this situation it was decided to use a differente version of the Synthea dataset (bulk version) and compile the information of the Smart dataset for each patient. This bulk version of the Synthea dataset means that instead of having all information (allergy, observation, etc..) in one single file called patient, we would have them spread into files that represent the sections. That means that we would have, one file "Allergy" where would contain the information regarding to allergy of all patients. This reasoning would be applied to the others types of information.

For each dataset, it will be shown the operations done for each dataset. Also, is worthy note that for all datasets, was used just a subset of them in order to do the mapping.

EMRBots

This dataset is divided into two parts because one of the files is too big and was left separated instead of being merge. This setting is as following:

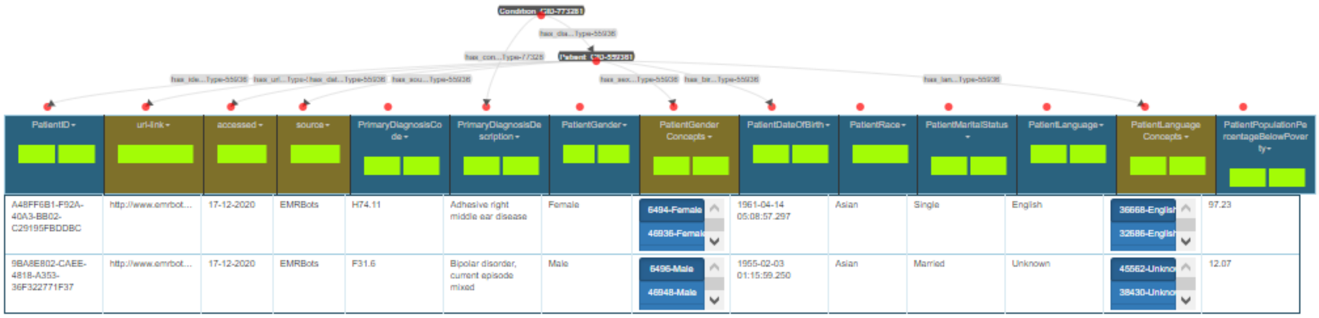
Part 1 - AdmissionsDiagnosesCorePopulatedTable and PatientCorePopulatedTable

This dataset contains all the information related to the patients and the diagnosis. The summary of the columns and classes used, with some description of the column are presented in Table 5.

Column Name	Description	Class
patientID	Unique ID of the patient	has identifier
url-link (added column)	Link to the EMRbots dataset	has url
accessed	Date last access	has date
source	Name of the dataset	has source
PrimaryDiagnosisDescription	Description of the diagnosis of the patient	has condition
PatientGender	Gender of the patient	has sex
PatientLanguage Concepts	Language spoken by the patient	has language

Table 5: EMRBots - Part1 (other files)

For Part 1, the result of the mapping operation can be seen in Figure 4.



PatientID	url-link	accessed	source	PrimaryDiagnosisCode	PrimaryDiagnosisDescription	PatientGender	PatientGender Concepts	PatientDateOfBirth	PatientRace	PatientMaritalStatus	PatientLanguage	PatientLanguage Concepts	PatientPopulationPercentageBelowPoverty
A48FF6B1-F92A-40A3-BB02-C29195FBDDBC	http://www.emrbot...	17-12-2020	EMRBots	H74.11	Adhesive right middle ear disease	Female	6494-Female 46936-Female	1961-04-14 05:08:57.297	Asian	Single	English	36668-English 32686-English	97.23
9BA8E802-CAEE-4818-A353-36F322771F37	http://www.emrbot...	17-12-2020	EMRBots	F31.6	Bipolar disorder, current episode mixed	Male	6496-Male 46948-Male	1955-02-03 01:15:59.250	Asian	Married	Unknown	45562-Unkno 38430-Unkno	12.07

