D5.10

Design of the Data Mapper and Converter to FHIR - V2

ABSTRACT

The role of data mapping and conversion is to transform health records produced or used locally at hospitals to a semantically interoperable representation as proposed by the InteropEHRate project. This document describes the novel data mapping and conversion method adopted by InteropEHRate, which includes the description of the general principles, roles, processes, as well as the programming interfaces and main software components involved. A comparison of the method to state-of-the-art approaches is also provided.

Delivery Date	July 9th, 2021
Work Package	WP5
Task	T5.3
Dissemination Level	Public
Type of Deliverable	Report
Lead partner	UniTN

InteropEHRate project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826106.

This document has been produced in the context of the InteropEHRate Project which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826106. All information provided in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose.

CONTRIBUTORS

	Name	Partner
Contributors	Gábor Bella	UniTN
Contributors	Simone Bocca	UniTN
Reviewers	Alain Pena	A7
Reviewers	Konstantinos Koutsoukos	BYTE

LOGTABLE

Version	Date	Change	Author	Partner
0.1	2020-08-10	Created as copy of V1 release, added section 2	Gábor Bella	UNITN
0.2	2020-08-20	Methodological clarifications	Gábor Bella	UNITN
0.3	2020-10-15	Description of new terminological standards supported	Simone Bocca	UNITN
0.4	2021-01-25	Added material on related work	Gábor Bella	UNITN
0.5	2021-06-10	Updated according to latest changes to hospital data flow	Simone Bocca	UNITN
0.6	2021-06-16	Ready for internal review	Simone Bocca, Gábor Bella	UNITN
0.9	2021-06-17	Reviewed by A7	Alain Pena	A7
0.91	2021-06-18	Reviewed by BYTE	Konstantinos Koutsoukos	BYTE
0.99	2021-07-01	Ready for final QC and submission	Gábor Bella	UNITN
1.0	2021-07-04	Quality check	Argyro Mavrogiorgou	UPRC
Vfinal	2021-07-09	Final technical revision and submission	Francesco Torelli Laura Pucci	ENG

ACRONYMS AND TERMS

Acronym	Term and definition
API	Application Programming Interface
ConversionI	Application Programming Interface of the InteropEHRate Conversion Services component, for the conversion of content and structure within health records into a Smart Health Record compliant form.
CKR	Central Knowledge Repository: a centralised (e.g. EU-wide) service that provides the standards that constitute the *InteropEHRate Interoperability Profile* as formal knowledge, that is to say in a form that can be downloaded and directly imported into the Health Data Integration Platform.
CSV/TSV	Comma-Separated Values / Tab-Separated Values: simple machine-readable tabular file formats.
IHS	InteropEHRate Health Services: a high-level software component (a collection of libraries) that provides high-level data translation and conversion services to end-user applications.
IHSI	Application Programming Interface of the InteropEHRate Health Services component that provides high-level data transformation services.
IHT	InteropEHRate Health Tools: a set of interactive helper tools that are used by hospital employees (data scientists) to set up and maintain the health data integration system. These tools are outside the scope of the project deliverables and thus are not fully specified in any deliverable document.
JSON	JavaScript Object Notation: standard, computer-readable, tree-structured file format.
LEI	Legacy EHR Interface: Application Programming Interface implemented by local health institutions (hospitals) to provide a unified access to EHRs to InteropEHRate services.
Smart Health Data	The interoperable, multilingual, standard, FHIR-based representation of health data, as defined and used by the InteropEHRate project.
TranslationI	Application Programming Interface of the InteropEHRate Translation Services component, for the translation of natural language text within EHRs.
XML	Extensible Markup Language: machine-readable tree-structured file format.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Scope of the document

This deliverable provides the specification of the services, software components, and processes responsible for converting local hospital EHRs into the interoperable Smart Health Data form that is proposed by the InteropEHRate project. The conversion services described in this deliverable rely on the lower-level functionalities InteropEHRate Data Integration Platform, specified in [D5.8], and are, in turn, used by the InteropEHRate Health Services (also specified in [D5.8]) that are responsible for adapting the conversion services to the local technical infrastructure of the healthcare institution.

1.2. Intended audience

This document adopts a "service" perspective in the sense that it concentrates on how data conversion services can be called and used by applications (such as the *HCP App*), and also on the mechanisms by which these services are set up and maintained by hospitals. As opposed to deliverable [D5.8] (*Health Data Integration Platform*) that provides a technical view on data integration, this deliverable presents data conversion services from the perspective of healthcare data and the related health domain data representation standards.

1.3. Structure of the document

Section 2 motivates mapping and conversion with respect to the InteropEHRate scenarios. Section 3 provides the high-level architecture of the InteropEHRate Health Services, and within them the Conversion Services and the Health Data Integration Platform. Section 4 gives an overview of the general principles of the knowledge-driven approach to health data integration and conversion. Section 5 presents in detail the process by which health knowledge and the data integration pipeline that uses it need to be set up, used, and maintained. Section 6 presents high-level information with respect to APIs and implementation of the software components involved in health data conversion. Section 7 provides details about the international (cross-border) knowledge that is being set up for the project following the guidelines of section 3. Section 8 presents related work and how our design goes beyond the state of the art. Section 9 gives conclusions and addresses future work.

1.4. Updates with respect to previous version

In the current version of this document, new sections have been added and already existing sections have been improved with more details. More specifically, the updated sections are the following:

- **Section 2:** this newly introduced section situates the need for data mapping and conversion within the usage scenarios covered by the InteropEHRate project, providing motivation for the contents of this deliverable.
- **Section 4:** contains a new clarifying sentence about the role of a central, EU-wide authority in the maintenance of formal international knowledge.
- **Section 5:** now makes more explicit the role of local and international data scientists, and provides a more detailed and more precise description of the processes.

- **Section 5.2:** now provides more details on the process of setting up the mapping and conversion mechanisms.
- **Section 6:** added description of non-standard datatypes, removed the specifications of the IHS Controller and Interface and moved it into [D5.8], where it belongs. Added an updated, detailed description of the automated conversion process, and the corresponding sequence diagram.
- **Section 7.6**: the definition of UCUM has been added, as a new standard supported by the project.
- **Section 7.7**: the definition of ICNP has been added, as a new standard supported by the project.
- **Section 8:** this newly introduced section provides comparisons between the solution exposed and related work in the context of knowledge and data mappings and transformations.

Small changes have also been made in other sections in order to add missing details with respect to the previous version of the document.

2. RELEVANCE WITH RESPECT TO THE INTEROPEHRATE SCENARIOS

Data mapping and conversion are essential functions in all four scenarios addressed in the InteropEHRate project and defined in [D2.3]. In the *Device-to-device* and *Research* scenarios, data mapping and conversion to the interoperable (FHIR-based) representation take place before the health record is loaded onto the patient's mobile device. Likewise, in the *Emergency* scenario, mapping and conversion are applied before smart health data is uploaded onto the S-EHR Cloud.

Optionally, mapping and conversion can also be re-applied later, in order to convert the FHIR-based health record into a representation better exploitable for local needs, such as into the representation locally used in the "target" hospital of a different country in order to allow the integration of the health record into the local EHR system (in the *Device-to-device* and *Emergency* scenarios) or into a filtered, transformed, and aggregated format better suited to the purposes of medical research (in the *Research scenario*).

Finally, the goal of the *Semantic data management scenario*, as defined in [D2.3], is to set up the formal knowledge and the data mapping and conversion rules that enable automated mapping and conversion in all of the three other scenarios. The knowledge-driven mapping and conversion methods are presented below in section 4.

3. CONVERSION SERVICES IN THE INTEROPEHRATE ARCHITECTURE

InteropEHRate Health Services (IHS) is a set of high-level software services provided to hospitals and health applications in order to carry out operations related to health record interoperability. Depicted in the centre of the diagram below, it contains the following components:

- a central *Data Integration Platform*, as specified in [D5.8];
- two high-level interoperability service implementations:
  - *Conversion Services:* specified in this deliverable, their role is to convert EHRs from the format locally used by a hospital to a cross-border interoperable format;
  - *Translation Services:* specified in deliverable [D5.11], their role is to translate and localise natural-language text contained in EHRs from the original source language to other European languages, such as the language of the patient or the language locally used by the hospital, as required by interoperability;
- an *InteropEHRate Health Services Controller* (IHS Controller) component, specified in [D5.8], that serves as middleware between the interoperability services, the Hospital Information System, and applications, adapting the data conversion and translation services to the local context from a technical point of view.

