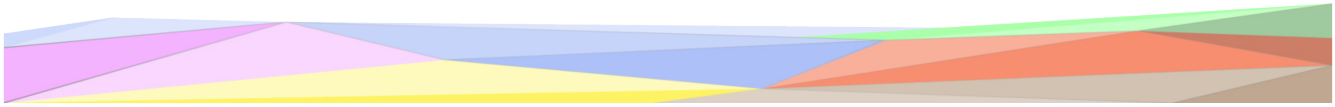
UNIVERSITY
OF TRENTO - Italy

Dipartimento di Ingegneria e Scienza dell'Informazione

– KnowDive Group –

Healthcare-EHR-SynData

Document Data:	Reference Persons:
January 17, 2023	Artusi Alessia, Salti Emma



Index:

Revision History:

Revision	Date	Author	Description of Changes
0.1	20.04.2020	Fausto Giunchiglia	Document created

1 Introduction

Reusability is one of the main principles in the Knowledge Graph Engineering (KGE) process defined by iTelos. The KGE project documentation plays an important role in order to enhance the reusability of the resources handled and produced during the process. A clear description of the resources and the process developed, provides a clear understanding of the KGE project, thus serving such an information to external readers in order to exploit that in new projects.

The current document aims to provide a detailed report of the KGE project developed following the iTelos methodology. The report is structured, to describe:

- **Section 2**: the project's purpose, the domain of interest (DoI) and the resources involved (both schema and data resources) in the integration process.
- **Section 3**: the input resources considered by the KGE project.
- **Section 4, 5**: the integration process along the different iTelos phases, respectively.
- **Section 8**: how the result of the KGE process (the KG) can be exploited.
- **Section 9**: conclusions and open issues summary.

There is an increasing interest across European Union (EU) institutions, government and healthcare systems to digitalise the healthcare ecosystem. As it was proven by the COVID-19 pandemic, a digital healthcare ecosystem would facilitate across-borders processes, creating homogeneity across Europe. There is not a single common European Electronic Health Record (EHR) system for all EU member states.

The objective of this project is to develop an easy-to-use knowledge graph that, integrated with EHR data, will result in a tool that facilitates healthcare data visibility and transfer.

2 Purpose and Domain of Interest (DoI)

The first and most important step of the KG development process defined by iTelos methodology is the purpose formalization and the definition of the domain of the project. The purpose and the DoI will strongly define the outcome of the data integration process, as it is purpose-driven.

2.1 Purpose

"A service which facilitates citizens to handle (their own) healthcare data, in a context in which the citizens moves around Europe."

The purpose of our project is to develop a service which facilitates citizens to handle their own healthcare data, in a context in which they move around Europe. We aim to integrate heterogeneous healthcare data to create a knowledge base that can be exploited in clinics, pharmacies, hospitals and research institutions. This type of information is already available in most European countries in the form of Electronic Health Record (EHR), a digital version of a patient's paper chart, but it is not integrated as a unique system across Europe because of the countries different healthcare systems. EHR implementation has many advantages, the most notable are:

- Improved productivity of physicians
- Improved patient care delivery
- Facilitated cross-border care
- Easier data processing

2.2 DoI

We defined as the project domain of interest the different European countries considered as a unique space entity. Several projects already exist and they are working on the facilitation of the cross-border interoperability of EHRs in the EU; in particular, the e-Health Digital Service Infrastructure (eHDSI) is a work carried out by the Commission through the Connecting Europe Facility (CEF) Programme [1]. Some improvements were made during the COVID-19 pandemic to make the Green Certificates available and valid across Europe and more.

3 Data Sources

3.1 Knowledge sources

A central challenge for healthcare standards is how to handle the wide variability caused by diverse healthcare processes. For this project we referred to two knowledge resources publicly available to model our entity types:

- **Schema.org**

  provides schemas for structured data on the Internet, on web pages, and beyond. Its vocabulary cover entities, relationships between entities and actions; in particular, we exploited the textit'Health and medical types: notes on the health and medical types under MedicalEntity's schema'.

- **FHIR**

  Fast Healthcare Interoperability Resources is a next generation standards framework created by HL7. The solutions built by FHIR are suitable or use in a wide variety of contexts, including the clinical and administrative world.

