D5.20

Library for information extraction and translation - V1

ABSTRACT

This deliverable provides installation and usage instructions for the InteropEHRate demonstrators related to multilingualism in health record data. Namely, it covers the demonstration of translation and of information extraction from unstructured multilingual text. Demonstration is provided both directly through stand-alone calls to the services and indirectly through an end-user application, namely the HCP App, that relies on translation and information extraction to display health record content to healthcare practitioners.

Delivery Date	21st June, 2021
Work Package	WP5
Task	T5.5
Dissemination Level	Public
Type of Deliverable	Demonstrator
Lead partner	UNITN

This 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	Simone Bocca	Partner
Contributors	Gábor Bella	Partner

LOGTABLE

Version	Date	Change	Author	Partner
0.1	17-03-2021	Document created	Gábor Bella	UNITN
0.2	20-03-2021	Added technical content	Simone Bocca	UNITN
0.3	16-06-2021	Prepared for internal review	Gábor Bella	UNITN
Vfinal	18-06-2021	Final check and submission	Laura Pucci	ENG

ACRONYMS

Acronym	Term and definition
API	Application Programming Interface
CT	Concept Translation
FHIR	Fast Healthcare Interoperability Resources: the international healthcare data representation standard adopted by the InteropEHRate project
HCP	Healthcare Practitioner
HCP App	Healthcare Practitioner Application: an end-user application for managing patient health records, used by healthcare practitioners; one such application is delivered by the InteropEHRate project
ICD	International Classification of Diseases
IHS	InteropEHRate Health Services: a high-level service component provided to healthcare institutions by the InteropEHRate project
JSON	A common file format for representing structured data
LOINC	Logical Observation Identifiers Names and Codes
MT	Machine Translation
SNOMED CT	An international terminology base for clinical terms

TABLE OF CONTENTS

1. INTRODUCTION 1
1.1. Scope of the document 1
1.2. Intended audience 1
1.3. Structure of the document 1
1.4. Updates with respect to previous version (if any) 1
2. SW DESCRIPTION 2
3. OVERVIEW 3
3.1. Building Guide 3
3.2. Installation Guide 4
3.3. User guide 4
3.4. Other related documentation (if any) 8

LIST OF FIGURES

Figure 1 - Concept translation example within a raw FHIR resource
Figure 2 - Machine translation example within a raw FHIR resource
Figure 3 - Screenshot of the HCP App showing data in its original language and the relative translation version

LIST OF TABLES

Table 1 – Translation Library Software Description

1. INTRODUCTION

1.1. Scope of the document

This document accompanies a software demonstrator deliverable available at on the main InteropEHRate project software repository: http://iehrgitlab.ds.unipi.gr.

It provides installation and usage instructions for the deployment and running of the demonstrators that showcase the translation of textual content within health records, as well as the information extraction from unstructured pieces of health record text in multiple languages. As information extraction is still work in progress at the time of writing the first version of this deliverable, this first version only covers the demonstration of translation. Demos on information extraction will be added in the subsequent version of this deliverable[D5.21].

1.2. Intended audience

This document is intended for hospital IT specialists (e.g. system developers or administrators) wishing to try out InteropEHRate technologies, as well as for non-technical users who wish to get a quick impression of the capabilities of the translation and information extraction technologies through the examples and demo screenshots provided in this document.

1.3. Structure of the document

Section 2 provides a schematic description of the translation software component. Section 3 describes how the Translation library works, and provides to the reader the building, installation and user guide for the translation software component. Some concrete examples of translation using the InteropEHRate Translation Services are included in section 3. The next versions of the current deliverable will contain a dedicated section for the information extraction software component.

1.4. Updates with respect to previous version (if any)

Not applicable as this is the first version of the deliverable.

2. SW DESCRIPTION

SW TITLE	Translation Library
SW VERSION	0.1
LICENCES AND PATENTS	Apache License
PROGRAMMING LANGUAGES	JavaScript (Node.js environment)
SUPPORTED PLATFORM(s)	Unix based OS, Node.js environment
http://iehrgitlab.ds.unipi.gr/interopehrate/health-services/machine-translation	http://iehrgitlab.ds.unipi.gr/interopehrate/health-services/label-translation
EXECUTABLE	N.A.