In order to facilitate setting up the formal knowledge for the Conversion and Translation services within the Data Integration Platform, the IHS is complemented by a set of *InteropEHRate Health Tools* (IHT):

- Knowledge Management Tools, comprising of knowledge modellers and viewers, that let data scientists define, update, and visualise the formal knowledge that drives the conversion processes (as described below in section 4) in interactive and batch modes;
- a Data Mapper Tool that lets data scientists define the conversion and mapping rules towards the Interoperability Profile [D2.8].

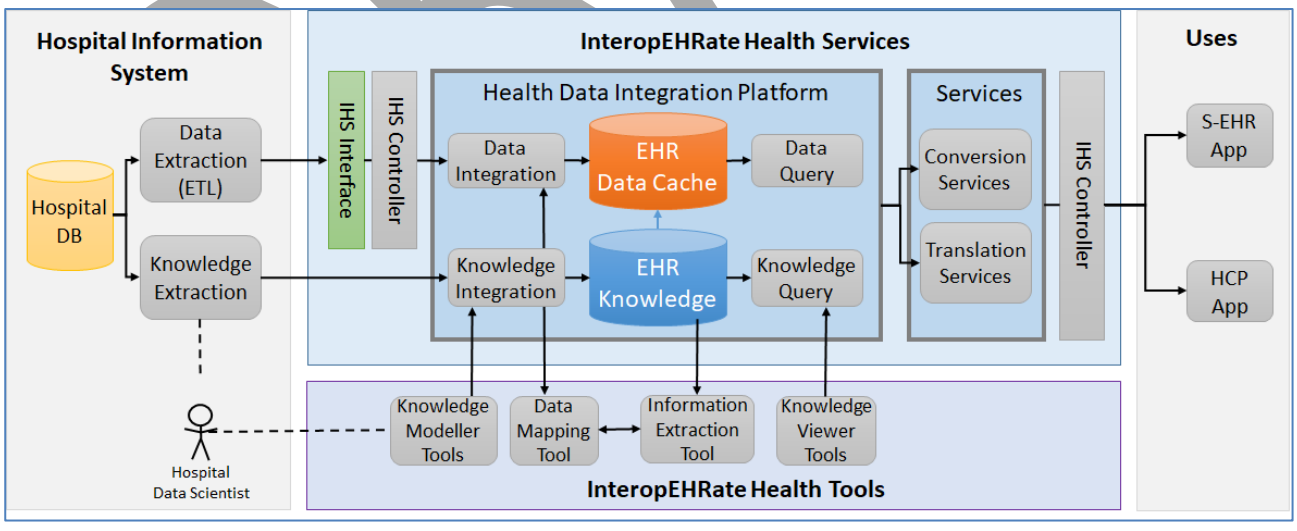

Figure 1: The architecture of InteropEHRate Health Services and its relation to the Hospital Information System, Applications, and InteropEHRate Health Tools

4. PRINCIPLES OF KNOWLEDGE-DRIVEN CONVERSION

The particularity of the InteropEHRate solution to data conversion is that it is entirely knowledge-driven: instead of being hardcoded into business logic, data conversion rules and methods are rather encoded dynamically in a flexible manner. This has the advantage of making the adaptation and the evolution of conversions a much more agile process: instead of going through costly software development cycles, the system can be adapted to changes in data structures, encoding standards, etc., virtually by pointing and clicking using interactive tools, the changes being immediately effective on the whole system.

The price of this flexibility is the need to formally encode all the knowledge necessary for the conversions. This includes locally relevant knowledge (locally used terminologies, data schemas, etc.), international knowledge as defined in the InteropEHRate *Interoperability Profile* [D2.8], as well as mappings between the two. As international knowledge is, in principle, identical for every participating country and institution, it will be pre-loaded into all deployable instances of the IHS. The InteropEHRate Project suggests that the formalisation and maintenance (extension, updating) of international knowledge should be the responsibility of a central, EU-wide authority. All local institutions need to formalise their locally relevant knowledge (only to the extent necessary for the conversions) and define their mappings to international knowledge. This is what is meant by *knowledge extraction* and *knowledge integration* in fig. 1 above (see the next section for details).

The table below shows the different kinds of knowledge that need to be integrated into the *EHR Knowledge Base* (see fig. 1) for conversions to be fully operational. The *EHR Knowledge Base* encodes knowledge in a layered architecture where each layer is constructed from elements from the ones below it:

- **the lexicon layer** encodes the meaningful and relevant words, domain terms, and codes that are used in local health data, in the languages supported by the IHS system;
- **the concept layer** encodes the supra-lingual meanings of words, terms, and codes; it is organised into a hierarchical graph and acts as a bridge across languages;
- **the schema layer** encodes data structures used to describe types of entities relevant to health data, such as patients, hospital visits, prescriptions, medicine, laboratory tests, etc.;
- **the entity layer** encodes data instances that, contrary to concepts, are richly described, such as substances or drug products.

The table below gives a summary of the kinds of knowledge contained in the EHR Knowledge Base.

	Local	International	Mappings
entities	optional: richly described locally used data instances (e.g., drug substances described with multiple attributes)	internationally defined data instances (e.g., drug substances richly described by ATC)	equivalence of local and international entities
schemas	optional: locally used data structures	international data structures, i.e., FHIR	structural mappings and data transformations (defined in the Data Mapping Setup step)

concepts	the meanings of locally used terms, encoded as nodes of a graph	international encoding systems, e.g., ICD or LOINC	equivalences or is-a relations between local and international terms
lexicon	natural language labels used for local concepts, including local codes	labels of international codes	equivalences between labels expressing the same meaning, both within and across languages

Table 1 - Kinds of knowledge

5. HEALTH DATA CONVERSION PROCESSES AND ROLES

In order for health data conversion to be an automatic process, the Platform needs to be set up and customised with respect to the data representations used in the hospital. This section describes the entire process by which conversion is set up, then used on a daily basis, and also regularly maintained to follow the natural evolution of data representations, both local and international. Two kinds of roles are involved in carrying out the process:

- **a data scientist** is the primary person in charge of formalising healthcare knowledge, its mappings, as well as defining data conversions. The data scientist has a good background in healthcare data and knowledge modelling. If (s)he is in charge of formalising the mappings of local data, (s)he also needs to be familiar with local data modelling practices. While (s)he should have some programming knowledge, (s)he does not need to be a professional software developer.
- **A software developer** has a supporting role behind the data scientist, with the goal of automating repetitive modelling tasks. For example, the integration of a coding system with tens of thousands of codes is best executed as a batch task that may require procedural automation (i.e., through scripting). Such one-shot tasks are best carried out by a software developer.

The people above are typically employed by a hospital and their role is to formalise the knowledge, knowledge mappings, and data conversions from local representations towards the representations defined in the InteropEHRate *Interoperability Profile* [D2.8]. These roles will be called *local data scientist* and *local software developer*. As international knowledge itself is invariant across hospitals, it is made available as a pre-loadable formal knowledge and is maintained by *international data scientist(s)* and *software developer(s)* typically employed by the central authority officially in charge of its maintenance and provision.

The mapping and conversion process is divided into six macro-steps:

1. **[ONLY ONCE] International knowledge integration:** coding systems, terminologies, data schemas, and concept mappings defined in the Interoperability Profile are semi-automatically encoded as formal knowledge by the *international data scientist* (supported by an *international software developer*) using the *Knowledge Management Tools* and are made available for hospitals in a format that is directly loadable into the Data Integration Platform of each hospital.
2. **[ONLY ONCE] Local knowledge extraction and integration:** local coding systems, terminologies, and concepts that are relevant to the support of smart health records are semi-automatically formalised by the local data scientist (supported by a local software developer) using the *Knowledge Management Tools*.
3. **[ONLY ONCE] Local data mapping setup:** data schema mappings and data value transformations are formally defined in a semi-automatic way, in the form of a *conversion recipe*, by the local data scientist.
4. **[DAILY, ON DEMAND] Automated data extraction and integration:** the conversion recipe is automatically applied to the health data that need to be transformed to the international format.
5. **[OCCASIONALLY, AS NEEDED] Local knowledge and data mapping maintenance for data evolution:** elements of the knowledge and/or the recipe are updated by the local data scientist so that they keep up with the evolution of data representations and encoding standards.