3.2 Data sources

Due to privacy concerns and technical burden, it is difficult to get access to Electronic Medical Records (EMRs); hence, we decided to exploit some databases from which we downloaded or generated synthetic healthcare data. With this method we bypassed the problem of anonymization and we had access to a large number of data, which is always challenging in the medical field. The databases we used are:

- **Synthea**

  It is a synthetic patient population simulation used to model the medical history of synthetic patients. By default it generates records for patients living in the United States, considering our DoI we decided to generate data by altering the demographics, zip codes, time zones and provider files exploiting files from the GitHub repository: `https://github.com/synthetichealth/synthea-international` on European demographics. We generated synthetic data for 500 patients by modifying some regional parameters in order to align it to the European context. The output `.csv` files are listed below:

  - allergies
  - careplans
  - claims
  - conditions
  - devices
  - encounters
  - imaging_studies
  - immunizations
  - medications
  - observations
  - organizations
  - patient_expenses
  - patients

  - procedures
  - providers
  - supplies

- **EMRBOTS**

  EMRBots are experimental artificially generated electronic medical records (EMRs). The aim of the database is to allow non-commercial entities to use the artificial patient repositories to practice statistical and machine-learning algorithms. We downloaded the 100-patient database that contains data on: 100 patients, 372 admissions, 111483 lab observations records. The properties described are not as various and detailed as the ones we generated with Synthea, but they report a good number of features nonetheless.

3.3 Metadata

Below we report the metadata relative to the databases described above.

- **Synthea**

  This synthetic data generation tool was developed by the MITRE corporation in 2016. It is an open-source Synthetic Patient Population Simulator. The goal is to output synthetic, realistic patient data and associated health records in a variety of formats. It is one of the most popular datasets in this domain and it is used for various purposes such as machine learning, deterministic predictive model tuning, and data integration projects.

- **EMRBOTS**

  The database was created in April 2015 by Kartoun Uri.[2] The database contains the same characteristics that exist in a real medical database such as patients' admission details, demographics, socioeconomic details, medications, while relying only minimally on real patient data.

4 Purpose Formalization

The aim of the project is to make health data available at anytime and anywhere the subjects might find themselves. To help us define in every shade our purpose, in this chapter we describe several personas that integrated with scenarios will form our competency queries (CQs). These CQs will drive our project together with the purpose, with our knowledge graph we aim to satisfy every scenario presented in this chapter.

- **Personas**

  P1: *Pedro*, cardiologist at La Fe university and polytechnic hospital in Valencia, Spain.

  P2: *Sonia*, an Italian woman from Olmeneta, Italy.

  P3: *Andreas*, biotechnologist at the university hospital in Berlin.

  P4: *Giulia*, researcher at the University of Trento.

P5: *Pierre*, PI in charge of the laboratory working on COVID-19. The laboratory is in contact and financed by the French government.

- **Scenarios**

  S1: Pedro is a cardiologist at La Fe Hospital in Valencia. During its work time a foreign patient arrives to the hospital unconscious. To intervene the doctor needs to know the patient medical records in order to safely operate.

  S2: Sonia is an Italian woman from Olmeneta, Italy. She is 67 years-old and decides to travel around Andalusia (Spain) for a couple of weeks. One day, during her trip she notices a cutaneous rush probabily due to some allergies she is not aware of. She travels to the nearest emergency room but does not speak spanish or english.

  S3: Andreas is a biotechnologist who works at the university hospital of Berlin. His usual day in the lab consists in performing analysis on the patients' lab samples. He need a records of the already performed tests in order to avoid running duplicate tests.

  S4: Giulia is a PhD researcher at the University of Trento, her study focus is rheumatoid arthritis. In particular, she would like to go through some existing medical records to find whether there are association with previous conditions of the patients' in the developing of the disease.

  S5: Pierre is a lab PI at Institute Pasteur in Paris. During the COVID-19 pandemic he was asked by the government to perform studies on the factor associated with higher risks of having a severe pathology after the COVID-19 vaccine (SARS-COV-2). He needs data relative to the patients' medical history in order to assess some causal relationship among different health status.

5 Inception

The first phase of the iTelos methodology is the Inception phase. In this paragraph we will first introduce the competency queries that will drive our project structure definition. The CQs are defined by the integration of the personas and scenarios as previously described in the Purpose Formalization chapter.

We will continue with the specification of how we collected the resources and their classification into core, common and contextual with respect to each CQ.

Personas	Scenarios	Competency Queries		
		Question	Action	Description
Pedro	Pedro is a cardiologist at La Fe Hospital in Valencia. During its work time a foreign patient arrives to the hospital unconscious. To intervene the doctor needs to know the patient medical records in order to safely operate.	1.1	Access the patient's medical history	A list of current and previus diagnoses is return
		1.2	Access to know the patient's allergies	A list of the patient's allergies is returned
		1.3	Access to the latest blood tests	A list of most recent blood tests with the correspondent results are shown
Sonia	Sonia is an Italian woman from Olmeneta, Italy. She is 67 years-old and decides to travel around Andalusia (Spain) for a couple of weeks. One day, during her trip she notices a cutaneous rush probabily due to some allergies she is not aware of. She travels to the nearest emergency room but does not speak spanish or english.	2.1	Access to the registered allergies	A list of the patient's allergies is returned
		2.2	Access personal information to make a formal identification	Name, lastname, gender and birthdate of the patient are returned
		2.3	Access to the condition history	A list of current and previus diagnoses is return
Andreas	Andreas is a biotechnologist who works at the university hospital of Berlin. His usual day in the lab consists in performing analysis on the patients' lab samples. He need a records of the already performed tests in order to avoid running duplicate tests.	3.1	Access to a complete list of the lab tests performed in his hospital	A list of lab tests grouped by the patient's id reported together with the date and time
Giulia	Giulia is a PhD researcher at the University of Trento, her study focus is rheumatoid arthritis. In particular, she would like to go through some existing medical records to find whether there are association with previous conditions of the patients' in the developing of the disease.	4.1	Access to general information on the patients'	The patients' personal information is returned
		4.2	Access to specific patients' diagnoses	The system returns a list of patients there where diagnosed with rheumatoid arthritis
		4.3	Access to complete diagnoses list	A list of diagnoses belonging to the patients that are affected by rheumatoid arthritis