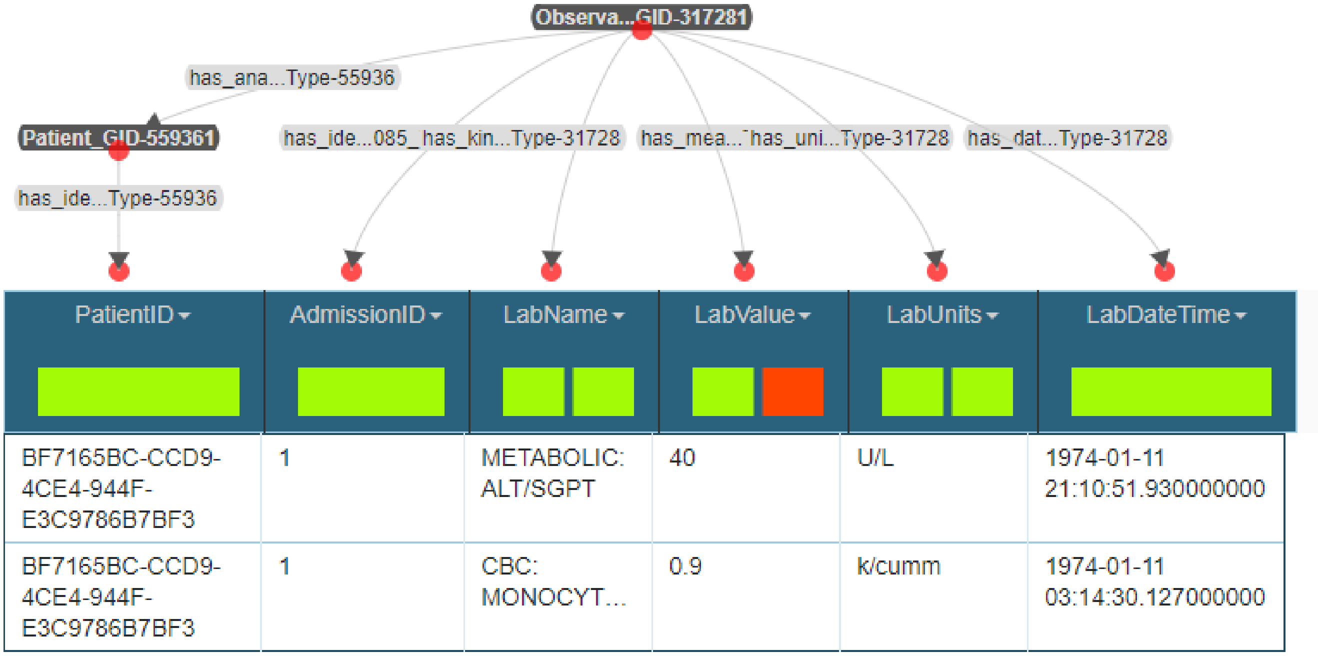
Figure 4: EMRBots - Part1 (other files)

Part 2 - LabsCorePopulatedTable This dataset contains all the information related to the lab tests and results. The summary of the columns and classes used, with some description of the column are presented in Table 6.

Column Name	**Description**	**Class**
patientID	Unique ID of the patient	has identifier
AdmissionID	ID of the Observation	has identifier
LabName	Description of the lab test	has kind
LabValue	Result of the lab test	has measure
LabUnits	Unit of measure of the lab test	has unit of measurement
LabDateTime	Date of the lab test	has date

Table 6: EMRBots - Part2 (LabsCorePopulatedTable)

For Part 2, the result of the mapping operation can be seen in Figure 5.



PatientID	AdmissionID	LabName	LabValue	LabUnits	LabDateTime
BF7165BC-CCD9-4CE4-944F-E3C9786B7BF3	1	METABOLIC: ALT/SGPT	40	U/L	1974-01-11 21:10:51.930000000
BF7165BC-CCD9-4CE4-944F-E3C9786B7BF3	1	CBC: MONOCYT...	0.9	k/cumm	1974-01-11 03:14:30.127000000

Figure 5: EMRBots - Part2 (Labcore)

Synthea

In case of Synthea, the whole dataset was divided into: AllergyIntolerance, Condition, Encounter, Immunization, MedicationRequest, Observation, Patient and Procedure. Some of the mapping images won't be shown because of the size of the image, being difficult to be presented.