6. **[OCCASIONALLY, AS NEEDED] International knowledge maintenance for data evolution:** elements of the international knowledge are updated by the *international data scientist* so that they keep up with the evolution of international data representations and encoding standards.

5.1. Knowledge Extraction and Integration

Knowledge extraction and integration are once-and-for-all tasks that are executed in an initial bootstrapping phase of the deployment of the IHS at a local healthcare institution. As international knowledge is pre-loaded into the knowledge base, this macro-operation concerns only local knowledge and its mapping to international knowledge. The operation is divided into the following steps:

1. **Local knowledge extraction:** this is a fully manual analysis of local datasets in order to interpret their contents in terms of the four formal knowledge layers. The result of this activity is a document that defines the subset of elements (strings, labels, codes) in local data that will need to be represented as formal knowledge.
2. **Local knowledge and mapping integration:** in this step, the data scientist and/or the software developer takes the document above as input and, using the InteropEHRate Health Tools and scripting, produces formal knowledge ready to be imported into the knowledge base. Knowledge integrated at this point does not need to describe local data exhaustively: it is enough to define the mappings that are necessary to convert locally used terms, codes, and data structures into international knowledge. Such mappings are defined in one of the following ways:
    - **on the lexical and concept levels:**
        - if the concept (meaning) of a locally used term has an equivalent concept in the international knowledge (such as a local ICD-9 code “433.00” being equivalent to the international ICD-10 code “i65.1”), define the label of the local term to be a synonym of the international term,
        - if the concept (meaning) of a locally used term does not have an equivalent concept in the international knowledge, create the new concept as broader or narrower than the closest existing international concept (if any), and add the local term as its lexicalisation;
        - the steps above are executed:
            - through filling in an Excel sheet manually for a low to medium number of terms and/or concepts (up to a few hundred),
            - through batch importing of automatically generated Excel sheets, in case the number of terms and their concepts is high (thousands to millions);
    - **on the schema level:** the formal definition of local schemas is optional; however, it can be useful for the purposes of locally harmonising heterogeneous data and in order to better define its meaning; it can be done:
        - through a GUI when the number of schemas and attributes to define is low (less than a dozen of schemas, less than about a hundred attributes);
        - automated batch importing of OWL schema definitions, for larger numbers of schemas and attributes.

For the purpose of illustrating the knowledge management operations, the screenshots below show the GUI of the Knowledge Management Tools from IHT, the development of which has been finished upon the v1 release of the corresponding demonstrator deliverables [D5.16] and [D5.18].

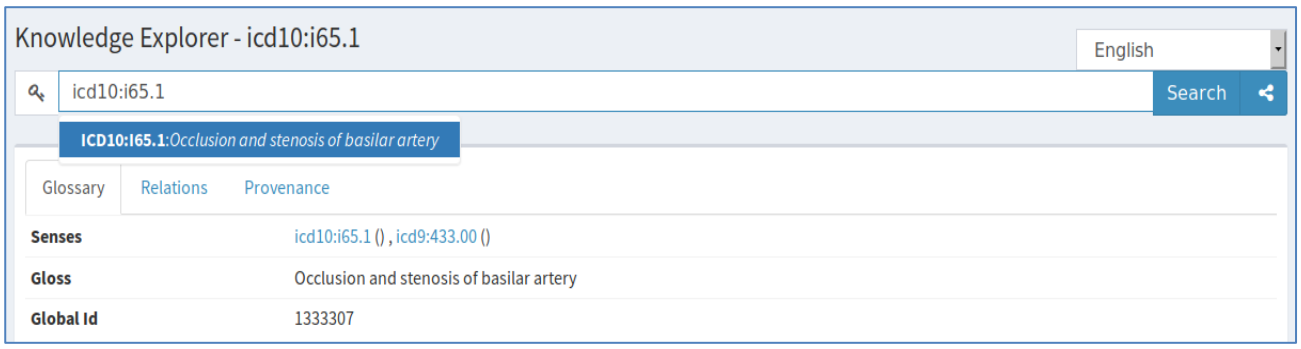

Figure 2: Screenshot from the concept and terminology explorer tool, showing a coded value from ICD-10, its meaning, its mapping to ICD-9, and its internal concept ID.

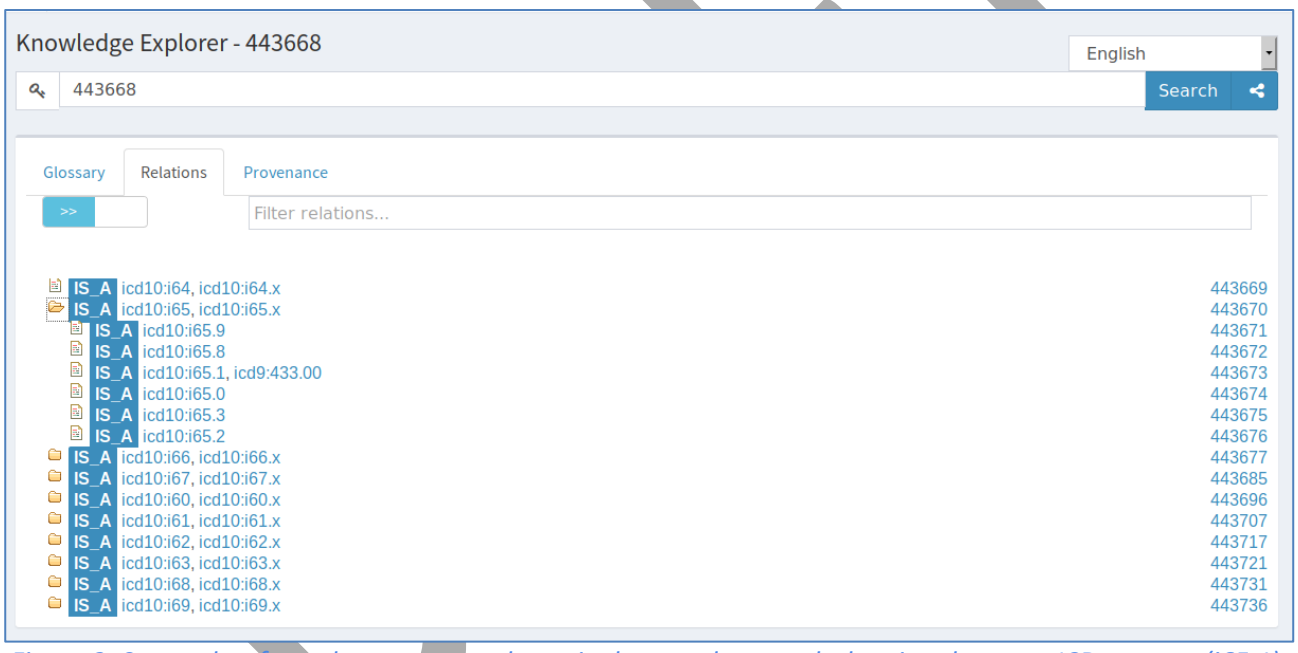

Figure 3: Screenshot from the concept and terminology explorer tool, showing the same ICD concept (i65.1) within the ICD hierarchy