Pierre	Pierre is a lab PI at Institute Pasteur in Paris. During the COVID-19 pandemic he was asked by the government to perform studies on the factor associated with higher risks of having a severe pathology after the COVID-19 vaccine (SARS-COV-2). He needs data relative to the patients' medical history in order to assess some causal relationship among different health status.	4.4	Access to lab tests	Lab tests results belonging to the patients that are affected by rheumatoid
		5.1	Specific immunizations	The system returns a list of patients that received the SARS-COV-2 injection
		5.2	Access to latest diagnoses	A list of diagnoses belonging to the patients that were vaccinated against COVID-19 infection

Figure 1: Competency Queries

5.1 Resources collection

Considering our purpose we focused on the collection of Electronic Health Records (EHRs), these data go beyond standard clinical data collection providing a broader view on a patient's care. They are intended to be handled by more than one healthcare organization.

We collected synthetic (therefore anonymous) health data from the sources described above. In particular:

- **Synthea**: we generated data for 500 patients located in seven different regions of England, namely Cumbria, Essex, Fife, Kent, Norflok, Wiltshire, and Somerset. The output consists in multiple `.csv` files listed earlier in the Data Sources section. This dataset contains the following attributes related to the patients:
  - Patient ID: unique identifier
  - Name: name and surname of the patient
  - Gender
  - Birthdate
  - Marital Status
  - Address: complete address
  - Language: spoken language
  - Encounters: encounters conducted on the patient with the relative diagnosis description and code
  - Careplans: description of the careplan for each encounter
  - Devices: devices used by the patient (i.e.: wheelchair)
  - Allergy

  - Treatment: the treatment prescribed to the patient
  - Dosage: dosage of the treatment
  - Immunizations

- **EMRBOTS**: we downloaded an already available database containing the following files:
  - `PatientCorePopulatedTable.txt`
  - `LabsCorePopulatedTable.txt`
  - `AdmissionCorePopulatedTable.txt`
  - `AdmissionDiagnosesCorePopulatedTable.txt`

  Where we find records on 100 patients, 372 admissions and 111483 laboratory observations. Overall the datasets contain the following information about the patients:
  - Patient ID
  - Gender
  - Birthdate
  - Population percentage below poverty
  - Marital status
  - Admission ID: unique identifier for the patient's admission
  - Admission start and end date and time
  - Primary diagnosis code: ICD-10 code for the diagnosis
  - Primary diagnosis description
  - Lab test name
  - Lab test result
  - Lab test date

  The diagnoses in this dataset are identified by the ICD-10 (International Classification of Diseases, 10th Revision) code, which is the World Health Organization's (WHO) medical coding system for labeling diseases.

6 Informal Modeling

This section describes the Informal Modeling phase. The Informal Modeling phase of iTelos consists in integrating the two layers of a knowledge graph: the data layer and the knowledge layer.For what concerns the data layer we will modify and filter our datasets accordingly to the ER model to include only the data relevant to our knowledge graph.

6.1 Knowledge Layer

This paragraph reports the first informal definition of the *Entity Types* (ETypes) and of the *Enhanced Entity Relationship* (EER) model constructed using them. Our EER model is based on the classifications of the resources into common, core and contextual, considering the competency queries described in the Inception chapter.

6.1.1 Resources classification

We defined the classification of our data as:

Core: we consider as core the entities connected to the subjects of our KG: patients, medical staff and researchers. In particular we consider as core:

- Person: defines the concept of human being.
- Allergy: defines allergy or intolerance that affects a person
- Immunization
- Diagnosis
- Encounter

Common: the only common data among the different CQs is the *location*.

Contextual: starting from the CQs, depending on the scenario, the contextual data can be identified as:

- Doctor
- Patient
- Hospital

From this classification of resources in common, core and contextual we defined our *ETypes* and their *properties* as shown below.

In our ER model we started with two entities, namely Patient and Doctor, since they are our ideal users of the KG; we noticed they both shared most of the data properties, such as: name, address, date of birth, gender, race and language, hence we decided to create the superclass **Person** with the data properties listed, that will be inherited by its subclasses **Patient** and **Doctor**.