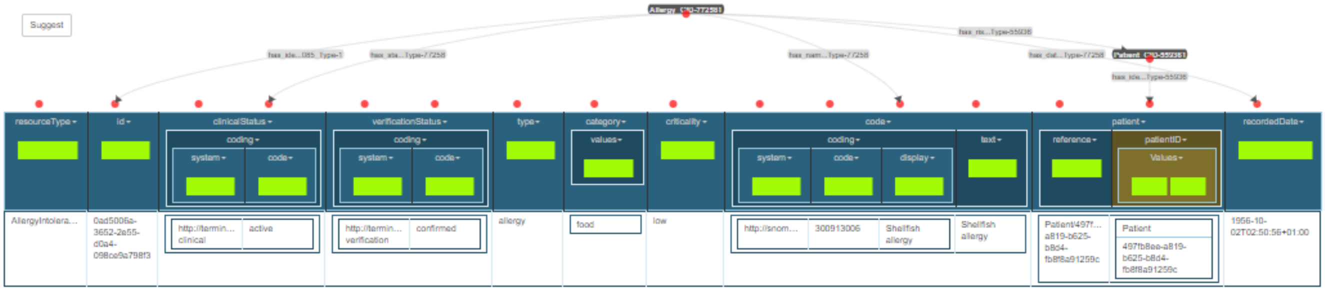
AllergyIntolerance

This dataset contains all the information related to allergies of our patients. The summary of the columns and classes used, with some description of the column are presented in Table 7.

In case of AllergyIntolerance, the link between Allergy and Patient would be 'has risk'. The result of the mapping operation can be seen in Figure 6

Column Name	Description	Class
id	ID of the entry for AllergyIntolerance	has identifier
code (from clinicalStatus)	Status of the allergy (active or not)	has state
display (from code)	Description of the allergy	has name
patientID	Unique ID of the patient	has identifier
recordedDate	Date of the entry	has date

Table 7: Synthea - AllergyIntolerance



resourceType	id	clinicalStatus	verificationStatus	type	category	criticality	code	patient	recordedDate
AllergyIntolera...	0ad5006a-3652-2a55-d0a4-098ce9a798f3	http://termin... clinical active	http://termin... verification confirmed	allergy	food	low	http://snom... 300913006 Shellfish allergy Shellfish allergy	Patient/497f... a819-b625-b8d4-fb8f8a91259c Patient 497fb8ee-a819-b625-b8d4-fb8f8a91259c	1956-10-02T02:50:56+01:00

Figure 6: Synthea - Allergy

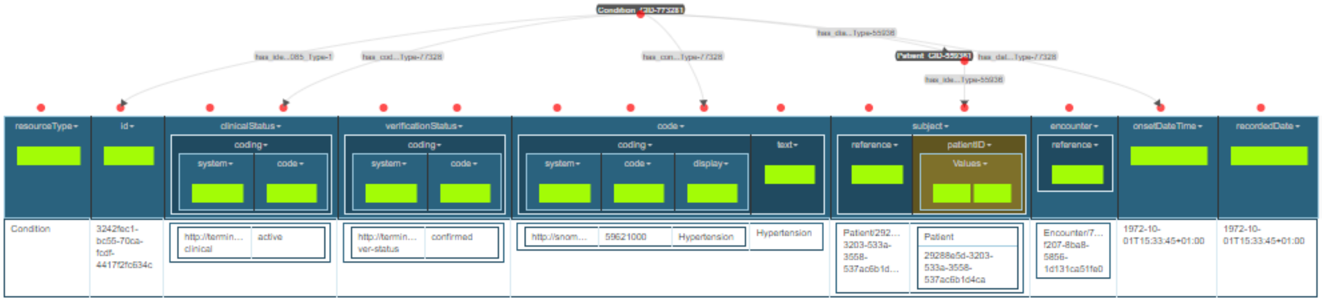
Condition

This dataset contains all the information regarding the historic of diseases of our patients. The summary of the columns and classes used, with some description of the column are presented in Table 8.

Column Name	Description	Class
id	ID of the entry for Condition	has identifier
code (from clinicalStatus)	Status of the disease (active or not)	has state
display (from code)	Description of the disease	has name
patientID	Unique ID of the patient	has identifier
recordedDate	Date of the entry	has date