Table 1 – Translation Library Software Description

3. OVERVIEW

The Translation library, integrated in the InteropEHRate framework, allows the translation of the information contained in the health records exchange across the European context in which the project operates. The functionalities offered by the library are available through a specific component of the InteropEHRate framework, called IHS Controller [D5.8] that acts as a middleware between calls from external applications (such as the HCP App) and health data translation functionalities. The Translation library, as described in [D5.11]] contains two different components dedicated to the translation of two different kinds of data included within the health records. The two translation components offered are described as follows:

- Machine Translation (MT) component: The MT is the translation library component in charge of translating those portions of the health records that include a small portion of *free-text*, or in other words, text expressed in natural language. These portions of health records managed by the MT are identified thanks to the FHIR interoperability profiles adopted in the project to describe the health data involved. The FHIR structure allows the MT to operate on the health records attributes that contain the *free-text* value. The MT exploits the functionality of an external text translation service. The above-mentioned external service adopted, in the first version of the component, is the IBM Watson Language Translator, which provides the translation of one million characters per month for free.
- Concept Translation (CT) component: The CT, similarly to the MT component, is able to identify specific attributes in the FHIR structure of the health records managed. Unlike MT, this component operates on FHIR attributes that includes *concepts*, more in specific concepts used in the health domain in which InteropEHRate works (medical standards such as SNOMED-CT, ICD-9, ICD-10, LOINC, and others listed in [D5.8]). The concept values are recognized in the health records, thanks to the support of a Data Integration (DI) Platform [D5.8] which stores a set of multilingual knowledge resources used to define such medical concepts, allowing their identification and translation. The CT component exploits the functionality offered by the DI platform, through an internal API layer, in order to obtain the translated version of the concepts analysed, and adding such translated value in the health record.

The two components of the Translation library are executed together on the health records that have to be translated, in order to exploit the translation of both the *free-text* and concepts values of the attributes included in the FHIR health record. A translated health record always contains the original version of the attributes translated, together with the translated values. In this way the original data are always present in the health record and can be retrieved in case of errors during the translation procedure.

The Information Extraction library will be described in the next version (v2) of the current deliverable.

3.1. Building Guide

The IHS controller component included in the InteropEHRate framework, is a system developed in JavaScript running in Node.js environments. Due to that also the libraries contained in, and accessible by, the IHS Interface, are developed in JavaScript. The features of the Translation library are provided as RESTful services accessible by the applications that need the data translation. In order to work properly, the Translation services have to communicate with the DI platform that has to be correctly installed and set up in the same machine where the IHS controller is running. The DI platform has to be previously installed and

set up, in a Docker environment, through a docker-compose file which defines the docker containers structure and execution, required by the DI platform to work properly, as well as the communication between those containers. Once the DI platform is correctly deployed and running, in order to allow the platform features support, the JavaScript files defining the API layer that allows the communication between the IHS and the DI platform, has to be present in the IHS working directory.

3.2.Installation Guide

In order to provide the usage of the Translation library, the IHS controller, as well as the DI platform, have to be correctly installed and executed in the host (Unix based) machine.

The DI platform works in a Docker environment, and it can be installed mounting the docker containers defined in the DI platform docker-compose file, executing the following command in the directory in which the docker-compose file is stored:

```
docker-compose up
```

In order to support the Concept Translation component of the Translation library, the DI platform has to include the knowledge resources required to identify and translate the medical concepts included in the health records. For this reason, before exploiting the features provided by the Translation library, such resources have to be imported in the platform as specified in [D5.8].

Once the DI platform is correctly set up, some configurations are needed before running the IHS controller. Those configurations are specified in a specific file contained within the IHS working directory, following the path: */config/default.json* . The configuration parameters requested are detailed in [D5.8]. To be sure that the Node.js environments is correctly configured the following command can be used executed first:

```
npm install
```

then the command to run the IHS controller:

```
node IHSI.js
```

3.3.User guide

Once the IHS controller and the underlying DI platform are correctly deployed, the services of the Translation library are accessible as RESTful API. The application that request the data translation (like the hospital HCP application) has to implement the management of the call to the translations service, defined as follow:

```
POST -d @<JSON-FHIR-resource-to-translate> http://<host-address>:<service port-
number>/extendWithTranslation?targetLang=<language code> --header 'content-type:
application/json'
```

where:

- <JSON-FHIR-resource-to-translate> : is the JSON file that has to be translated. This file contains a FHIR bundle resource where the health record is defined following the FHIR profiles defined by InteropEHRate.
- <host-address> : is the address of the machine where the IHS controller is running.

- <service-port-number> : is the number of the port of <host-address>, where the IHS Interface is accessible.
- <language-code> : is a two-character code indicating the language into which the health data have to be translated.