We continued by modeling the data properties of the Person, since it is the entity that is linked with most data resources. We decided to model its connections as ETypes that have their own data properties. Hence the EType Person is directly associated with the entity types:

- **Allergies**
- **Encounter**
- **Immunization**
- **LabTest**

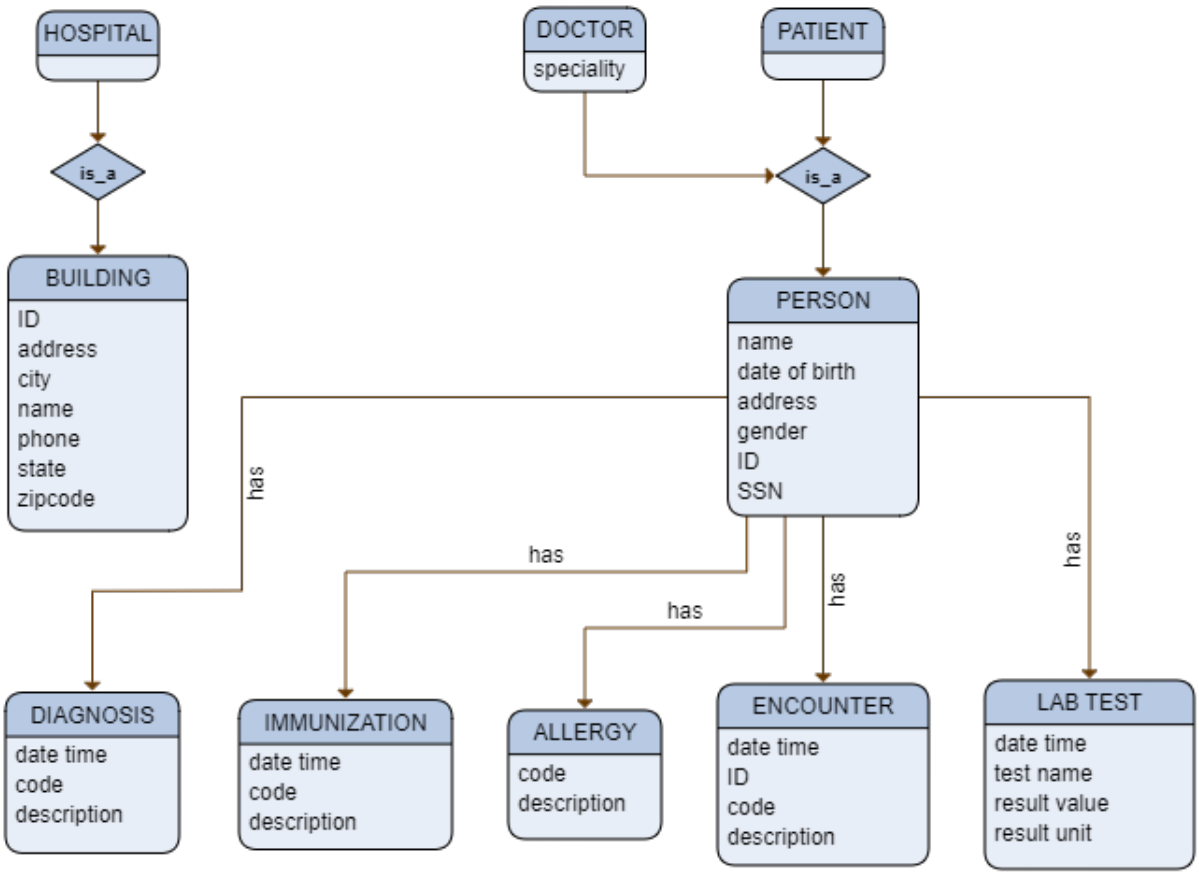


Figure 2: starting EER model

- **Diagnosis**

Moreover we introduced the EType **Organization** and created its subclass **Hospital**.

In this phase we only introduced the relations of type *has* and *is_a*, because of this Organization - Hospital and the remaining part of the EER model are not connected. They will be when more detailed object properties will be introduced.

6.2 Data Layer

During this phase the dataset previously collected are filtered and handled to obtain more suitable sets of data for our EER model. The data handling was minimal as all the data we downloaded/created comes as `.csv` files with little cleaning needed. We spent some time over the language and denomination issue, as the same property is referred to with different names in different datasets.

The Synthea information were filtered (kept) accordingly to what is reported below: `patients1.csv`: 500 instances

- patient id
- birth date

- first name
- last name
- gender
- SSN

`allergies.csv`: 436 instances

- patient id
- allergy description
- allergy code
- encounter id

`encounters.csv`: 19533 instances

- patient id
- organization id
- encounter id
- description
- date-time

`providers.csv`: 372 instances

- doctor id
- organization id
- specialization
- gender
- first name
- last name

`building.csv`: 394 instances

- building id
- name
- phone
- zip
- address
- city

- state

While, for what concerns The EMRbots datasets, every field was kept except for the Admission codes that are only sequential numbers not useful for data mapping. The files obtained are: `patients2.csv`: 100 instances

- patient id
- birth date time
- gender

`labtest.csv`: 111483 instances

- patient id
- date time
- lab test name
- lab test result value
- lab test result unit

`diagnoses.csv`: 373 instances

- patient id
- diagnosis description
- diagnosis code

The data types were modified when needed (i.e.: dates); the script (`dataset_processing.ipynb`) used to modify the data sets is available at the GitHub page of the project.