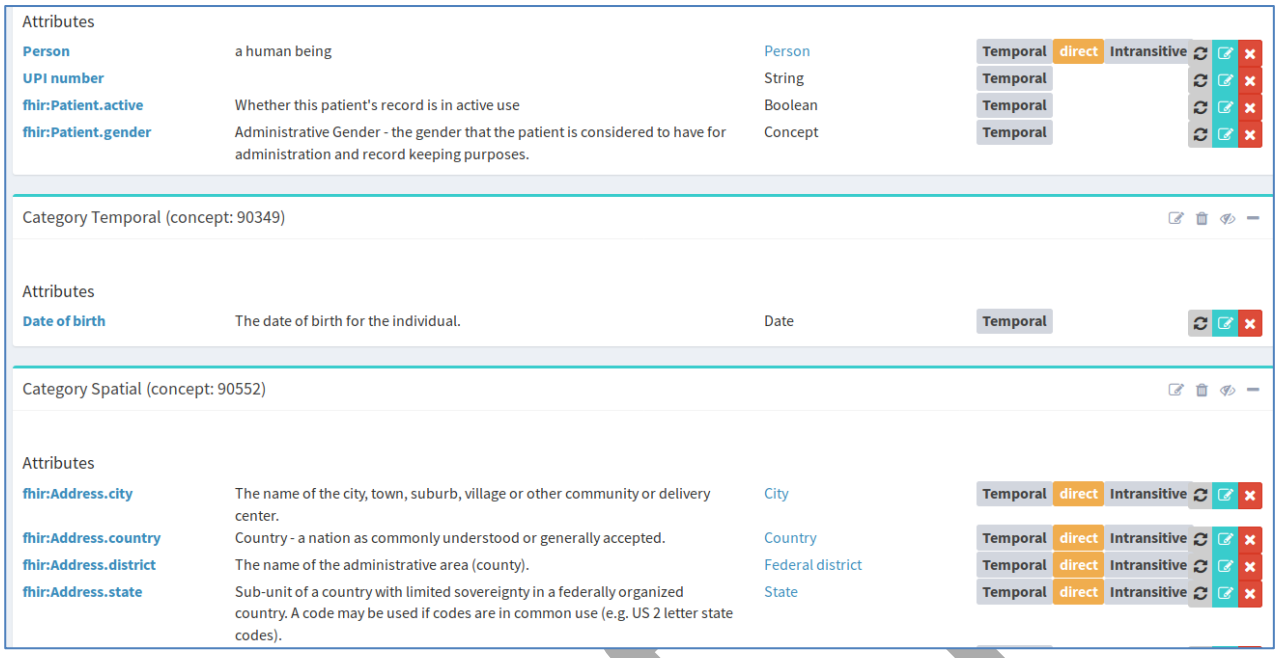

Figure 4: Screenshot from the entity type modeller tool

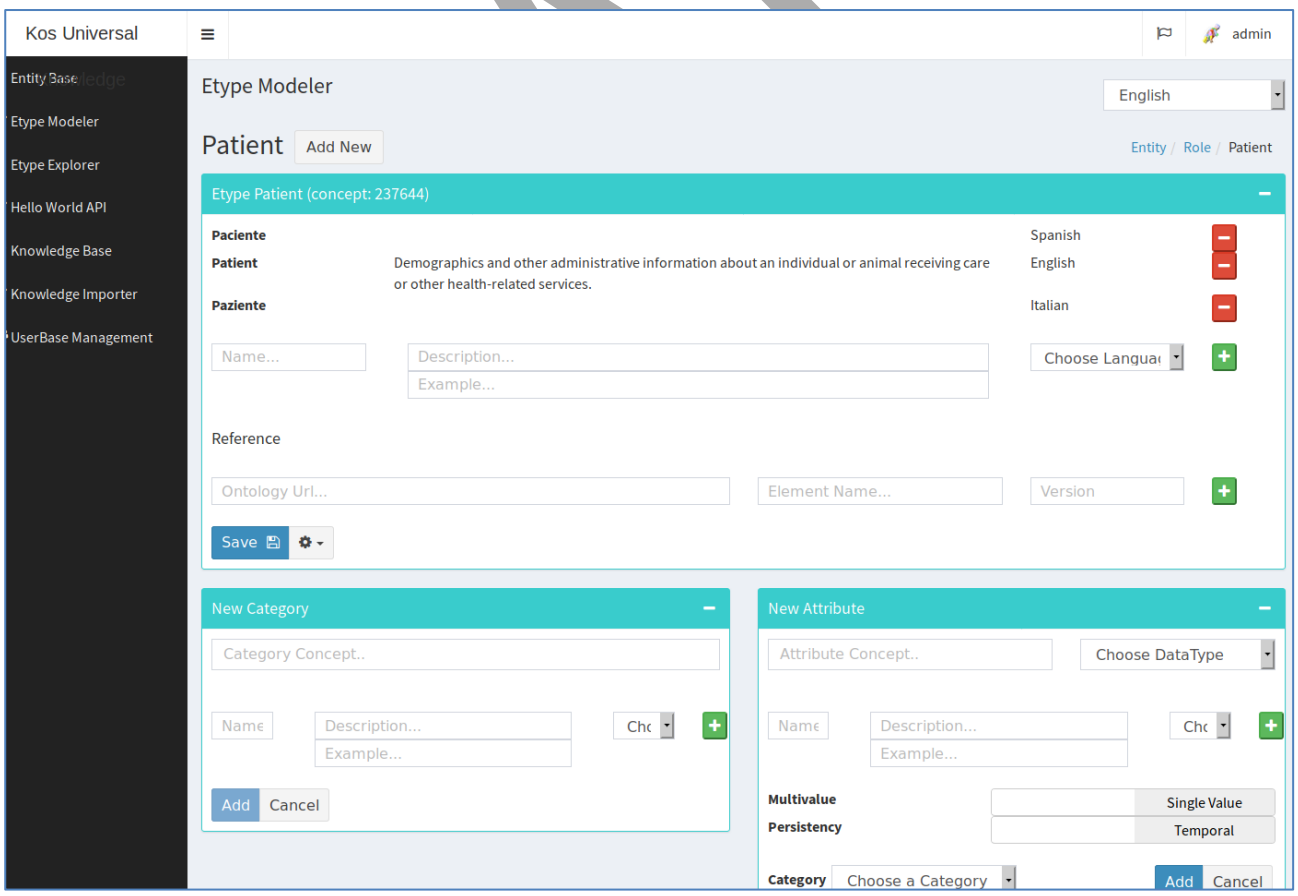

Figure 5: Screenshot from the entity type modeller tool

5.2. Data Mapping Setup

After setting up knowledge mappings, in this step the data integration, i.e. data transformations from local data formats to the FHIR-based Smart Health Data format, is defined. This process only needs to be done once, in the bootstrapping phase of the IHS. It is done through two steps:

1. **data extraction setup:** simple surface-level data format conversion in order to provide data to the Data Mapper Tool in a format that it accepts;
2. **data integration setup:** this is the step that performs the structural conversion to FHIR and also the meaning-level conversions of data values.

5.2.1. Data Extraction Setup

Data extraction consists typically of a simple script that converts "raw" health data (i.e., as extracted from local DBs) into the formats that are accepted by the following data integration step. This adaptation step is also necessary for reasons of data privacy and security: the data extraction code, typically implemented by the local institution, acts as a separation layer between local data/logic and the systems deployed as part of InteropEHRate.

The input of data integration, which is the output of data extraction, must respect the following requirements:

Category	Requirement
Format	CSV, XML, or JSON, always well-formed
Character encoding	UTF-8 is mandatory
Structure	Tabular (preferred) or tree-structured (acceptable).
Level of aggregation (number of attributes per file)	Due to the GUI-based approach, It is easier for the data scientist to work with multiple input files with a smaller number of attributes (10-20) rather than with one file containing hundreds of attributes.
Depth of nesting	In the case of tree-structured data, for a better understanding and usability by the data scientist in the data integration step, the levels of nesting should not be too deep (typically not more than 3-4 levels) and the tree should not be too complex. In the case of a wide and/or deep and/or complex dataset, splitting it into separate structures (files) greatly enhances usability. In this case, however, keys may need to be introduced in order not to lose information about data associations.
Number of records	Data integration can deal both with a single record and with multiple records in the same file.
Heterogeneity	While it is possible to deal with heterogeneity (the same kind of data, e.g. dates, expressed in different ways within a dataset) in the successive data integration step, the more of it is eliminated upstream, the less work the data scientist needs to do.

Table 2 - Input Requirements

5.2.2. Data Integration Setup

This is a crucial phase for the data integration process which allows to define the mappings between several local data formats and FHIR international data format adopted as interoperable format by InteropEHRate. In other words, in this step the knowledge, both local and international, extracted and formally defined in the previous steps, is linked to the data values of the health records, obtaining what is called a *data integration model* (or also *mapping model*) which contains all the mapping operations needed to convert a local version of a health record into the FHIR international version.

The data integration phase is set up through the graphical Data Mapper Tool that takes two inputs:

- access to the **knowledge** previously extracted and defined as described in section 4.1; this involves the international FHIR schemas, exported as an OWL file from the Data Integration Platform, as well as RESTful access to the knowledge defined within the Platform;
- the well-formed **data** previously extracted in the Data Extraction phase (section 4.2.1).

