applied sciences

MDPI

Article

Augmented EHR: Enrichment of EHR with Contents from Semantic Web Sources

Alejandro Mañas-García [1], José Alberto Maldonado [1,2], Mar Marcos [3,*], Diego Boscá [2] and Montserrat Robles [1]

[1] ITACA Institute, Universitat Politècnica de València, 46022 Valencia, Spain; almaaga1@upv.es (A.M.-G.); jamaldo@veratech.es (J.A.M.); mrobles@upv.es (M.R.)
[2] Veratech for Health SL, 46022 Valencia, Spain; diebosto@veratech.es
[3] Department of Computer Engineering and Science, Universitat Jaume I, 12071 Castellón, Spain
* Correspondence: mar.marcos@uji.es

Abstract: This work presents methods to combine data from the Semantic Web into existing EHRs, leading to an augmented EHR. An existing EHR extract is augmented by combining it with additional information from external sources, typically linked data sources. The starting point is a standardized EHR extract described by an archetype. The method consists of combining specific data from the original EHR with contents from the external information source by building a semantic representation, which is used to query the external source. The results are converted into a standardized EHR extract according to an archetype. This work sets the foundations to transform Semantic Web contents into normalized EHR extracts. Finally, to exemplify the approach, the work includes a practical use case in which the summarized EHR is augmented with drug–drug interactions and disease-related treatment information.

Keywords: augmented EHR; EHR archetypes; personalized medicine; computed-aided systems; EHR enrichment; Semantic Web; linked data; data exchange

Citation: Mañas-García, A.; Maldonado, J.A.; Marcos, M.; Boscá, D.; Robles, M. Augmented EHR: Enrichment of EHR with Contents from Semantic Web Sources. *Appl. Sci.* **2021**, *11*, 3978. https://doi.org/10.3390/app11093978

Academic Editor: Syoji Kobashi

Received: 28 March 2021
Accepted: 24 April 2021
Published: 27 April 2021

1. Introduction

The Electronic Health Record (EHR) constitutes a key piece of a patient's healthcare, [particularly because the paradigm of healthcare services has been changing towards a patient-centered scenario [1]. The completeness and accuracy of the EHR has a direct impact on the patient's healthcare results [2]. Moreover, additional potentially relevant information may be useful to complement a patient's EHR data for several scenarios [3,4], including secondary uses of the EHR or clinical decision support. Part of this useful content may be obtained from public data sources such as the Linked Open Data (LOD) cloud [5,6]; for example, the set of contraindicated drugs for a patient considering active medication and disorders. An integration of the EHR with other information sources, such as Semantic Web sources, could complement the EHR with such information.

This work introduces the augmented EHR (or EHR enrichment), understood as an existing EHR complemented with additional contents from external data sources. These contents may be any kind of information and/or knowledge that may be of interest for inclusion in the EHR. There are several approaches to achieve the augmented EHR, ranging from transforming information from external sources into a normalized EHR data, to transforming EHR data to the format of such sources. In both cases, ensuring a seamless integration requires some sort of data transformation to obtain a representation of both contents expressed in the same format. Data transformation in turn requires defining the relationships existing between both sources. Taking into account the kind of additional information to be included in the EHR, the EHR enrichment can be considered at the instance level or the schema level. In the first case, the enrichment of the EHR is individually performed for each patient (for instance, including additional information related to a specific patient's medication), whereas in the second case, the enrichment is performed

only once and is valid for all patients (for instance, including additional explanations about the meaning of the "active medication" section in the EHR). Nonetheless, the EHR and the external sources are continuously evolving and changing, so in both cases, the EHR enrichment should be performed when necessary and several versions of augmented EHRs could be required to ensure it remains valid over time.

This work proposes a preliminary approach to address the augmented EHR at instance level, enriching EHR extracts with contents from the Semantic Web, in addition to presenting a prototype to demonstrate its feasibility. Note that the concept of augmented EHR, as introduced here, comprises the inclusion of information from heterogeneous sources in the EHR, regardless of the initial representation of such information in their original sources. Although this work introduces a first approximation to build the augmented EHR with contents from the Semantic Web, the presented methodology can be adapted to obtain augmented EHRs using contents from other heterogeneous sources and/or even other EHRs. Specifically, the methodology is based on the identification and formal definition of bindings between an EHR and a Semantic Web source, starting from an EHR archetype. The binding, which must be designed manually, is used to identify and collect additional information (henceforth, the augmentation content) potentially relevant for the patient from Semantic Web sources.

Finally, the augmentation content is normalized as an EHR extract and embedded within the original EHR, resulting in an augmented EHR extract that represents a seamless integration of both sources. Note that in this work "EHR extract" and "EHR archetype instance" are interchangeably used to refer to normalized patient's health data. Additionally, we introduce the augmented summary EHR, a use case where the original summary EHR is augmented with augmentation contents from two different Semantic Web sources. Specifically, the *drug-drug-interaction.v1* augmentation content, which enriches the summary EHR by including drug–drug interactions from the DrugBank [7] database based on the active medication of the patient, and the *disease-related-treatments.v1* augmentation content, which retrieves information of treatments and effects for specific patient diseases from the National Drug File-Reference Terminology (NDFRT) ontology [8].

1.1. Background

Our work focuses on dual-model EHR architectures, which consist of separately defining the EHR schema (i.e., EHR archetypes) from the EHR data (i.e., EHR instances or EHR extracts). Specifically, a dual-model architecture defines two different models: first, a generic Reference Model (RM) designed to represent the most basic properties and structures of any EHR, examples are OpenEHR and ISO13606; and second, an Archetype Model (AM). Archetypes are detailed definitions of clinical concepts in the form of structured and constrained combinations of the entities of the reference model [9]. Archetypes are used to model concepts in the EHR, and EHR extracts are instances of archetypes containing the patient's EHR data.

This work introduces a methodology to enrich a dual-model-based EHR with potentially relevant information from the Semantic Web. The Semantic Web incorporates semantic metadata from ontologies and terminologies into the Web. It defines URIs that link to data, and documents, which allow categorization of each datapoint in the Semantic Web with coded values from terminologies and ontologies. In addition, The Semantic Web establishes mechanisms to define semantic relationship between data. Metadata languages, such as "Resource Description Framework" (RDF) or "Ontology Web Language" (OWL), allow indexing of web information resources with knowledge representations and store them in documents. The Semantic Web provides machines with the capability to interpret and make inferences over the data [10]. It introduces mechanisms to retrieve data, such as "SPARQL Protocol and RDF Query Language" (SPARQL), and to perform data reasoning, such as "Semantic Web Rule Language" (SWRL).