7 Formal Modeling

As formal modeling we intend the creation of an ontology using Protégé, the language alignment and an initial setup for the data integration step.

7.1 Knowledge Level

Starting from the informal schema described in the previous section, we built the ontology in Protégé. To define entities in accordance with the FHIR standard we used the resource to search our entities and the related data properties that were to be added. Through the provided KOS tool we downloaded the validated ontology with the GID codes incorporated (*Note: some adjustments to the ontology were made after 31st of December, when the application was already closed, hence some classes, data properties or object properties are not defined by a GID code). KOS is a knowdive group tool that allows to explore the knowledge contained in the

UKC (Universal Knowledge Core). For each EType we looked for already present definition that matched our context and assigned them to our entities.

To detail our ontology with data properties and object properties, we used Protégé. The result is listed below:

1. **PERSON**

   Data attributes:

   - has_person_id [string]
   - has_person_name [string]
   - has_person_surname [string]
   - has_person_gender [string]
   - has_person_birthdate [DateTime]
   - has_person_role [string]

2. **Patient (GID-55936)**
3. **Doctor (GID-53662)**

   Data attributes:

   - has_doctor_speciality [string]

4. **Encounter (GID-45307)**

   Data attributes:

   - has_encounter_id [string]
   - has_encounter_date [DateTime]
   - has_encounter_description [string]
   - has_encounter_code [string]

5. **Diagnosis (GID-765)**

   Data attributes:

   - has_diagnosis_code [string]
   - has_diagnosis_description [string]

6. **Allergy (GID-77258)**

   Data attributes:

   - has_allergy_code [string]
   - has_allergy_description [string]

7. **Immunization (GID-100786)**

   Data attributes:

   - has_immunization_code [string]

- has_immunization_date [DateTime]
- has_immunization_description [string]

8. **Lab Test (GID-19865)**
   Data attributes:
   - has_labtest_date [DateTime]
   - has_labtest_name [string]
   - has_labtest_result [decimal]
   - has_labtest_unit [string]
9. **BUILDING**
   Data attributes:
   - has_building_name [string]
   - has_building_address [string]
   - has_building_id [string]
   - has_building_phone [int]
   - has_building_state [string]
   - has_building_zipcode [string]
   - has_building_city [string]
10. **Hospital (GID-43695)**

To have a dataset that resembles best some real-world data, we created the following object properties that are linked the entities as shown in the logic model in figure reffig:logic_model.
Object properties:

- BelongTo
- ConductedBy
- ConductedOn
- DiagnosedDuring
- Discovered
- Employs
- WorkAt
- has
- Ordered
- OrderedDuring