The data scientist in charge of this process creates the correct associations between the two inputs described above, performing the following operations, that can be divided in two different categories *syntactic* and *semantic*:

- The syntactic (mandatory) mappings operations:
    1. the data scientist identifies the FHIR resource(s) to which local data schemas need to be mapped;
    2. the data scientist maps the local data attributes to their respective FHIR attributes;
    3. the data scientist, through the functionalities offered by the Data Mapper Tool, transforms the data contained in each attribute in order to ensure FHIR compliance in terms of datatypes;
    4. the Data Mapper Tool also allows the data scientist to perform additional data cleaning operations, if necessary, such as the modification of date formats, numerical representations, or value conversions.
- The semantic (recommended) mappings operations:
    1. Coded values (such as the ICD-10 code “R58.X”), units of measure (such as “mg”) and terms (such as “parenteral”) can be converted from a string-based to a concept-based representation, linking them to concepts previously defined as knowledge. By specifying the language of the term, datasets in multiple languages can be correctly analysed and integrated. The advantages of concept-based representation are reduced ambiguity, reduced heterogeneity, and a more straightforward translatability of terms to other languages.
    2. The same coded values and healthcare terminology can be extracted from longer pieces of text.

Once all mappings are defined, the data scientist saves the entire mapping process as a *data integration model*. This model serves as a recipe for the subsequent automation of health data conversions. This means that the *model* containing all the operations performed by the data scientist, can be reapplied on the "same kind of data input". This does not mean that the input data must be exactly the same, but the *data integration model* can be reused automatically on input data defined using the same schema (same structure but different values) as the data input used to create the *model* at the first time. Thanks to this, a hospital which has all health records defined using the same schema, has to create the *data integration model* only once and then can re-apply it to all the health records it wants to convert.

The figure below shows a screen capture from the interactive tool used for defining the mappings, in the form that it was implemented for the v1 of the corresponding demonstrators [D5.16] and [D5.18].

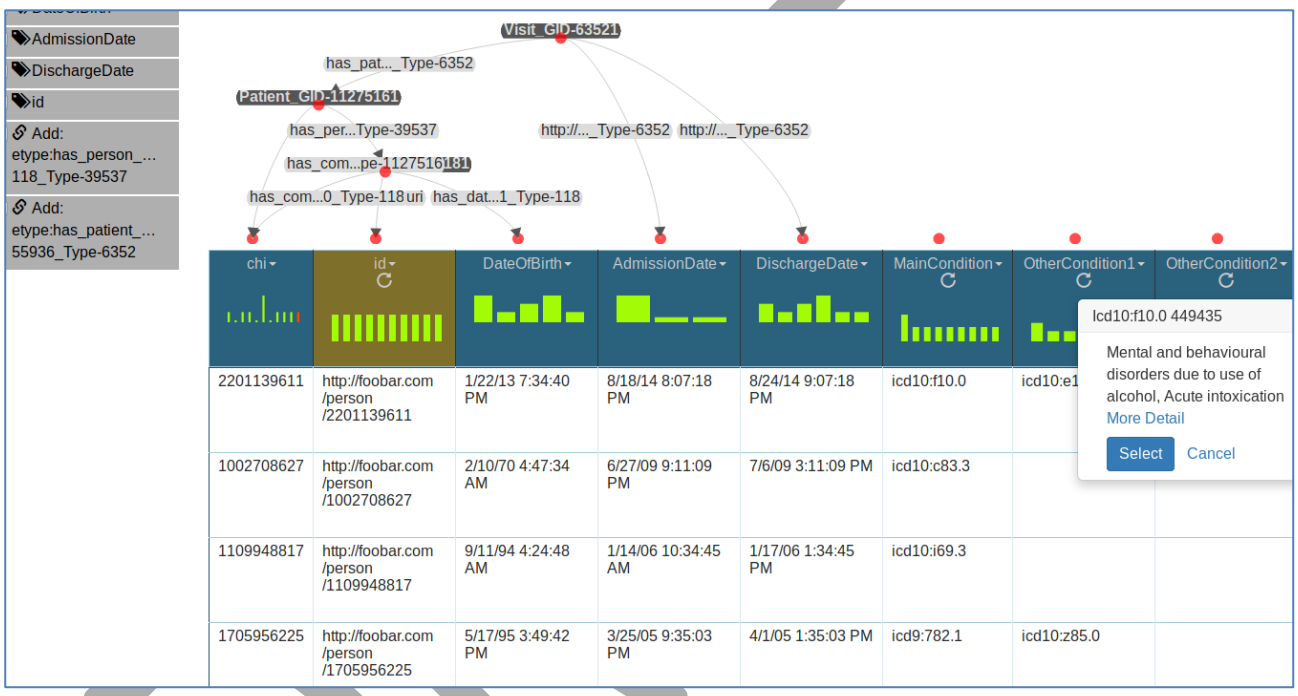

Figure 6: The Data Mapper Tool UI for performing data integration setup; part of the "recipe" is visible in the upper left corner

5.3. Automated Data Extraction and Integration

Once both the extraction script and the *data integration model* for data conversion have been generated in the previous step, they can be reapplied to any number of health records in a fully automated manner. This is achieved using the Conversion Services that, in turn, call the Data Mapper Tool in batch (non-interactive) mode. While the steps above are executed once and for all in a bootstrapping phase of the system, automated data extraction and integration is called for each Smart Health Data download request coming from a citizen.

All automated methods of the data conversion APIs are described in section 5 below. As defined in [D2.4], data extraction and integration are always executed at the source institution. This is because only the originating institution has sufficient knowledge of its own data representations to be able to define mappings precisely and to keep them up to date with the constant evolution of local data.

In case the input (legacy) health data format (structure and/or content) changes in a major way that is not compatible anymore either with the extraction script or the data integration model generated (e.g., the attribute names are modified), then the automated conversion may produce unexpected results or fail entirely. In such cases a pipeline maintenance step (see below) is necessary.

Section 6.4 describes in detail the automated data conversion process.

5.4. Knowledge and Data Mapping Maintenance for Data Evolution

The recipe-based automation of data integration is robust to a certain extent: it is able to deal with previously unseen data values (such as codes, text, or attributes) provided that such values are already encoded as formal knowledge. In case they are not encoded and the change is monotonic (the existing knowledge remains intact), the automated integration process will not fail: it will simply ignore data that it cannot recognise. In the case of non-monotonic changes, an explicit failure may occur.

The strength of the InteropEHRate data integration process, designed by UniTN, is that the system is able to follow most of the typical cases of data evolution by simple, agile changes to the underlying knowledge, as opposed to having to modify the source code through software development cycles. The table below lists the possible knowledge and/or data maintenance operations that may occur due to data evolution.

Kind of data evolution	Example	Maintenance operation	Executed by
New health term or coded value	A new health intervention code	Adding the new term as a new word, and also as a new concept if necessary, into the knowledge through the IHT Knowledge Management Tool.	Local data scientist
New or evolving encoding standard	Adoption of ICD-11	Adding the codes of the new standard as new words (and new concepts for previously non-existent meanings) into the knowledge through batch knowledge import.	CKR data scientist to formalise and publish the standard as formal knowledge, local data scientist to import it locally
New input data attribute	Addition of a new element in the history of the patient	Manual extension of the existing *data integration model* by mapping the new attribute using the Data Mapper Tool	Local data scientist
Evolving international data schema	evolution of FHIR v4 to FHIR v5	Extension of the FHIR pivot schemas (entity types) in the formal knowledge and update of the data integration model (attribute mappings) using the Data Mapper Tool	CKR data scientist
Major, disruptive change in the input format	The hospital DB schemas are completely reorganised	Rewriting of the data extraction script and manual generation of a new data integration model using the Data Mapper Tool	Local data scientist

Major, disruptive change in the international data schema	FHIR is replaced by another standard	Creation of new entity types in the formal knowledge representing the new international data schema, followed by the update of the data integration model using the Data Mapper Tool	CKR data scientist to formalise and publish the standard as formal knowledge, local data scientist to import it locally

Table 3 - Data Evolution and operations

6. INTERFACES AND IMPLEMENTATION OF DATA CONVERSION SERVICES

This section describes both the interfaces for accessing the conversion services, and the interfaces through which these services connect to the underlying Platform. Calls from external applications go through a series of API layers, from the most project-specific to the most generic.

1. Applications such as the HCP App should call the *IHS Interface* (IHSI) for high-level services that are based on standards and solutions adopted by the InteropEHRate project.
2. The *IHS Controller* takes such high-level calls and decomposes them into lower level *health data conversion* and *translation* calls through the *ConversionI* and *TranslationI* interfaces. These interfaces are more generic than IHSI in the sense that they are agnostic of the standard stack of InteropEHRate.
3. The *Conversion* and *Translation Services* implementation decomposes requests to *ConversionI* and *TranslationI* to low-level requests to the domain-agnostic *Data Integration Platform*.