1.2. State of the Art

The benefits of combining the EHR with contents from heterogeneous information sources are well known and have been widely described. Most works in this field have focused on transforming EHR data to a semantic representation to perform reasoning for secondary uses of the EHR. Lezcano et al. [11] presented a solution for reasoning clinical archetypes using OWL ontologies and SWRL expressions. They argued that currently available archetype languages do not provide direct support for mapping to formal ontologies and then exploiting reasoning on clinical knowledge. They translated definitions expressed in the OpenEHR Archetype Definition Language (ADL) to formal OWL representation, and then applied SWRL rules to instances of clinical data. Similar approaches have been employed to improve secondary uses of the EHR. Tao et al. [12] introduced an OWL representation of the clinical element model (CEM), to perform reasoning and consistency checking on derived models for secondary uses of the EHR. J. T. Fernández-Breis et al. [13] applied Semantic Web technologies to leverage the EHR standards for the identification of patient cohorts. Both works established approaches to convert EHR data into semantic representation for reasoning. However, EHR data expressed as semantic representation has a limited degree of interoperability in normalized health IT systems in compliance with the dual-model EHR architecture.

The conversion of EHR data into semantic representation has been widely explored to improve the interoperability of EHR data. D. J. Odgers et al. [14] designed a method for EHR data mining using linked data. They transformed a de-identified version of the Sandford's STRIDE database [15] into a semantic clinical data warehouse to perform cohort selection, phenotyping profile, and identification of disease genes. V. Kilintzis et al. [16] presented a telehealth data management framework to support integrated care services for chronic patients. They built it upon HL7 and ontologies to obtain a semantically enriched representation of the EHR data following linked data principles. Nevertheless, none of recent works addressed the transformations in the opposite direction, from Semantic Web representation to a formal representation of the EHR.

J. J. Cimino et al. [17] introduced Infobuttons, a method to access external knowledge resources from the EHR. It consists of a set of specifications defining URLs parametrized with data from the EHR extract. When the user clicks one of these links, additional information is retrieved from an external source and shown to the user as HTML. Although the Infobuttons approach has been incorporated as part of the HL7 standard, its main limitation is the lack of interoperability of this solution with EHR standards based on the dual model representation. Similarly, M. Alfano et al. [18] presented a method to obtain additional definitions of clinical terms. It starts from EHR extracts and queries a thesaurus to provide explanations of medical terms through a web service. A. Chetta et al. [19] introduced a non-standard solution to enrich EHRs with additional contents from the nursing service. It helps nurses to document and reason about patient data and about the clinician's understanding of patient data. The solutions of both M. Alfano et al. and A. Chetta et al. highlight the relevance of enriching EHR data with potentially relevant information from external sources. However, there is a lack of standardization of the additional information combined with EHR extracts, leading to a limited degree of interoperability of the additional information in normalized health IT systems. The augmented EHR presented in this paper aims to mitigate this limitation by enriching archetype-based EHR extracts with additional information from Semantic Web sources expressed in terms of EHR archetypes.

2. Approach

The aim is to enrich EHR extracts with data drawn from heterogeneous semantically rich information sources. To achieve a widely applicable solution, the starting points are: (i) EHR instances normalized according to a set of **EHR archetypes** (i.e., ISO13606 or OpenEHR archetypes), and (ii) augmentation content expressed in Semantic Web format (usually structured and stored as RDF datasets). Next, another set of EHR archetypes (hereinafter **augmentation archetypes**), modelled specifically to represent the augmen-

tation content as normalized EHR extracts, allow homogenization of the format of the augmentation content and the initial EHR extract. A last EHR archetype called **augmented archetype**, which comprises the initial EHR archetype and augmentation archetypes, allows a seamless integration of the initial EHR extract and the augmentation content into the augmented EHR. Note that the augmented archetype may contain one or more augmentation archetypes. For coherency reasons, the structure of the augmentation archetype will be determined by the one of the initial EHR archetype.

The process to define an augmented EHR relies on EHR standards, EHR archetypes, and mapping scenarios. Formally, this process can be defined in terms of two data exchange problems (data exchange 1 and 2 in Figure 1): to normalize augmentation contents into EHR extracts, and to combine initial EHR extracts with normalized augmentation contents into augmented EHR extracts. In our approach, a data exchange setting is defined by a 3-tuple <[SS], TS, M>. Here a tuple is used to formally define a data exchange operation, and should not be confused with the possible RDF tuples to which such operation can be applied. In each tuple, [SS] stands for one or more source schemas, TS for a target schema, and M for a set of mapping specifications between [SS] and TS. Furthermore, data exchanges at data level are represented by the transformation T, which applies the mapping definitions in M to transform instances of [SS] into instances of TS.