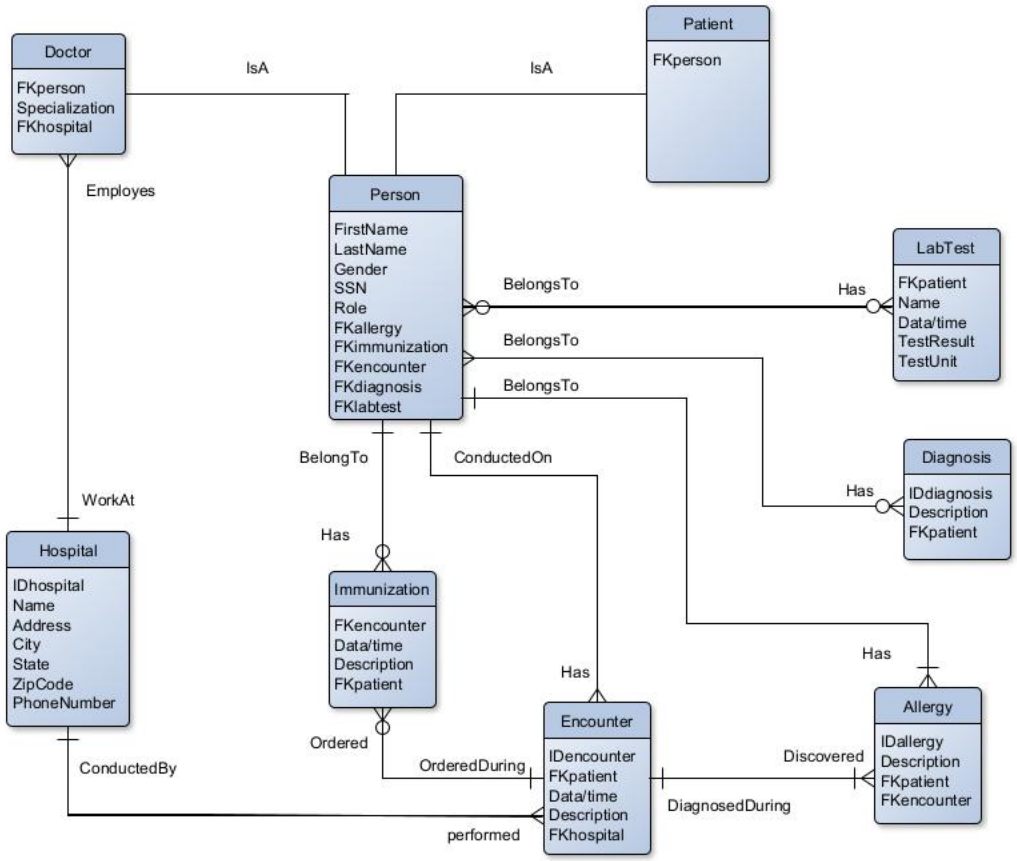


Figure 3: Complete logic model

Using as reference the EER diagram earlier described, we built the *logic model*. A logic model is the formalization of the EER model, we integrate three main concepts: entities, relationships and attributes. In our logic model we integrated a data type for each attribute and more specific relationship between entities, namely one-to-one, one-to-many and many-to-many relationships.

7.2 Data Layer

Data from the two different data sources was joined in a unique data base (`.csv` files loaded onto GitHub page). In order to have more clarity and uniformity on the columns names, some were renamed across the whole dataset.

*(*Note: the following step was elaborated at the end because the program GraphDB does not recognise the relationship "is_a").*

We merged every file containing information about people ([`patients1.csv, patients2.csv, doctors.csv`]) in a single file (`people.csv`). A column *role* was added, which specifies "PA-

TIENT"/"DOCTOR" according to the file of origin, so that when querying the dataset we have a way of filtering patients or doctors only. Two more columns were added: *organization* and *speciality*, so that whenever an occurrence is referred to a doctor, we also keep record of the hospital for which he/she works and of his/her speciality. This cleaning step resulted in seven data sets (namely: people, buildings, immunizations, allergies, labtests, diagnoses, encounters) each representing an entity type; while the ETypes Hospital, Patient, and Doctor are contained inside the tables implicitly. When needed the format of the data types were modified with Python accordingly to what they represent (i.e.: dates where converted into date-time objects using the library pandas).

8 KGC

In this section we detail the operations required to import the data into KOS, build the KGC in Karma, this tool allowed us to connect our data with the reference ontology.

8.1 Karma

The data integration process started by loading the ontology in `owl` format and the data sets into Karma. Once everything has been successfully loaded, we associated to each column its semantic type to align the data with the schema. Examples of these alignments are shown in the figures 4, 5 below:

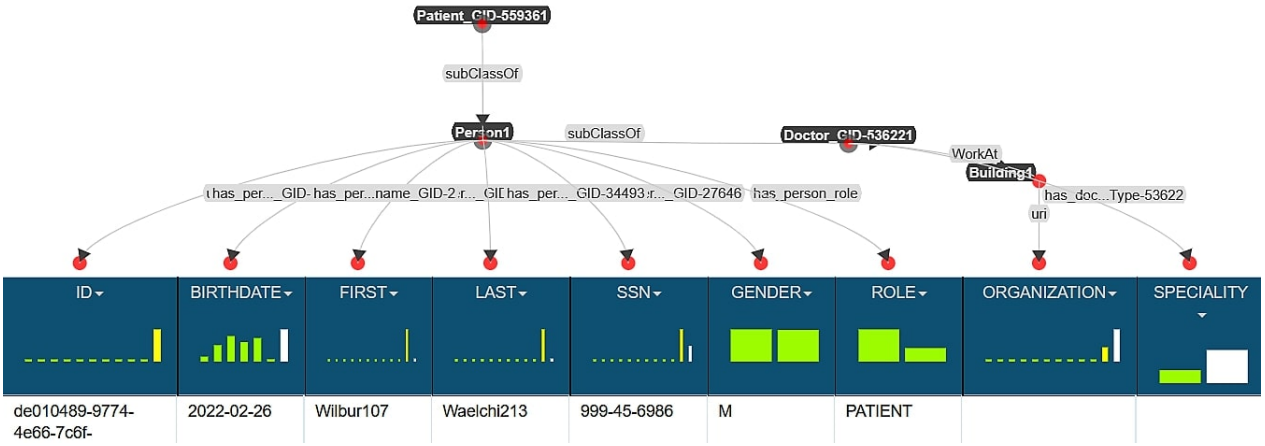


Figure 4: Data integration of the table `people.csv` using Karmalinker. In the column *organization* and *speciality* no values appear as the occurrence refers to a patient which does not have values for those two columns.