The figure below provides the component diagram for the interfaces and components within the IHS.

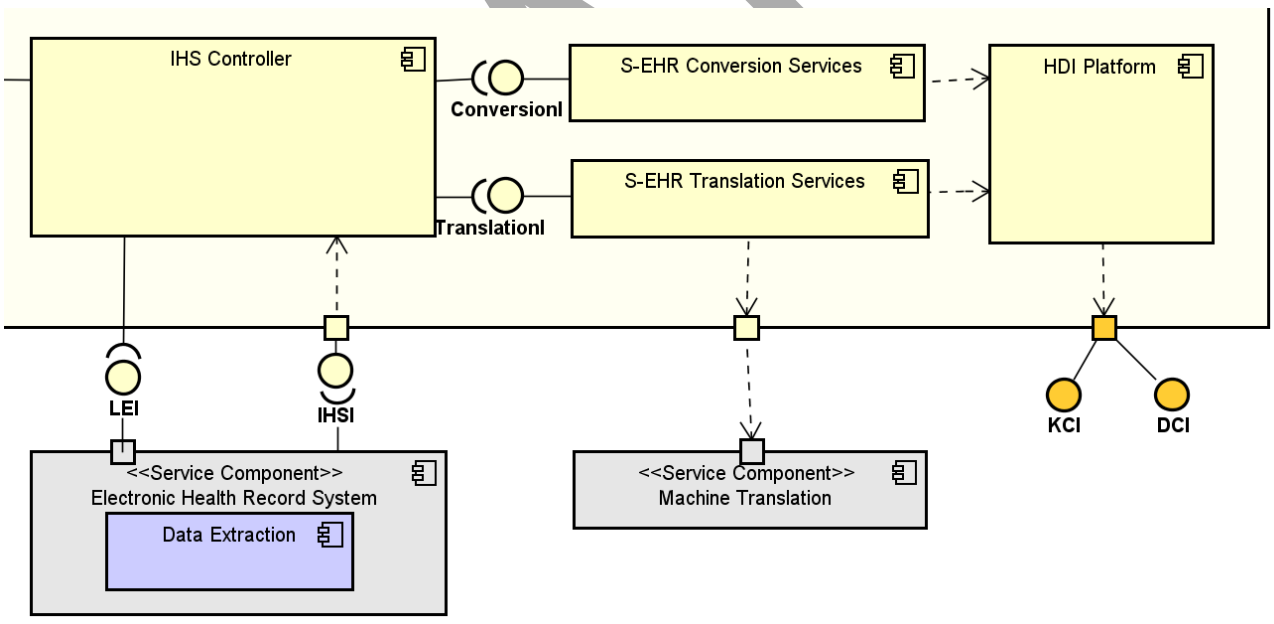

Figure 7: Relevant detail from the IHS component diagram showing the interfaces and components discussed in this section

Below we present the IHSI and ConversionI interfaces. The Platform interface is presented in the Platform deliverable [D5.8].

6.1. Interface Datatypes

The following table provides descriptions of the non-standard datatypes appearing in the *LEI* and *ConversionI* interfaces.

Datatype	Description
FHIRResource	Represents a FHIR resource or entire bundle, in one of its standard serialised forms, as supported by the interface.
DataIntModel	Represents a data integration model ("recipe"), i.e. a set of rules that allow the data integration process to be carried out automatically by the Conversion Services. The data integration model is generated by the Data Scientist using the Data Mapper Tool, as described in section 5.2.2.

Table 4 - Interface datatypes

6.2. The Conversion Interface (ConversionI)

The *Conversion Services* are a set of medium-level services provided through the *ConversionI* API. They are "medium-level" in the sense that, on the one hand, they hide technical details about the use of the Data Integration Platform and instead provide services that are meaningful from a healthcare organization perspective. On the other hand, they are not as high-level as the endpoints provided by the *IHS Controller* (described in [D5.8]) in order to remain adaptable to a wide variety of applications (applications used by medics, nurses, patients, or data scientists).

Accordingly, the implementation of the conversion services has a double role:

- it translates high-level requests for converting entire health records into a series of lower-level requests to the Data Integration Platform;
- it acts as an abstraction layer on top of the Platform, hiding its specific implementation details.

The API of the Conversion Services provides the following endpoints, called from the IHS controller but also callable by any third-party application needing to use the knowledge-based data conversion services:

ConversionI Endpoint	Input	Output	Example call URI	Description
POST /convertHealthData	File inputFile, DataIntModel model	Integer processID	/convertHealthData	Used to convert, file by file, a local health record, represented as a list of CSV/XML/JSON files, into an interoperable format as defined by the input data integration model (see Section 5.2.2 for the data integration model definition). This call returns the processID of the integration process in charge of integrating the files.

GET /checkIntegrationStatus	Integer token, (Optional) Integer logDetailLevel	boolean result, JSON logFile	/checkIntegrationStatus?token=9876&logDetailLevel=2	(Polling call) Called by the IHS periodically to check the status of the integration process identified by a "token" representing the process ID. This call returns a boolean value equal to true if the process is completed and false otherwise. Moreover, if the logDetailLevel input parameter is specified, the requested log file is part of the output.
GET /retrieveConvertedFHIR	String ehrID, String targetLang	FHIRResource bundle	/retrieveConvertedFHIR?ehrID=1234&targetLang=ita	Retrieves the already converted health record specified through its local patient ID from the EHR Data Cache, and returns it in FHIR format and translated to the language lang
POST /convertTerms	String[] srcTerms, String srcStd, String trgStd	String[] trgTerms	/convertTerms	converts a list of source terms (e.g., codes) from a source standard to its equivalents, if they exist, in the target standard

Table 5 - ConversionI endpoints

Some of the parameters reported in the table above, need a more detailed description, in order to understand their purpose. Those parameters are:

- token: this is an integer value equivalent to the process ID returned by the call of *convertHealthData* that initiates the data integration process. Due to the fact that the whole conversion is an asynchronous process, the process ID is provided to the client (in this specific case represented by the IHS Controller) in order to monitor the status of the integration process through the polling call *checkIntegrationStatus.*
- logDetailLevel: this is an optional parameter represented by an integer value in the range [1, 2] which specifies the level of information detail to be included in the log file returned, only if this parameter is specified. More precisely if this parameter is equal to 1, the log file will include the information regarding the data integration process, while instead if the value is equal to 2, the log file includes the information regarding both the integration process and the import process in charge of storing the converted data within the EHR data cache [D5.8].

The services provided by the Conversion Interface produce health records in which the original information is maintained together with the conversions performed (i.e. both the local medical codes and the respective interoperable versions are maintained in the same health record). This allows the sharing of health records without losing information. It also allows the verification of the correctness of the conversions performed.

This feature is implemented within the conversion services, adding in the resource structure the converted information without replacing the original one. This is the specific case in which the Conversion service operates on structured portions of the FHIR structure, defined by the interoperability profiles, containing the information that has to be converted. More precisely the FHIR attribute used is *coding*. An example of this is reported in the figure below.

```
"coding": [
  {
    "system": "http://hl7.org/fhir/sid/icd-9",
    "code": "428.1",
    "display": "Insufficienza del cuore sinistro (scompenso cardiaco Sinistro)"
  }
]
```

Figure 8: coding FHIR attribute example.

This FHIR attribute contains a list of elements used to define medical standard codes. The conversion service, in order to keep both the original information and the converted one, adds a new element to the *coding* list containing the converted version of the original code. The result of the conversion applied to the information in Figure 8 is reported in Figure 9 below, where the interoperable version of the icd9 medical code (icd10 is the medical standard used at international level) is added to the *coding* attribute list.

```
"coding": [
  {
    "system": "http://hl7.org/fhir/sid/icd-9",
    "code": "428.1",
    "display": "Insufficienza del cuore sinistro (scompenso cardiaco Sinistro)"
  },
  {
    "system": "icd10",
    "code": "i50.1",
    "display": "Left ventricular failure"
  }
]
```

Figure 9: Example of *coding* FHIR attribute with conversion.

6.3. Health Record Conversion Sequence Diagram

This section provides details over the API calls relative to the automated conversion process introduced in section 5.3. The diagram below shows how the interfaces above are called to implement the conversion of a legacy health record into Smart Health Data.