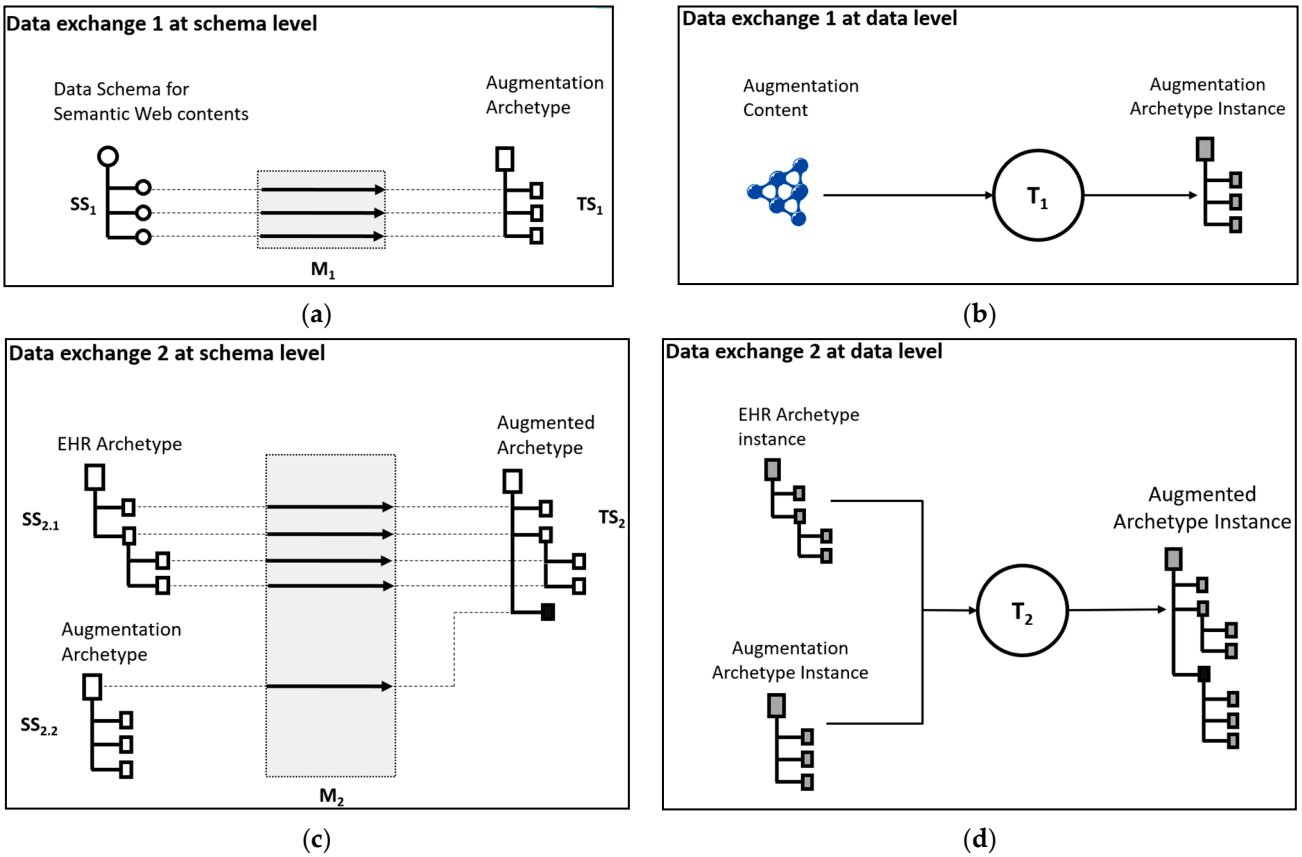


Figure 1. Data exchange operations required to build augmented EHR extracts. (**a**) At schema level and (**b**) at data level depict the operations to normalize the augmentation content as an EHR extract, whereas (**c**) at schema level and (**d**) at data level show the operations to transform the initial EHR extract plus the normalized augmentation content into an augmented EHR extract.

The first data exchange problem is to normalize the augmentation content. This is defined at the schema level by the tuple <[SS_1], TS_1, M_1>, where SS_1, the data schema for Semantic Web contents, is a canonical representation of the augmentation content expressed in Semantic Web format. TS_1, the augmentation archetype, is a canonical form

of the augmentation content expressed in EHR archetypes. Finally, M_1 is the specification of relationships between elements in SS_1 and elements in TS_1. At the data level, the transformation T_1 is an implementation of M_1 intended to normalize the augmentation content into an EHR extract by converting instances of $[SS_1]$ into instances of TS_1.

The second data exchange problem is to combine initial EHR extracts with normalized augmentation contents into augmented EHR extracts. The tuple $<[SS_{2\cdot1}, SS_{2\cdot2}], TS_2, M_2>$ defines this data exchange at the schema level, where $SS_{2\cdot1}$ is the initial EHR archetype and $SS_{2\cdot2}$ is the augmentation archetype. The two source schemas are canonic representations of the initial EHR extract and the normalized augmentation content, respectively. The target schema TS_2, the augmented archetype, is a canonical form of the augmented EHR extract. Finally, M_2 represents the mapping specifications between elements in $[SS_{2\cdot1}, SS_{2\cdot2}]$ and elements in TS_2. At the data level, the transformation T_2 is an implementation of M_2 intended to generate augmented EHR extracts by converting instances of $[SS_{2\cdot1}, SS_{2\cdot2}]$ into instances of TS_2. Note that TS_1 and $SS_{2\cdot2}$ are the same schemas and are designed to handle the same type of instances (i.e., augmentation content normalized as an EHR extract).

The use case presented in this work (the augmented summary EHR) employs the W3C schema for query results [20] as the canonical form of Semantic Web contents (SS_1), ISO13606 archetypes as the canonical representation of EHR extracts ($SS_{2\cdot1}$, $SS_{2\cdot2}$, TS_1, and TS_2), and the high level declarative mapping language of the LinkEHR platform [21] to define mapping specifications (M_1 and M_2) at the schema level, and to generate data transformations (T_1 and T_2) at the data level.

Each EHR archetype may contain several items susceptible to be complemented with additional information from external sources (for example, coded values identifying a diagnosis). These items are the entities of interest (EOIs onwards). EOIs are represented by a path from the root of the EHR archetype to the EOI, and can be used to establish the parts of an EHR archetype that are potentially relevant to be augmented. Moreover, EOIs allow identification of the contents from the external sources that may be considered to be augmentation content to enrich the EHR. The EOIs are the binding between the EHR and the augmentation content, and are the starting point to explore external sources in search of related content. Some of these contents may be relevant in the context of the initial EHR archetype and are employed as the augmentation content.

The process to design and create an augmented archetype for a given EHR archetype, using the Semantic Web as the external source, requires the following tasks to be performed and definitions at the schema level (Figure 2):

1. Identify the EOIs in the initial EHR archetype. These entities are typically coded values and must be manually selected. Alternatively, free text values can be considered as EOIs whenever these values are translated into coded values (e.g., applying natural language processing methods or querying terminology repositories).
2. Identify a potentially interesting augmentation content from an external source. This requires selecting an information source related to the EOIs' coded values. Such a source must include data items representing the concepts of the EOIs (for instance, a set of drugs or a set of diseases), and must contain potentially relevant information to enrich the initial EHR archetype. The augmentation content will be collected from this source.
3. Ensure the feasibility of binding the EHR with the augmentation content by finding mappings between EOIs and equivalent data items in the external source. Typically, different terminologies use different codes to represent the same concept. This concept code mismatch among terminologies can be overcome by employing a terminology mapping service. In addition, VoID [22] descriptors may be useful for finding such mappings because they allow identification of the public datasets from the Semantic Web (such as those available in the LOD cloud) that contain coded values of the terminologies involved.
4. Build the binding between the EHR and the augmentation content. The binding must link EOIs from the initial EHR with equivalent data items in the external source,

preserving the meaning of the EOIs in the context of the augmentation content. As this work is focused on collecting the augmentation content from Semantic Web sources, the binding (henceforth, the EOI binding triples) is expressed in RDF. The building of the EOI binding triples can be automatized with a procedure that, starting from the coded values of the EOI and the name of the involved terminologies, uses a terminology mapping service and information collected from VoID descriptors to generate the EOI binding triples.