As specified earlier in the report, the sub classes Patient and Doctor where not recognised by GraphDB, hence the data properties that were originally connected to them are now linked to the EType Person. The same goes for Hospital, which links are now on the EType Building.

The same process was applied to the five remaining data sets. No issue was encountered during this phase, except for the modification needed after noticing the error in GraphDB.

After creating the models in karma for each data set we exported the files in `.ttl` format and loaded them into GraphDB.

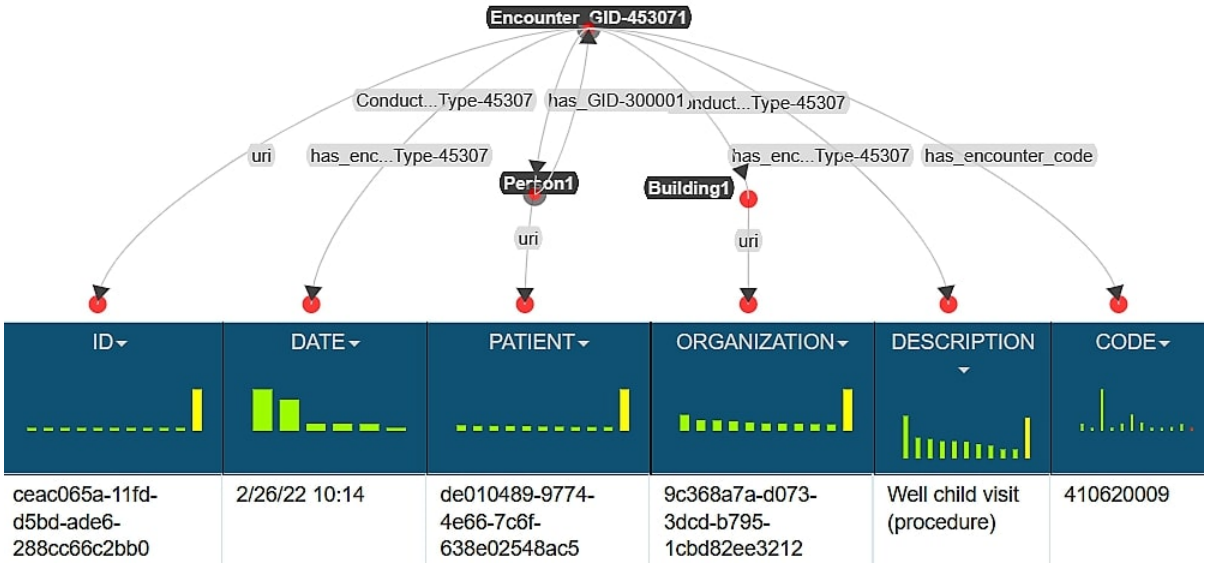


Figure 5: Data integration of the table `encounter.csv` using Karmalinker.

9 Outcome Exploitation

This section aims to provide a description of the KGE process outcome. Here you have to report the final Knowledge Graph information statistics (like, number of ETypes and properties, number of entities for each EType, and so on). Moreover this section has to provide a description for the KG possible exploitation, like examples of queries executed, execution time, and so on.

9.1 GraphDB

The seven turtle files, one for each data set, generated with Karma were imported into GraphDB. Every file was successfully loaded onto GraphDB, however it did not recognise the sub classes as such, that is when we modified our logic model and EER model such that the data sets could be interrogated through GraphDB.

Below and example of a query that we implemented using SPARQL (integrated in GraphDB). This query (figure 6a) is guided by the competency query 1.3 (Pedro).This example query outputs the data shown in figure 6b.

Unfortunately, due to an issue in the data type of value in column 'LabTestDate', the dates can not be filtered or ordered, as they are recognised as strings an not as a DateTime value. Ideally one would have added the piece of code: `FILTER(?LabTestDate > "6/30/92 3:50")`

Where "6/30/92 3:50" is an arbitrary date value, that in a real world example would coincide with the date of TODAY.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?PersonName ?PersonSurname ?LabTestName ?LabTestValue ?LabtestUnit ?LabTestDate WHERE {
    ?Person a <http://knowdive.disi.unitn.it/etype#Person> ;
            <http://knowdive.disi.unitn.it/etype#has_person_name_GID-2> ?PersonName ;
            <http://knowdive.disi.unitn.it/etype#has_person_surname_GID-34003> ?PersonSurname.
    ?LabTest a <http://www.semanticweb.org/simone/ontologies/2021/10/untitled-ontology-34#LabTest_GID-19865> ;
             <http://www.semanticweb.org/simone/ontologies/2021/10/untitled-ontology-34#has_labtest_name_GID-2_Type-19865> ?LabTestName
;
             <http://www.semanticweb.org/simone/ontologies/2021/10/untitled-ontology-34#has_labtest_result_GID-36475_Type-19865> ?
LabTestValue ;
             <http://www.semanticweb.org/simone/ontologies/2021/10/untitled-ontology-34#has_labtest_unit_GID-102744_Type-19865> ?
LabtestUnit ;
             <http://www.semanticweb.org/simone/ontologies/2021/10/untitled-ontology-34#has_labtest_date_GID-80737_Type-19865> ?
LabTestDate .