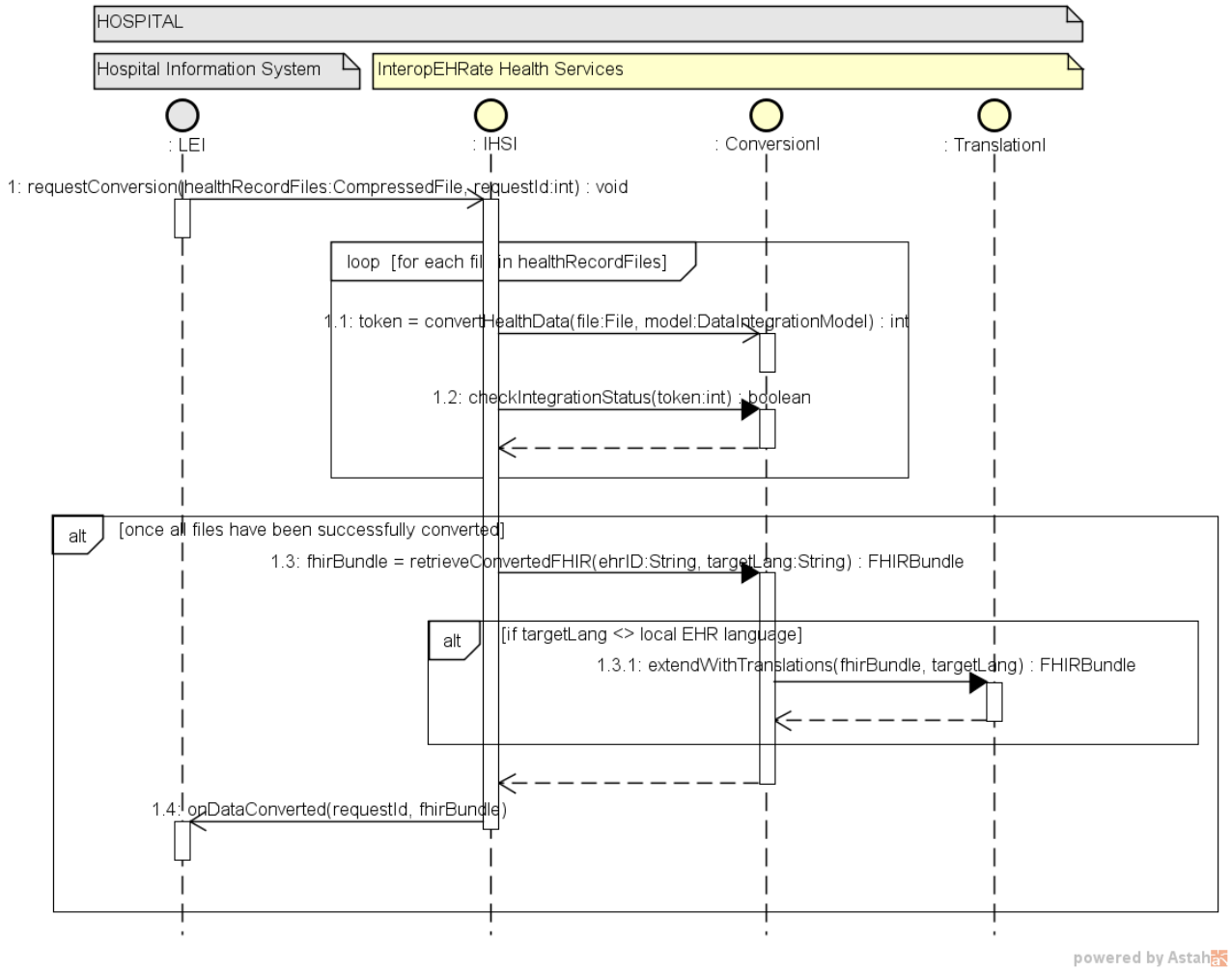

Figure 10: Sequence diagram of a request from a S-EHR App to retrieve an existing health record, convert it into a Smart Health Record, and optionally translate it to the patient's language

The automatic data conversion process is executed as follows:

1. An application or service running within the premises of the healthcare organisation but outside of the InteropEHRate Health Services, in charge of providing health records to be uploaded onto citizens' S-EHR Apps, calls the *requestConversion* endpoint of the IHSI with the legacy health record within an input compressed file.
2. The IHS Controller converts each file within the compressed bundle, one by one, according to the rules defined by its internal configuration. Conversion is implemented as a pair of two calls made to the Conversion Services:
    a. the asynchronous *convertHealthData* method sends the structured input file to the Data Integration Platform, together with the *data integration model* ("recipe") that is

appropriate for that file. The appropriate recipe is chosen based on the IHS Controller configuration (see [D5.8] for details); this is represented as an IHS-internal *getRecipe* call in the figure above. The *convertHealthData* call returns a token which allows the IHS to poll the Conversion Services (using the *checkIntegrationStatus* call) to find out whether the conversion process has terminated.

   b. Upon success, the integrated health record is stored in the internal *EHR Data Cache* of the Data Integration Platform, as a formal knowledge graph. The *retrieveConvertedFHIR* call serves to retrieve the standard, FHIR-based, serialised bundle of the entire Smart Health Record from the EHR Data Cache.

3. If the outside application asked for translation into the patient's language, the IHS Controller sends the FHIR bundle to the Translation Service, indicating the target language. The Controller also provides an interoperability level, which indicates to the Translation Service the translation method(s) to apply (i.e. in case of syntactic interoperability, only machine translation is applied).
4. Finally, the converted (and possibly translated) FHIR bundle is returned to the hospital's EHR information system by calling the *onDataConverted* method of the hospital's LEI interface (details in [D5.8]).

7. INTERNATIONAL KNOWLEDGE AND ITS MAPPINGS

This section presents the *international knowledge* content developed for specific IHS installations in preparation for the project scenarios. By *international knowledge* we understand 'pivot' terminology, concepts, and data structures used for interoperability within SEHRs. The standards adopted for international knowledge have been defined in deliverable [D2.7]. The final version of the current deliverable (v2, D5.10) will describe the implementation of the following standards:

- data structures:
    - FHIR v4.0 profile for International Patient Summary,
    - other FHIR profiles as required by the three project scenarios;
- concepts:
    - FHIR concepts,
    - ICD-9 and ICD-10 concepts,
    - SNOMED CT International concepts,
    - LOINC concepts,
    - UCUM concepts,
    - ICNP concepts,
    - general language concepts,
- natural language labels (in English, French, Romanian, Greek, and Italian):
    - FHIR labels,
    - ICD labels,
    - SNOMED CT labels,
    - LOINC labels,
    - UCUM labels,
    - ICNP labels,
    - general language labels;
- entities (structured instances):
    - ATC pharmaceutical substances (drugs).

UniTN adopts an iterative methodology for developing support for these standards. Such a methodology adapts to the general project management methodology where development has to proceed based on partial information. The formalisation of all international knowledge above, except for the translation of language-specific labels, is being carried out by a UniTN data scientist with experience in healthcare informatics. The results will be reviewed by healthcare provider partners from the project. In the case of natural language labels where translations do not exist in any of the four target languages (Romanian, Greek, Italian, French), the help of local domain experts from among the project partners will be necessary.

7.1. FHIR IPS Support

FHIR v4.0 IPS (http://hl7.org/fhir/uv/ips/2019Sep/) is an international standard for representing *patient summaries* in a cross-border interoperable form. It is the FHIR-based version of the *International Patient*

Summary as defined by HL7. According to the methodology described in section 4, the FHIR IPS data structures will be represented as three separate levels of knowledge:

- *natural language labels* of the data attribute and resource names in all five supported languages;
- *concepts* representing the meanings of data attributes and entity types (FHIR resources);
- *entity types* (data structures) themselves, represented in a language-independent manner using concepts only.

FHIR IPS concepts (there are over 500 of them) and entity types (there are over 20 of them) will be described manually by a UniTN data scientist. The concepts will be mapped to equivalent SNOMED CT concepts whenever such an equivalence exists. The corresponding English labels will be taken from the original FHIR specifications. As FHIR is not available in the remaining four languages, translated labels in these languages will have to be provided (for all concepts or at least a subset of the concepts) by local domain experts.