Table 8: Synthea - Condition

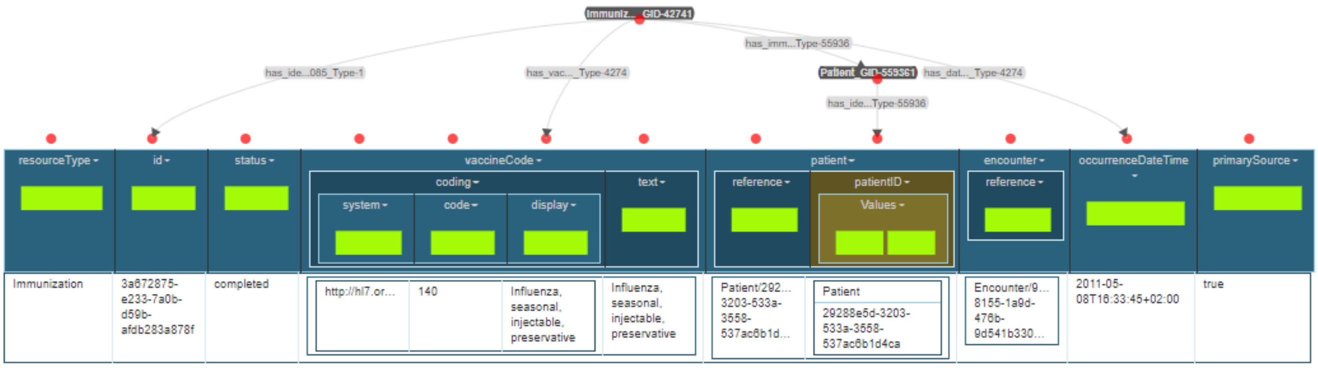
In case of Condition, the link between Condition and Patient would be 'has diagnosis'. The result of the mapping operation can be seen in Figure 7



resourceType	id	clinicalStatus	verificationStatus	code	subject	encounter	onsetDateTime	recordedDate
Condition	3242fec1-bc55-70ca-fcdf-4417f2fc534c	http://termin... clinical active	http://termin... ver-status confirmed	http://snom... 59621000 Hypertension Hypertension	Patient/292... 3203-533a-3558-537ac6b1d... Patient 29288e5d-3203-533a-3558-537ac6b1d4ca	Encounter/7... f207-8ba8-5856-1d131ca51fe0	1972-10-01T15:33:45+01:00	1972-10-01T15:33:45+01:00

Figure 7: Synthea - Condition

For Condition, the result of the mapping operation can be seen in Figure 8



resourceType	id	status	vaccineCode				patient		encounter	occurrenceDateTime	primarySource
			coding			text	reference	patientID	reference		
			system	code	display			Values			
Immunization	3a672875-e233-7a0b-d59b-afdb283a878f	completed	http://hl7.or...	140	Influenza, seasonal, injectable, preservative	Influenza, seasonal, injectable, preservative	Patient/292... 3203-533a-3558-537ac6b1d...	Patient 29288e5d-3203-533a-3558-537ac6b1d4ca	Encounter/9... 8155-1a9d-476b-9d541b330...	2011-05-08T16:33:45+02:00	true

Figure 8: Synthea - Immunization

Immunization

This dataset contains all the information regarding the historic of immunizations of our patients. The summary of the columns and classes used, with some description of the column are presented in Table 9.

Column Name	Description	Class
id	ID of the entry for Immunization	has identifier
display (from code)	Description of the immunization	has name
patientID	Unique ID of the patient	has identifier
recordedDate	Date of the entry	has date

Table 9: Synthea - Immunization

Smart

For smart the figure was too big to be added to the document. Differently from Synthea, this dataset has all information in one file (Patient). A summary of the columns and classes used can be seen at Table 10.

Column Name	Description	Class
city	city where the patient is living	has city
country	country where the patient is living	has country
line	street name where the patient is living	has address
state	state where the patient is living	has state
birthDate	Date of birth of the patient	has state
gender	Gender of the patient	has sex
family	Family name of the patient	has surname
given	Name of the patient	has name
patientID	Unique ID of the patient	has identifier
class(Encounter)	Description of the type of the encounter	has type
end (period - Encounter)	Date of the end of the encounter	has date (GID-110379)
EffectiveDateTime(Observation)	Date of the observation	has date (GID-31728)

Table 10: Smart

GraphDB

After exporting the Turtle file from Karmalinker, it could be imported into GraphDB. While the importing worked fine, there were issues with visualization of the DKG due to the lack of IRIs, instead Python was used to convert the Turtle file into a DKG. The following are a subset of our examples from our finished DKG, the first

(Figure 9) from the EMRBots dataset and the second (Figure 10) from the Smart on FHIR dataset. Note these were done outside GraphDB.