Similarly, if the application needs to use the MT component or the CT component only, translating singularly only the *free-text* attributes or the concepts attributes, it can implement the management of the following two REST service call, respectively:

- `POST -d @<JSON-FHIR-resource-to-translate> http://<host-address>:<service port-number>/machineTranslation?targetLang=<language code> --header 'content-type: application/json'`
- `POST -d @<JSON-FHIR-resource-to-translate> http://<host-address>:<service port-number>/conceptTranslation?targetLang=<language code> --header 'content-type: application/json'`

where the parameters involved are the same as the unified translations service call, described above, called *extendWithTranslation.*

The following figures show how concretely the calls listed above work on an example of a medical resource represented by a FHIR-JSON file describing medication instructions (FHIR MedicationRequest resource type). The first and the second figures show the same response of a translation service call (/extendWithTranslation) where the two kinds of translations are highlighted; the Concept Translation (Figure 1) and Machine Translation (Figure 2).

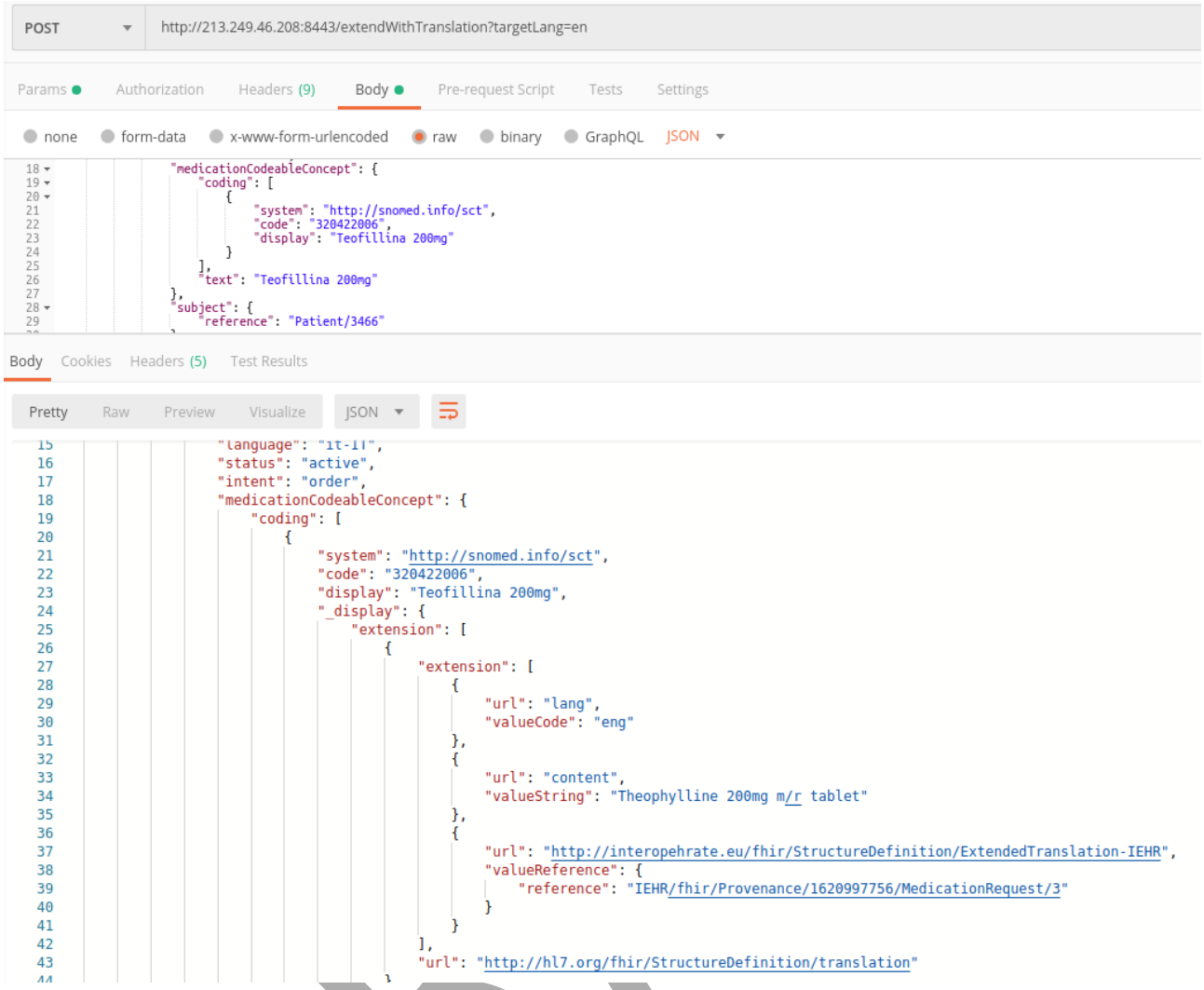

Figure 1 - Concept translation example within a raw FHIR resource

In Figure 1, the original medication information, expressed through a specific SNOMED standard code with an Italian description (upper JSON file portion in figure), has been extended in the translation call's response (lower JSON file portion in figure) by adding the extra attribute '*_display*' containing the language extension that includes the English version of the same conceptual (SNOMED standard code) information.

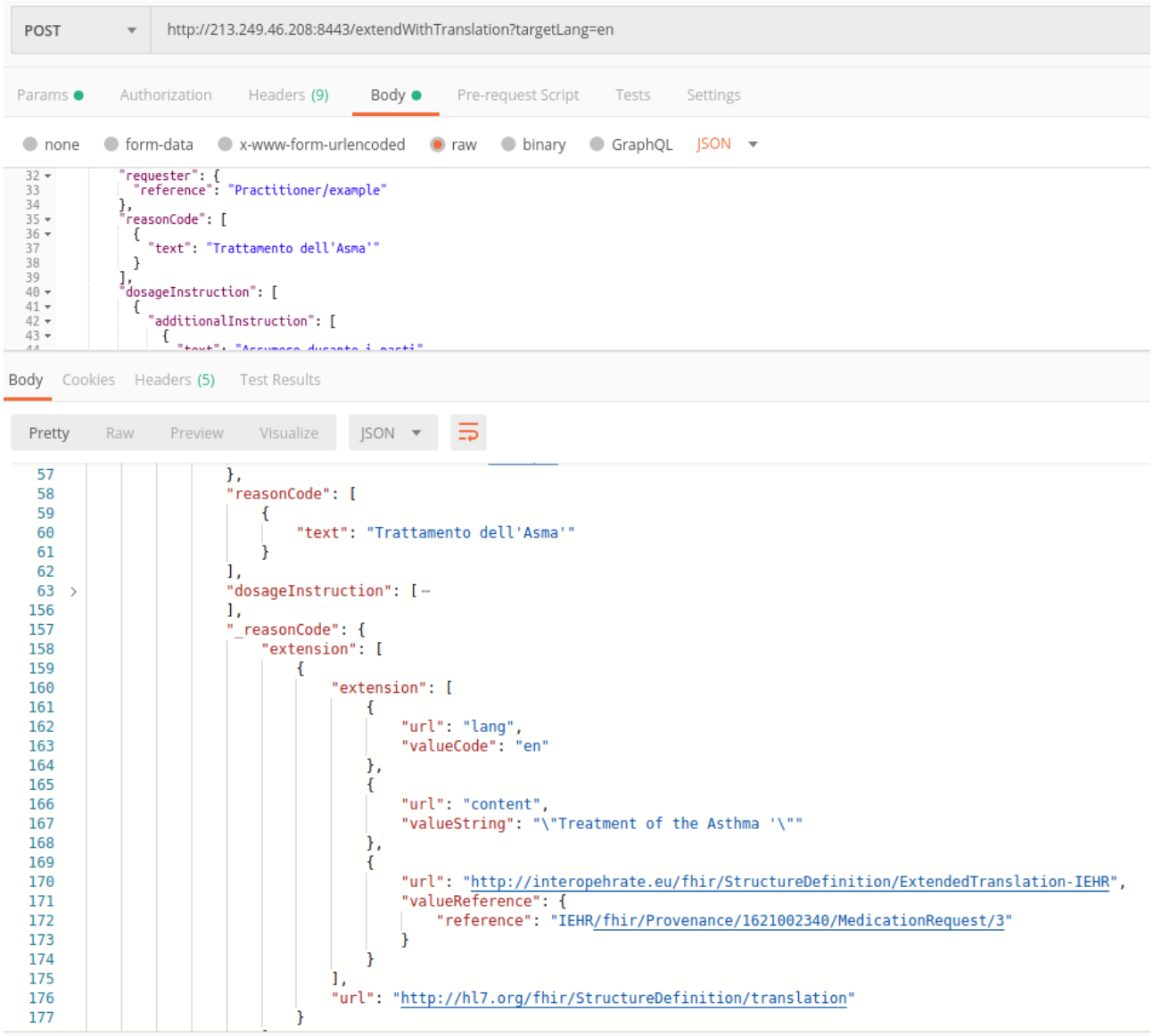

Figure 2 - Machine translation example within a raw FHIR resource

In Figure 2, another portion of the response received is reported, calling the same translation service endpoint as in Figure 1. The figure highlights how the textual Italian information "*Trattamento dell'Asma*", in the original medication information (Upper JSON file portion in figure), has been extended in the call's response with the extra attribute '*_reasonCode*' containing the language extension that includes the English version of the same information