    FILTER REGEX (?LabTestName, "^CBC*") .
    FILTER(?PersonName = "Bess947") .
}
```

(a) SPARQL query example

PersonName	PersonSurname	LabTestName	LabTestValue	LabtestUnit	LabTestDate
Bess947	Huels583	CBC: HEMATOCRIT	440	%	9/30/97 9:38
Bess947	Huels583	CBC: HEMATOCRIT	324	%	9/30/97 22:57
Bess947	Huels583	CBC: WHITE BLOOD CELL COUNT	91	k/cumm	9/30/97 16:56
Bess947	Huels583	CBC: MEAN CORPUSCULAR VOLUME	778	fl	9/30/97 16:52
Bess947	Huels583	CBC: ABSOLUTE NEUTROPHILS	728	%	9/30/97 16:41

(b) GraphDB SPARQL query example output. For an arbitrary patient ('Bess947'), its surname and values relative to blood lab tests are shown, namely lab test name, lab test results with the unit and the date of the test.

10 Evaluation of the Final KG

There are two types of evaluation: the first on how much the final KG is able to satisfy the competency queries (explicit goal), and the second on how much reusable is the final KG (implicit goal).

Starting with the evaluation of the satisfaction of the purpose, we consider two parts: Schema Layer Data Layer To evaluate each of these segments of the KG, we exploit the **coverage**, which is a measure of how much knowledge (ETypes and properties) does the KG cover.

10.1 Knowledge Layer

To evaluate the coverage of the knowledge layer, we compute how much the ETG covers the Entities and properties extracted from the CQs. Because of the issues we had, which made us go back and forth the steps throughout the process, we ended up defining the CQs and modeling our ETG with little discrepancy, hence every entity type is basically covered by the ETG. We see less coverage for what concerns the properties though. We compute the coverage as: $Cov = \frac{Number of covered Entities}{Entities in Ontology}$.

We applied the same formula for properties (object and data properties together).

Entity Coverage	$10/10$	1
Property Coverage	$(39+20-7)/(39+20)$	0.88

10.2 Data Layer

This evaluation aims to understand how much the KG is connected. The connectivity could be measured over two dimension: Entity connectivity: How much the entities are connected to each other. Property connectivity: How much the entities are connected to their properties.

To get an idea of the connectivity of our KG we checked the number of links in it. As a result we obtained the following numbers :

ENTITY TYPE	ENTITIES E(T)	OBJECT P. Op(T)	DATA P. Dp(T)
ALLERGY	7000	2	2
BUILDING	12000	1	6
DIAGNOSIS	744	1	2
DOCTOR	3000	2	7
ENCOUNTER	50000	3	3
HOSPITAL	4000	2	6
IMMUNIZATION	10000	2	3
LABTEST	20000	1	4
PATIENT	5000	1	6
PERSON	55000	5	6
TOTAL	**159744**	**20**	**45**

Figure 7: Connectivity of the final KG

11 Open Issues

This section concludes the project and aims to describe any issues/problems that occurred along the entire KGE process.

11.1 Data Collection

We managed to collect healthcare data of synthetic patients and of some fractions of the healthcare systems. However, we only generated data from specific UK regions, hence they are not representative of the entire Domain of Interest we considered (Europe). EMRbots data comes without specification on the location of the patients. The choice of modifying the provenance of the data with Synthea is not easy, therefore we kept the data as we originally generated it (given also that we did not need to confront medical data from different regions). However, to better represent a real-world data one could generate more representative datasets.

11.2 ETG

Because GraphDB did not (still does not) recognise the subclass relationship, we went backwards with the whole process and modified the ER model and everything that comes after, so that we could check with GraphDB if everything worked. Unfortunately, these changes modified our initial idea of the ETG model, as we wanted to keep as the center of everything the patient and not the person as we managed to do at the end. Nonetheless, the queries work and the identification of patients as person could even be a better solution, as sometimes doctors are going to be identified as patients.

12 Conclusions

In many ways the project was challenging. We are happy with the results as we managed to obtain a working Knowledge Graph. We started with a more complex idea, but because of synthetic data scarcity and lack of time we chose to build a simpler model that could somehow shape some real-world events. iTelos was useful in creating a directed workflow, that gives the KG shape as new steps are approached, however it is quite impossible to modify a single detail without deleting the work that comes after.

References

[1] European Commission. Exchange of electronic health records across the eu. https://digital-strategy.ec.europa.eu/en/policies/electronic-health-records.

[2] Uri Kartoun. Advancing informatics with electronic medical records bots (emrbots). *Software Impact*, 2(100006), 2019.