5. Use the binding defined in step 4 to gather the augmentation content from the external source. A SPARQL query combines triples from the external source dataset and the EOI binding triples to collect the augmentation content. The triples from the external source resulting from the query determine the augmentation content to be added to the augmented EHR, and they must be manually selected according to their relevance in the context of the EOIs and the initial EHR archetype.
6. Modeling and mapping the augmentation archetype. The purpose of this archetype is to normalize the augmentation content as an EHR extract. The augmentation archetype should contain at least an element for each data item in the augmentation content. Moreover, the mapping specifications between the data schema of the augmentation content and the augmentation archetype must be defined at this point. Note that this step could be performed once the augmentation content has been identified, in parallel with steps 3, 4, and 5. In addition, the augmentation archetype can be automatically modeled by building an archetype consisting of a cluster that holds one element for each data item in the augmentation content. Finally, a similar procedure can be applied to automatize the mapping specifications.
7. Modeling and mapping the augmented archetype. This archetype allows the initial EHR archetype and the augmentation archetype to be combined. As in the previous step, the mapping specifications between the source schema (i.e., the initial EHR archetype and the augmentation archetype) and the target schema (i.e., the augmented archetype) must be defined. Naturally, this step should be performed once step 6 has been completed. The augmented archetype results from cloning the initial EHR archetype and then adding slot references to the augmentation archetypes at specific locations. Although this paper also introduces an algorithm for suggesting possible locations that are suitable for such slot references, the final locations must be manually selected to preserve the meaning of the initial EHR archetype. Once a location has been chosen for each slot reference, the augmented archetype can be modeled automatically. In addition, the mapping definition can be automatized after determining the locations of the slot references.

The following considerations should be noted. First, steps from 2 to 6 define the augmentation archetype, whereas steps from 1 to 7 describe the whole process to obtain the augmented archetype. Second, an augmented archetype may include different augmentation archetypes, one per each EOI identified at schema level in the initial EHR archetype. In such case, the process would involve the design of several augmentation archetypes. Third, there are significant differences between the definition of an augmented archetype and the generation of instances of an augmented archetype. The definition is a design task. It is done at schema level and requires manual intervention at some points. In contrast, the execution occurs at data level and is performed in a fully automatic manner. Thus, once an augmented her is defined, it automatically enriches any EHR extract that fits the initial EHR archetype. Finally, the cardinality of EOIs in the initial EHR archetype determines the cardinality of the root item of the augmentation archetype (usually a cluster). A plural cardinality of an EOI may lead to several EOI binding triples (one for each coded value that the initial EHR extract may contain in such an EOI). Therefore, several sets of augmentation content may be found that must fit into the same instance of the augmentation archetype. Thus, the cardinality of the root item in the augmentation archetype must be greater than or equal to the cardinality of the EOI that constitutes the binding with this augmentation archetype.

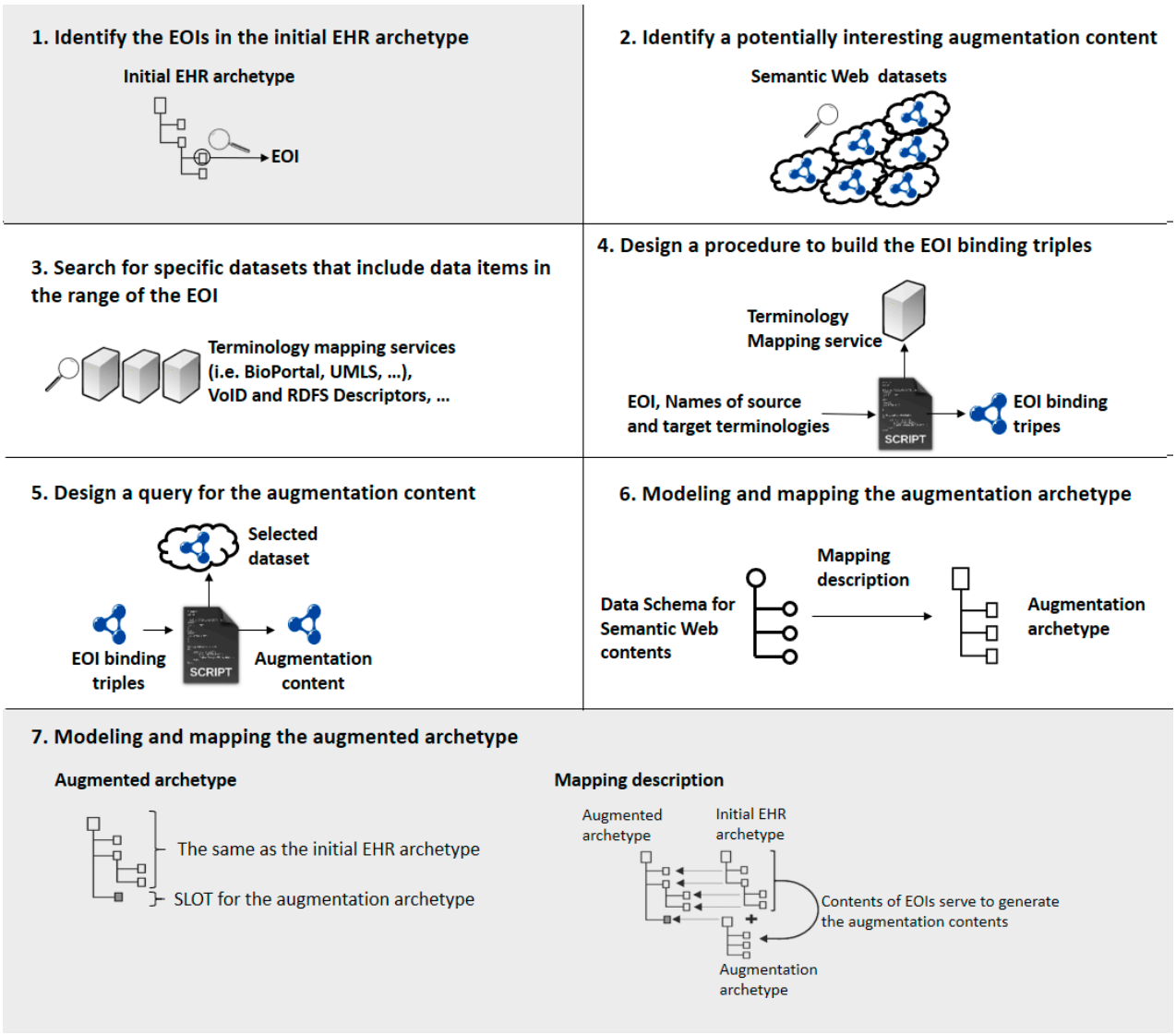


Figure 2. Methodology to define the augmentation archetype (steps 2 to 6) and the whole process to obtain the augmented archetype (steps 1 to 7).

3. Materials and Methods

This section describes the methodology and tools used to design an augmented archetype. For the sake of clarity, some of the steps in Figure 2 are merged in the explanation. A step-by-step example is also included for a better understanding of the methodology. The example shows how to build the augmented medication archetype starting from the medication archetype, and adding information about drug–drug interactions as augmentation content. As mentioned previously, the starting point is an EHR archetype.