Figures 3 show how the JSON resources translated, using the translation service, are visualized by the HCP Application. The original Italian information about allergies is reported both in its original form and in the translated version (English language).

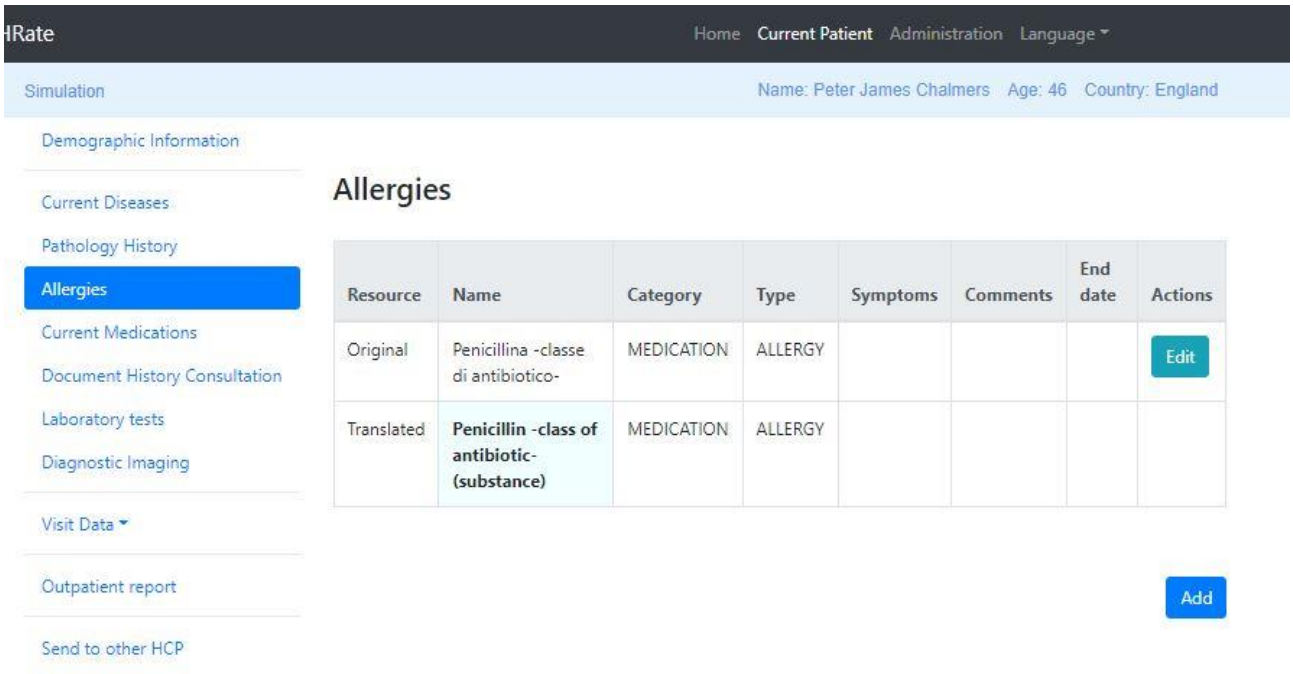

Figure 3 - Screenshot of the HCP App showing data in its original language and the relative translation version

3.4. Other related documentation (if any)

For further details on the fundamental conceptual design and the technological underpinnings of the translation and information extraction methods demonstrated here, please refer to the specification deliverable [D5.11].

REFERENCES

[1] **[D5.8]** InteropEHRate Consortium, Design of the Data Integration Platform v2 (deliverable D5.8). https://www.interopehrate.eu/resources/

[2] **[D5.11]** InteropEHRate Consortium, Design of information extractor and natural language translator v1 (deliverable D5.11). https://www.interopehrate.eu/resources/

[3] **[D5.21]** InteropEHRate Consortium, Library for Information Extraction and Translation v2 (deliverable D5.21). https://www.interopehrate.eu/resources/