Lastly, while we could not perform SPARQL queries due to the very problem that halted the visualization of our DKG (that is, a lack of suitable IRIs), we can assume that if the SKG was properly specified, that the following would work. We provide here an example of query that could be performed and explain the reasoning behind it.

```
select ?genid ?name
where
        {
        _:etype_Observation_GID-317281 <http://knowdive.disi.unitn.it/etype
        #has_analysis_GID-31507_Type-55936> ?genid .
        ?genid <http://knowdive.disi.unitn.it/etype#has_name_GID-2_Ty
        pe-55936> ?name.
        }
    filter (?name = ”Giannantonio”)
```

Is expected to match those triples with subject an instances of class observation, predicate its relationship with the patient class, and object the genid of the patient instance linked. Then, the second part of the query would return the ?name of all instances of patient that match the ?genid matched in the first part of the query. Finally, the filter statement returns only the values for the specified name (”Giannantonio”).

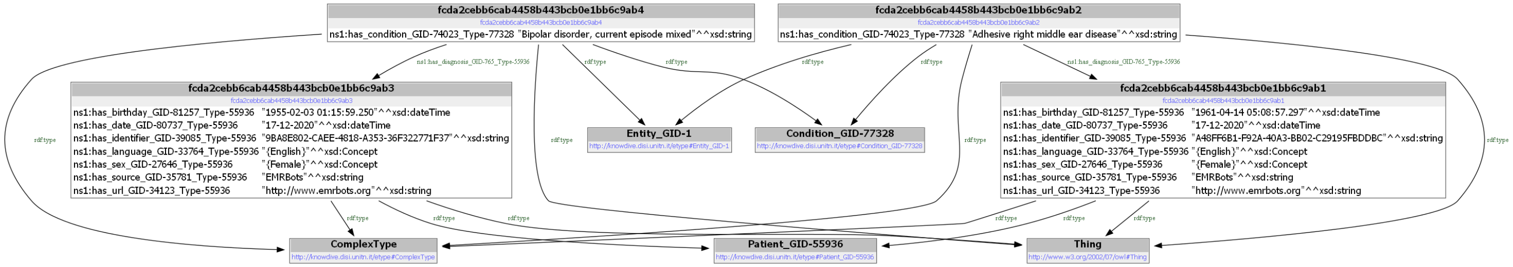

fcda2cebb6cab4458b443bcb0e1bb6c9ab4
ns1:has_condition_GID-74023_Type-77328 "Bipolar disorder, current episode mixed"^^xsd:string
fcda2cebb6cab4458b443bcb0e1bb6c9ab2
ns1:has_condition_GID-74023_Type-77328 "Adhesive right middle ear disease"^^xsd:string
ns1:has_diagnosis_GID-765_Type-55936
rdf:type
fcda2cebb6cab4458b443bcb0e1bb6c9ab3
ns1:has_birthday_GID-81257_Type-55936 "1955-02-03 01:15:59.250"^^xsd:dateTime
ns1:has_date_GID-80737_Type-55936 "17-12-2020"^^xsd:dateTime
ns1:has_identifier_GID-39085_Type-55936 "9BA8E802-CAEE-4818-A353-36F322771F37"^^xsd:string
ns1:has_language_GID-33764_Type-55936 "{English}"^^xsd:Concept
ns1:has_sex_GID-27646_Type-55936 "{Female}"^^xsd:Concept
ns1:has_source_GID-35781_Type-55936 "EMRBots"^^xsd:string
ns1:has_url_GID-34123_Type-55936 "http://www.emrbots.org"^^xsd:string
Entity_GID-1
http://knowdive.disi.unitn.it/etype#Entity_GID-1
Condition_GID-77328
http://knowdive.disi.unitn.it/etype#Condition_GID-77328
fcda2cebb6cab4458b443bcb0e1bb6c9ab1
ns1:has_birthday_GID-81257_Type-55936 "1961-04-14 05:08:57.297"^^xsd:dateTime
ns1:has_date_GID-80737_Type-55936 "17-12-2020"^^xsd:dateTime
ns1:has_identifier_GID-39085_Type-55936 "A48FF6B1-F92A-40A3-BB02-C29195FBDDBC"^^xsd:string
ns1:has_language_GID-33764_Type-55936 "{English}"^^xsd:Concept
ns1:has_sex_GID-27646_Type-55936 "{Female}"^^xsd:Concept
ns1:has_source_GID-35781_Type-55936 "EMRBots"^^xsd:string
ns1:has_url_GID-34123_Type-55936 "http://www.emrbots.org"^^xsd:string
ComplexType
http://knowdive.disi.unitn.it/etype#ComplexType
Patient_GID-55936
http://knowdive.disi.unitn.it/etype#Patient_GID-55936
Thing
http://www.w3.org/2002/07/owl#Thing