(A) Identifying the augmentation content for a given EHR archetype. Designing an augmented archetype requires (i) identifying EOIs in the EHR archetype, (ii) identifying potentially interesting external information sources, and (iii) finding equivalent concepts between them (steps 1, 2, and 3 in Figure 2). Usually, EOIs are coded values (e.g., diagnostics, procedures, and medications), that can be bound to external information sources to enrich the initial EHR extract with additional information or knowledge. Semantic Web datasets are used for this work as external information sources. Therefore, potentially interesting Semantic Web datasets should be identified to complement the information provided by the EOIs. In addition, a terminology mapping service allows identification

of equivalencies between coded values from different terminologies. This can be used to obtain local codes from the target terminology (i.e., the terminology used in the Semantic Web dataset) equivalent to the EOI codes in the source terminology (i.e., the terminology used in the initial EHR archetype). Finally, VoID descriptors allow identification of public LOD datasets exploitable through SPARQL queries.

Example: As a starting point, the initial EHR archetype is the medication EHR archetype published by the Spanish NHS [23] (*CEN-EN13606-ENTRY.Medicacion.v1*). The selected EOIs are the data elements containing the active medication (located within the archetype at*/items[archetype_id="at0005"]/parts[archetype_id="at0010"]/value*) To build the augmented medication archetype, the EOIs are augmented with additional information about drug–drug interactions (i.e., the augmentation content). In this case, the EOIs are defined as 'CD' data type, which are coded values that may contain SNOMED-CT [24] codes. To identify data items equivalent to SNOMED-CT drug codes in different terminologies/ontologies, a random subset of SNOMED-CT drug codes is used as a sample to query the BioPortal [25,26] terminology mapping service. For each data item in the sample subset, the service returns a list of equivalent terms from other sources, combined with URLs of these sources (e.g., the source URL for an item of the DrugBank database: *http://www.drugbank.ca/drugs/DB00099*). Usually, source URLs point to the organization that created the terminology, which often does not provide a public LOD dataset to exploit the terminology through a query language. In this situation, description mechanisms available in the LOD cloud, such as VoID descriptors and/or RDFS properties, are useful for finding LOD datasets containing the required terminology. In this example, an SPARQL query using VoID descriptors serves to identify the DrugBank database as a LOD dataset in the LOD cloud (e.g., the source URL for an item of the DrugBank LOD dataset: *http://bio2rdf.org/drugbank:DB00099*).

(B) Transforming EOI into a semantic representation. A formal relationship is defined between EOIs and their equivalent data items in the target dataset. This relationship is expressed through the EOI binding triples, which are RDF triples composed of the EOI coded values as subjects and the equivalent data items as objects (step 4 in Figure 2). These triples can be built using upper-level ontology predicates, intended to describe the relationships between different ontologies and/or terminologies. The choice of predicate varies according to the purpose of the augmentation content and its relationship to the EOI. Typically, predicates such as *match* or *exactMatch* from the SKOS ontology are appropriate to define relationships between coded values from different terminologies. We encourage the use of the exactMatch predicate for unambiguous matches. Additionally, other predicates from the SKOS mapping properties can be used to detail the mapping relationship between the EOI coded value and the data item in the target dataset: mappingRelation, closeMatch, exactMatch, broadMatch, narrowMatch, and relatedMatch.

At runtime, an automatic procedure can generate the EOI binding triples starting from the elements obtained in (A): the initial EHR extract, the names of the involved terminologies, and the URL of an existing LOD dataset holding the augmentation content. Specifically, the procedure takes the following inputs:

1. To identify the EOI coded values (subjects in the EOI binding triples):
   - 1.1 Initial EHR extract
   - 1.2 Path to the EOI in the initial EHR archetype
2. To establish the predicate for the EOI binding triples:
   - 2.1 The predicate URL (for instance, *skos:exactMatch*)
3. To collect the equivalent data items in the target dataset (objects in the EOI binding triples):
   - 3.1 The name of the source terminology/ontology employed by the EOI coded values
   - 3.2 The URL of the target terminology/ontology returned by the terminology mapping service. Both inputs (3.1 and 3.2) are used to query the terminology mapping service and retrieve data items from the target terminology equivalent to the EOI coded values.

4. To format the equivalent data items (3) as URLs from an existing LOD dataset:
   4.1 The URL prefix of data items in the target dataset. This allows the use of data items obtained from the terminology/ontology mapping service as objects in the EOI binding triples. Otherwise, the data returned by the mapping service might not be exploitable through a query language.

By using these inputs, it is possible to implement a procedure that extracts EOI coded values, queries the terminology mapping service for equivalent data items, and builds the EOI binding triples using the provided predicate. This algorithm returns the EOI binding triples as a result. Note that, for a given augmentation content, the only argument varying among different executions is the initial EHR extract (1.1). The rest of the arguments must be identified once at the time of design and remain valid for all of the executions. Similarly, the described procedure to build the EOI binding triples only needs to be implemented once and is valid for all executions. The example shows how this procedure works.

Example: Continuing with the example introduced in (A), the next step is to develop the algorithm described in (B) to build the EOI binding triples. In the example, the algorithm takes the inputs:

1.1. An EHR extract of the medication EHR archetype that has a coded value for the drug Filgrastim (SNOMED-CT code *386948008*) in the EOI.

1.2. The path to the EOI (i.e., drug coded value) in the medication EHR archetype: */items[archetype_id="at0005"]/parts[archetype_id="at0010"]/value*.

2.1. The predicate skos:exactMatch, whose full URL is https://www.w3.org/2009/08/skos-reference/skos.html#exactMatch [27].

3.1. The name of the source terminology to be used in the terminology/ontology mapping service: *SNOMEDCT*.

3.2. The name of the target dataset to build the EOI binding triples: *DrugBank*.

4.1. The URL prefix *http://bio2rdf.org/drugbank:* [28] pointing to a LOD representation of the DrugBank dataset.

Taking these input arguments, the EOI binding triples are generated by performing the steps:

I. Extracting EOI coded value (*386948008*) from the argument 1.1 using the path from 1.2.
II. Querying the terminology/ontology mapping service using the EOI coded value (*386948008*) and the argument from 3.1 (*SNOMEDCT*): http://data.bioontology.org/ontologies/SNOMEDCT/classes/http%3A%2F%2Fpurl.bioontology.org%2Fontology%2FSNOMEDCT%2F386948008/mappings?apikey=\T1\textless{}apikey> [26].
III. Obtaining the equivalent data item in the target dataset from the response of the terminology/ontology mapping service, filtering by the identifier of the target dataset (from argument 3.2): @id: http://www.drugbank.ca/drugs/DB00099 [29] The URL *http://www.drugbank.ca* does not provide a LOD representation of the DrugBank dataset. A further step is required (step IV) to convert this data item into a URL belonging to an existing LOD dataset that can be exploited through SPARQL queries.
IV. Using the URL prefix provided in the argument 4.1, plus the coded value of the data item in the target terminology (*DB00099*). This results in a URL belonging to an existing LOD dataset that can be used in a query: *http://bio2rdf.org/drugbank:DB00099*
V. Building the EOI binding triples using the predicate from argument 2.1 (*skos:exactMatch*):
<http://purl.bioontology.org/ontology/SNOMEDCT/386948008>
<https://www.w3.org/2009/08/skos-reference/skos.html#exactMatch>
<http://bio2rdf.org/drugbank:DB00099>

The *skos:exactMatch* predicate is used to generate EOI binding triples of the form <*SnomedDrug*> *skos:exactMatch* <*DrugBankDrug*>. Values for subjects and objects in these triples are obtained at runtime. Specifically, the <*SnomedDrug*> is defined by the value of the EOI in the EHR archetype instance, whereas the <*DrugBankDrug*> is a term from the DrugBank LOD dataset equivalent to <*SnomedDrug*>.