7.2. ICD Support

The *International Classification of Diseases* (https://www.who.int/classifications/icd/en/) is an international standard used in all project partner countries. While its most recent version is 11, most countries are still using version 10, while Italy is using version 9. For this reason, in the project both ICD-10-CM and ICD-9-CM will be used and formally represented:

- ICD-10-CM will be used to define the concept hierarchy;
- ICD-10-CM codes will be defined as natural language labels in all supported languages, prefixed by the standard identifier (e.g., “icd10_i65.1”);
- natural language descriptions (e.g., “Occlusion and stenosis of basilar artery”) in all supported languages will be used as definitions (“glosses”) for the codes; the translations of ICD-9 or ICD-10 (whichever applicable) will have to be provided to the project by national healthcare provider partners;
- equivalent ICD-9-CM codes will be mapped as synonymous labels to the existing ICD-10-CM concepts (e.g., “icd9_433.00” and “icd10_i65.1” are synonyms).

Definitions of the ICD-9-CM and ICD-10-CM hierarchies were provided by the *Centers for Disease Control and Prevention (https://www.cdc.gov), Department of Health & Human Services, USA*. Mappings between ICD-9 and ICD-10 were provided by the *Centers for Medicare & Medicaid Services (CMS) (https://www.cms.gov)*, part of the *Department of Health and Human Services (HHS), USA*.

7.3. SNOMED CT Support

SNOMED CT (http://www.snomed.org/) is becoming a general-purpose reference terminology for interoperability in the healthcare field. Therefore, the project is adopting SNOMED CT both as one of its “pivot” terminologies and, wherever applicable, as a bridge between more specialised terminological resources.

For the concept layer, the latest *International Full* version of SNOMED CT will be used. For the language layer, SNOMED CT provides English labels but none of the other languages of the project are supported yet by SNOMED. Therefore, labels in these languages will either not be provided or only if translated within the project on a case-by-case basis (the translation of the hundreds of thousands of SNOMED CT terms is obviously out of the scope of the project).

7.4. LOINC Support

The project will support LOINC version 2.66 (https://loinc.org). The entire LOINC of around 94.000 concepts will be formalised and imported in a way similar to ICD, with "loinc_" prefixes used in front of labels representing codes. As from https://loinc.org/international/, translations of LOINC labels seem to be available in Italian, Belgian French, Greek, but not Romanian. Therefore, the possibility of having Romanian translations for LOINC needs to be investigated.

7.5. ATC Support

The *WHO Anatomical Therapeutic Chemical Classification System* (https://www.whocc.no/) is an international coding system for pharmaceutical substances. ATC is widely used for interoperability through the mapping of national coding systems (such as AIC for Italy or BNF for the UK). Contrary to other coded values, ATC codes will be encoded as *entities*: this allows the representation of rich structured knowledge about substances beyond their code and their name, such as the *defined daily dose*.

7.6. UCUM Support

The *Unified Code for Units of Measure* (https://unitsofmeasure.org/trac) is a code system intended to include *all* units of measure used in international science, engineering, and business. UCUM codes are frequently used in the healthcare domain in resources such as medical prescriptions or for reporting vital signs and other parameters such as blood pressure. By supporting UCUM, the data exchanged among healthcare providers can be standardized and described more precisely.

7.7. General Language Support

As EHRs often contain words and expressions from general language that are not covered by domain-specific terminologies, health knowledge is extended with the words and word senses from the general vocabulary. For this *wordnets* will be used for all five supported languages. Wordnets define the most widely used nouns, verbs, adjectives, and adverbs from the general vocabulary. Our source of multilingual wordnets is the *Open Multilingual Wordnet* (http://compling.hss.ntu.edu.sg/omw/).

8. RELATED WORK

Mapping of terminology and coded values. The need to perform mappings or "crosswalks" across coding systems and term bases has existed for a long time. It was made necessary not only by the coexistence of overlapping codifications (such as diseases encoded both in SNOMED CT and in ICD) but also by disruptive changes across different versions of the same standard (such as ICD-9 versus ICD-10). Such mappings, that more often than not are complex and involve many-to-one relationships, subsumption (broader-narrower), and even approximations, have been manually produced and validated by domain experts for obvious reasons of precision. A well-known aggregator of mapping knowledge is the UMLS Metathesaurus [UMLS], a useful source for such information. The InteropEHRate mapping solution presented in this deliverable also reuses and aggregates expert-sourced mapping information, and is able to represent both equivalence and subsumption relations. What the InteropEHRate solution offers beyond a mere aggregation of knowledge is support for its customisation and evolution. The *InteropEHRate Knowledge Management Tools* allow data scientists working at hospitals to extend the knowledge with locally relevant terminology and their mappings to international standards in an agile manner, as well as to update it whenever the local or the international terminologies or codes evolve.

Schema mapping and data transformation. Schema mappings and data transformations are an essential part of any data integration solution. Traditionally, they were implemented programmatically as custom business logic developed "once and for all". The downside of such an approach is the high cost of maintenance, as the slightest change in either the source or the target schemas requires the opening of a new software development cycle. In order to accelerate the process, so-called graphical *ETL (Extract-Transform-Load) tools* have been proposed that allow the definition of schema mappings and transformations in an interactive manner. Many of these tools are domain-agnostic [OpenRefine, Karma, Talend, Pentaho, Oracle] (there are too many of them to provide an exhaustive list), others, such as NextGen Connect [NextGen], are verticalized to the healthcare field and come with pre-loaded mappings between well-known standard formats. Once the mapping and transformation rules have been defined, they can be executed automatically in batch over large amounts of data. The InteropEHRate *Data Mapper Tool*, an extension of the open-source Karma, is a domain-agnostic tool that provides similar functionalities to the ones mentioned above, with one major improvement not found in any other such tool: support for multilingual semantic data transformations. The Data Mapper Tool employs an extensible multilingual NLP engine in its back-end that can parse natural language labels in data and, relying on knowledge stored in the Data Integration Platform, annotate terms and names by their meanings. While annotation is automatic, it can be governed and constrained by a human for better precision.

9. CONCLUSIONS AND NEXT STEPS

This document provided the final (v2) specifications for the InteropEHRate data mapping and conversion services in support of cross-border data interoperability. The services are intended to be deployed within the information systems of healthcare institutions. This implies (1) the use of an underlying semantic platform exploited by these services; (2) the adaptation of the services to the local technological context; as well as (3) the adaptation to local and international data representation standards and practices. This deliverable covered the third aspect, while points (1) and (2) are covered in the closely related deliverable [D5.8].

The use of the data mapping and conversion will be demonstrated in all pilots of the InteropEHRate project, as the conversion of health record data to an interoperable representation is a prerequisite in all project scenarios. The tools and background knowledge implementing the conversion operations will be made available as part of the InteropEHRate software releases.

REFERENCES

- **[D2.3]** InteropEHRate Consortium, InteropEHRate Requirements v3 (deliverable D2.3). https://www.interopehrate.eu/resources/
- **[D2.6]** InteropEHRate Consortium, InteropEHRate Architecture v2 (deliverable D2.6). https://www.interopehrate.eu/resources/
- **[D2.8]** InteropEHRate Consortium, Interoperability Profile v2 (deliverable D2.8). https://www.interopehrate.eu/resources/
- **[D5.8]** InteropEHRate Consortium, Design of the Data Integration Platform v2 (deliverable D5.8). https://www.interopehrate.eu/resources/
- **[D5.11]** InteropEHRate Consortium, Design of information extractor and natural language translator v1 (deliverable D5.11). https://www.interopehrate.eu/resources/
- **[D5.16]** InteropEHRate Consortium, Data integration platform for healthcare professionals - v1 (deliverable D5.16). https://www.interopehrate.eu/resources/
- **[D5.18]** InteropEHRate Consortium, Library and tool for data mapping and conversion - v1 (deliverable D5.18). https://www.interopehrate.eu/resources/
- **[UMLS]** The UMLS Metathesaurus. https://www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/index.html
- **[NextGen]** NextGen Connect (formerly Mirth Connect). http://github.com/nextgenhealthcare/connect/
- **[OpenRefine]** The OpenRefine data transformation tool. https://openrefine.org/
- **[Karma]** The Karma data integration tool. https://usc-isi-i2.github.io/karma/
- **[Talend]** Talend Open Studio for Data Integration. https://www.talend.com/products/talend-open-studio/
- **[Pentaho]** Pentaho Data Integration. https://www.hitachivantara.com/en-us/products/data-management-analytics/pentaho-platform/pentaho-data-integration.html?source=pentaho-redirect
- **[Oracle]** The Oracle Data Integrator. https://www.oracle.com/middleware/data-integration/management-pack-for-odi/