Figure 9: Example DKG using EMRBots information

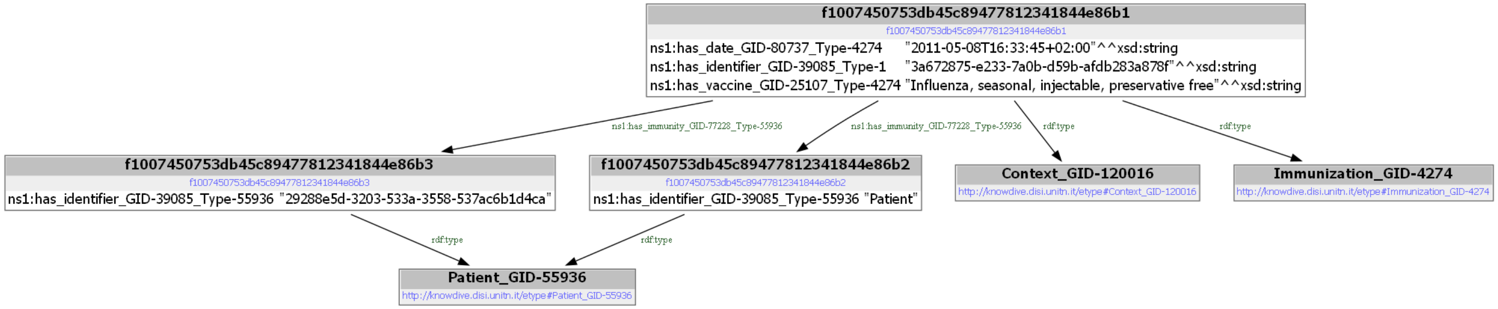

f1007450753db45c89477812341844e86b1
f1007450753db45c89477812341844e86b1
ns1:has_date_GID-80737_Type-4274 "2011-05-08T16:33:45+02:00"^^xsd:string
ns1:has_identifier_GID-39085_Type-1 "3a672875-e233-7a0b-d59b-afdb283a878f"^^xsd:string
ns1:has_vaccine_GID-25107_Type-4274 "Influenza, seasonal, injectable, preservative free"^^xsd:string
ns1:has_immunity_GID-77228_Type-55936
ns1:has_immunity_GID-77228_Type-55936
rdf:type
rdf:type
f1007450753db45c89477812341844e86b3
f1007450753db45c89477812341844e86b3
ns1:has_identifier_GID-39085_Type-55936 "29288e5d-3203-533a-3558-537ac6b1d4ca"
f1007450753db45c89477812341844e86b2
f1007450753db45c89477812341844e86b2
ns1:has_identifier_GID-39085_Type-55936 "Patient"
Context_GID-120016
http://knowdive.disi.unitn.it/etype#Context_GID-120016
Immunization_GID-4274
http://knowdive.disi.unitn.it/etype#Immunization_GID-4274
rdf:type
rdf:type
Patient_GID-55936
http://knowdive.disi.unitn.it/etype#Patient_GID-55936

Figure 10: Example DKG using Smart Immunization information

2.7.2 Variance respect Formal Modeling datasets

In this final section we detail the last changes that were made to our knowledge graphs, including modifications to our ontology and things needed to be changed with the data when compared to the Formal Modelling section.

Our ontology when compared to the formal modelling underwent modifications. Mostly because that it wasn't correctly specified, or had small errors in the specification. Similarly, Karmalinker required us to "chunk" the data since it could not read in the JSON files that we had prepared.

While we would have liked to run SPARQL queries on our linked data, there was just not enough time to do that.

In terms of the datasets, we couldn't use the final format from the previous steps due to some restrictions in how the data was organized and read through karmalinker. Also, the way that our synthetic data was organized, through entries in a json, made it difficult for the filtering part. Since all parts of the data that related to the patient (i.e. immunization, procedures..) was scattered.

Another issue was the size of our datasets and the limit of Karmalinker. This made the use of the full data unviable. Because of this issue, it was used only part of the dataset for the mapping.
The generation of the final graphs had issues as well, we had to reduce even more our data in order to get a readable image.