(C) Querying Semantic Web datasets. The next step is to build a SPARQL query to retrieve the augmentation content. The query graph is formed by the target dataset plus the EOI binding triples (Figure 3). In this way, the query can be parametrized with the EOI binding triples, allowing the target dataset to be searched for the augmentation content using the EOI coded values as query criteria. At runtime, this approach leads to identification and retrieval of the augmentation content for a given EHR extract (step 5 of Figure 2), by requesting only the data items from the query graph that include the predicates used in the EOI binding triples. The results of the SPARQL query are formatted according to the W3C schema for query results, which is a canonical representation of contents from the Semantic Web.

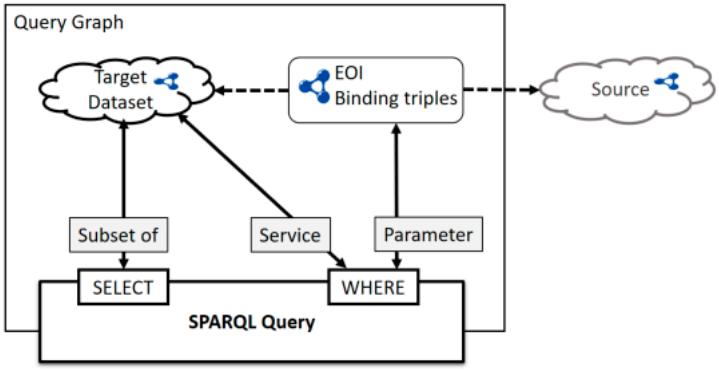


Figure 3. The SPARQL query and the query graph to obtain the augmentation content. The query graph is formed by the EOI binding triples plus the target dataset. The EOI binding triples are injected at runtime in the query graph as parameters, whereas the target dataset is included through a service (i.e., using the *SERVICE* clause of federated SPARQL queries). Although the source remains outside the query graph, it is possible to identify the augmentation content in the *WHERE* clause, using the EOI binding triples as a filter. Finally, the *SELECT* clause establishes which subset of the target dataset constitutes the augmented content.

Example: The example illustrates a SPARQL query to obtain the augmentation content, which consists of information about drug–drug interactions for the active medications in the initial EHR extract (an instance of the medication archetype). The augmentation content includes interaction descriptions (*?interactionDescription*), names (*?targetDrugName*), and codes (*?targetDrugCode*) of the interacting drugs. Figure 4 shows both the query and the query graph for this example. The query graph consists of the DrugBank dataset (i.e., the target dataset) plus the EOI binding triple resulting from the example in section B. At runtime, the EOI binding triple sets the value of*?sourceDrugCode* as *DB00099* (*Filgrastim*). Then, the triples in the *SERVICE* clause query the DrugBank dataset to obtain descriptions of drug–drug interactions related to *Filgrastim*, in addition to the names and codes of the interacting drugs.

The execution of this SPARQL query provides the drug–drug interactions for *Filgrastim*. Figure 5 shows the results formatted according to the W3C schema for query results. Note that the variables in the *SELECT* clause of the SPARQL query (Figure 4) appear in the augmentation content (Figure 5).

(D) Transforming the augmentation content into a normalized EHR extract. This section focuses on addressing the sixth step of Figure 2, which encompasses two tasks: (i) modeling the augmentation archetype, which must contain at least an element for each data item in the augmentation content (i.e., for each variable in the *SELECT* clause of the SPARQL query in section C), and (ii) defining mapping specifications between the W3C schema for query results and the augmentation archetype (i.e., data exchange 1 at schema level in Figure 1). The modeling of the augmented archetype can be automated by generating an archetype whose root is a cluster, and adding an element for each variable included in the augmented content. In addition, the mapping specification could also

be automated by following a similar procedure. However, to ensure that the involved concepts preserve their original meaning, it is advisable to perform these steps manually.

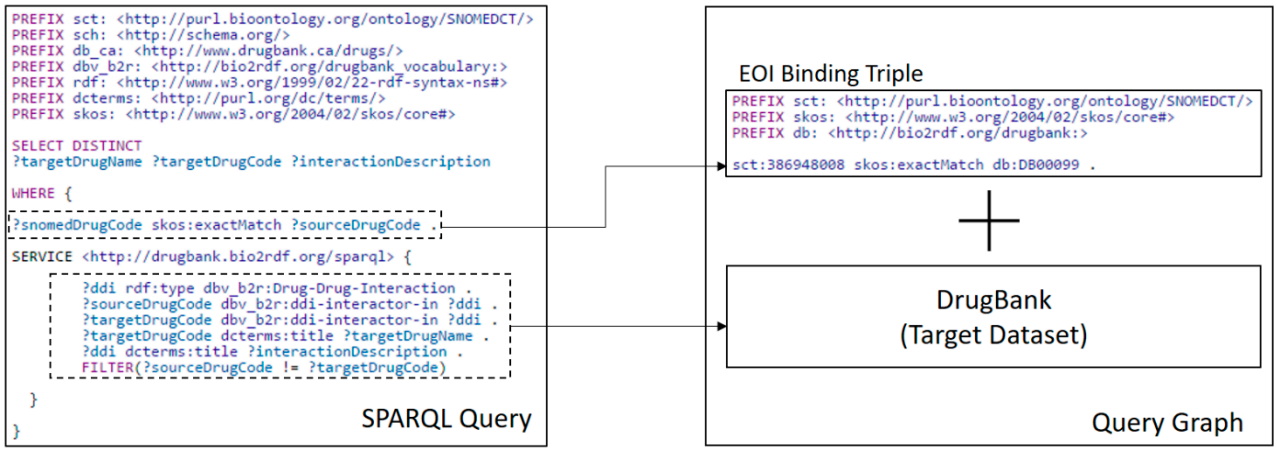


Figure 4. The SPARQL query and the query graph of the example in section C. The first part of the query in the *WHERE* clause (*?snomedDrugCode skos:exactMatch ?sourceDrugCode.*) references the EOI binding triples. The second part of the query, the *SERVICE* clause, specifies how the EOI binding triples filter the target dataset. Finally, the *SELECT* clause specifies the data items from the target dataset that constitute the augmentation content (*?targetDrugName ?targetDrugCode ?interactionDescription*).

The LinkEHR platform provides the specification of mappings between the W3C schema for query results and the augmentation archetype. LinkEHR provides a set of tools to manually model EHR archetypes. Moreover, the mapping capabilities of LinkEHR allow the definition of high-level declarative mappings relating a set of data schemas. The mapping language basically specifies how to compute a value for an atomic attribute of the target schema by using a set of atomic elements from the source schema. These value correspondences are defined by a set of pairs, consisting of a transformation function and a filter. The simplest kind of transformation function is the identity function, which copies a source value into a target atomic attribute. In cases in which it is necessary to transform source atomic values, a wide range of transformation functions is supported. Once mapping specifications are defined, LinkEHR compiles them into a XQuery script able to transform instances of the source schema to instances of the target schema.

LinkEHR can be used to model the augmentation archetype, and to define mapping specifications between the W3C schema for query results and the augmentation archetype. Compiling these mapping specifications results in a XQuery script, whose execution transforms the augmentation content into a normalized EHR extract according to the augmentation archetype (i.e., data exchange 1 at data level in Figure 1).

Example: The augmentation archetype for drug–drug interactions is modeled as a cluster containing three elements: the drug–drug interaction description, and the name and the code of the interacting drug (one element for each data item in the augmentation content *?interactionDescription*, *?targetDrugName* and *?targetDrugCode* respectively). The cluster can be repeated as many times as drug–drug interactions are found for the active medication in the initial EHR extract. Next, mapping specifications are defined by means of identity mappings between elements in the W3C schema for query results and elements in the augmentation archetype. Both the modeling of the augmentation archetype and the definition of mapping specifications are performed using the LinkEHR platform.

```
<sparql xmlns="http://www.w3.org/2005/sparql-results#" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3.org/2001/sw/DataAccess/rf1/result2.xsd">
 <head>
  <variable name="targetDrugName"/>
  <variable name="targetDrugCode"/>
  <variable name="interactionDescription"/>
 </head>
 <results distinct="false" ordered="true">
  <result>
   <binding name="targetDrugName"><literal xml:lang="en">Bleomycin</literal></binding>
   <binding name="targetDrugCode"><uri>http://bio2rdf.org/drugbank:DB00290</uri></binding>
   <binding name="interactionDescription"><literal xml:lang="en">DDI between Filgrastim and Bleomycin
                        - Filgrastim may enhance the adverse/toxic effect of Bleomycin.
                        Specifically, the risk for pulmonary toxicity may be increased.</literal></binding>
  </result>
  <result>
   <binding name="targetDrugName"><literal xml:lang="en">Cyclophosphamide</literal></binding>
   <binding name="targetDrugCode"><uri>http://bio2rdf.org/drugbank:DB00531</uri></binding>
   <binding name="interactionDescription"><literal xml:lang="en">DDI between Filgrastim and Cyclophosphamide
                        - Filgrastim may enhance the adverse/toxic effect of Cyclophosphamide.
                        Specifically, the risk of pulmonary toxicity may be enhanced.</literal></binding>
  </result>
  <result>
   <binding name="targetDrugName"><literal xml:lang="en">Topotecan</literal></binding>
   <binding name="targetDrugCode"><uri>http://bio2rdf.org/drugbank:DB01030</uri></binding>
   <binding name="interactionDescription"><literal xml:lang="en">DDI between Filgrastim and Topotecan
                        - Filgrastim may enhance the adverse/toxic effect of Topotecan.</literal></binding>
  </result>
 </results>
</sparql>
```

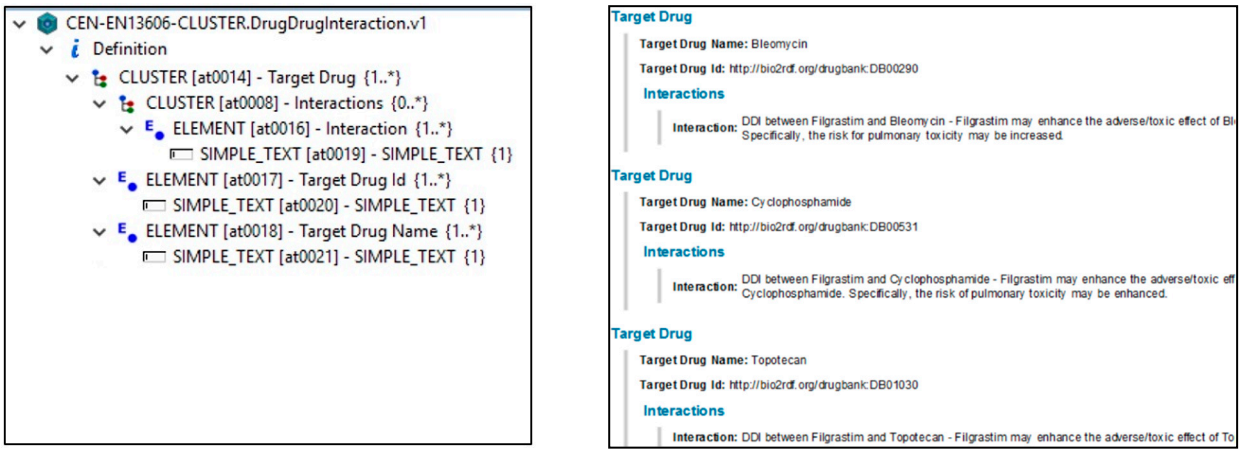
(**a**)

targetDrugName	targetDrugCode	interactionDescription
Bleomycin	http://bio2rdf.org/drugbank:DB00290	DDI between Filgrastim and Bleomycin - Filgrastim may enhance the adverse/toxic effect of Bleomycin. Specifically, the risk for pulmonary toxicity may be increased.
Cyclophosphamide	http://bio2rdf.org/drugbank:DB00531	DDI between Filgrastim and Cyclophosphamide - Filgrastim may enhance the adverse/toxic effect of Cyclophosphamide. Specifically, the risk of pulmonary toxicity may be enhanced.
Topotecan	http://bio2rdf.org/drugbank:DB01030	DDI between Filgrastim and Topotecan - Filgrastim may enhance the adverse/toxic effect of Topotecan.

(**b**)

Figure 5. SPARQL query results of the example in section C. These results are the augmentation content. (**a**) The results are formatted in XML according to the canonical W3C schema for query results. (**b**) The results are presented in a table for better understanding.

Figure 6 depicts the *CEN-EN13606-CLUSTER.DrugDrugInteraction.v1* archetype defined for this example (i.e., the augmentation archetype), and the augmentation content resulting from the example in section C formatted according to this archetype.



(**a**) (**b**)

Figure 6. (**a**) A LinkEHR visualization of the augmentation archetype *CEN-EN13606-CLUSTER.DrugDrugInteraction.v1* for drug–drug interactions of the example in section D. (**b**) an instance of the *CEN-EN13606-CLUSTER.DrugDrugInteraction.v1* augmentation archetype presented in HTML and holding drug–drug interactions for *Filgrastim* (i.e., the augmentation content resulting from the example in section C).

(E) Integrating the augmentation content and the initial EHR extract. The augmented archetype comprises the initial EHR archetype and the augmentation archetype (step 7 in Figure 2). An approach to building the augmented archetype is to specialize the initial EHR archetype by including a slot reference to embed the augmentation archetype. The path to the slot reference must be manually selected to ensure that the resulting archetype preserves the original meaning. However, once the slot location is chosen, the modeling of the augmented archetype can be automatic based on the initial EHR archetype, the path to the slot reference, and the augmentation archetype (whose root component establishes the data type of the slot reference).

To assist in the process of choosing the proper slot location for the augmentation archetype, this work introduces an algorithm (Figure 7) to identify which components of the initial EHR archetype are valid to contain the slot reference. The result of this algorithm is a set of candidate paths ordered by proximity to the EOIs in the initial EHR archetype:

```
Require
   -  Root_Type: Type of the root element in the augmentation archetype
   -  EHR_Archetype_Elements: The set of elements in the EHR archetype
   -  EOI_Paths: Paths to the EOIs in the EHR archetype
   -  Preferred_Slot_Path: Slot location provided by the user (Optional, it may
      be empty)

Returns
   -  Slot_Locations: The candidate path(s) to hold the augmentation archetype

Initialize
   -  Slot_Paths: Set of candidate paths for the slot location (initially empty)

for each e ϵ EHR_Archetype_Elements do
      If  e.type can hold an element of type Root_Type
            and  e.max_cardinality > e.cardinality  then
           Slot_Paths.add(e.path)
      end if
end for
If Slot_Paths contains Peferred_Slot_Path  then
      Slot_Locations ← Preferred_Slot_Path
else
      Slot_Locations ← sortByProximity(Slot_Paths, EOI_Paths)
end if
return Slot_Locations
```

Figure 7. Proposed algorithm to determine components in the initial EHR archetype candidate to include the slot reference for the augmentation content.

The returned value *Slot_Locations* includes a set of candidate paths from the initial EHR archetype that are valid to hold the augmentation archetype. The path for the slot reference to the augmentation archetype must be selected manually from the *Slot_Locations* set of paths. Next, the augmented archetype can be modeled as a specialization of the initial EHR archetype, containing the same elements than the initial EHR archetype plus an additional slot for the augmentation archetype.

The LinkEHR platform provides tools to model the augmented archetype, and to define mapping specifications between the initial EHR archetype, the augmentation archetype, and the augmented archetype (i.e., data exchange 2 at schema level in Figure 1). In this case, the mapping specifications are based on identity functions and slot inclusions. Finally, the LinkEHR platform serves to compile the mapping specifications into a XQuery script, whose execution transforms instances of the initial EHR archetype and the normalized augmentation content into an augmented EHR extract, expressed as an instance of the augmented archetype (i.e., data exchange 2 at data level in Figure 1).

Note that an augmented archetype could be modeled to hold several augmentation archetypes by including additional slots for them in the modeling stage. In such cases, it would be required to repeat the steps in section E for each augmentation archetype to be included in the augmented archetype.

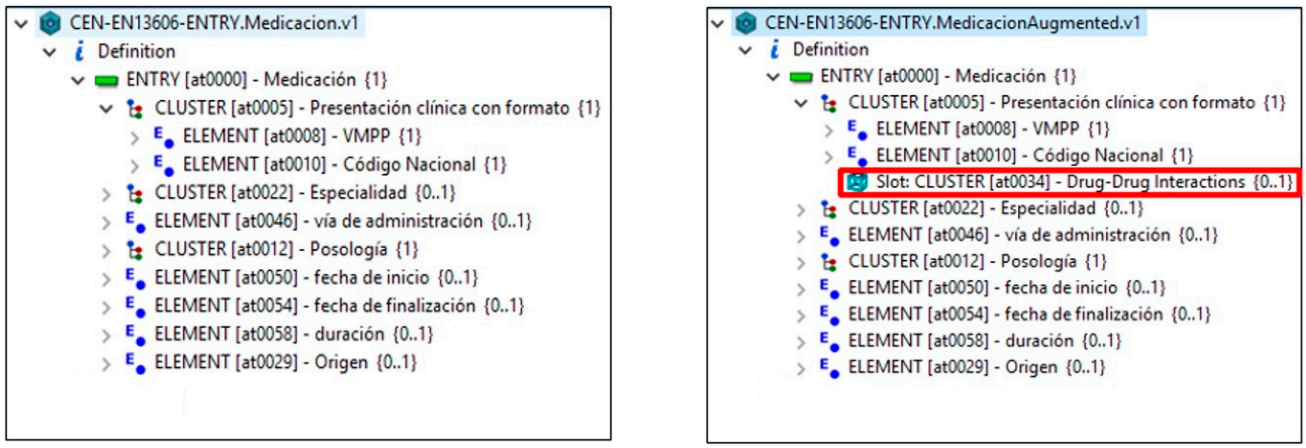
Example: The last step of this augmented EHR example is to generate the augmented archetype *CEN-EN13606-ENTRY.MedicacionAugmented.v1*, as a specialization of the initial EHR archetype *CEN-EN13606-ENTRY.Medicacion.v1*. In this example, the augmented archetype is intended to contain both the *CEN-EN13606-CLUSTER.DrugDrugInteraction.v1* archetype (i.e., the augmentation archetype), plus the initial EHR archetype.

To identify the path for the slot reference to the augmentation archetype, the algorithm introduced in section E is executed using the inputs:

- *Root_Type: CLUSTER*
- *EHR_Archetype_Elements: Elements in CEN-EN13606-ENTRY.Medicacion.v1*
- *EOI_Paths: Path to the medication coded value, /items[archetype_id="at0005"]/parts[archetype_id="at0010"]/value*
- *Preferred_Slot_Path:/items[archetype_id="at0005"]*

Among the candidate paths returned by the algorithm, the first is the path */items[archetype_id="at0005"]*. This path is selected as a valid and suitable location for the slot reference in the initial EHR archetype. The path is valid because it points to a cluster component, which can contain the augmentation archetype (whose root component is also a cluster) according to the ISO13606 reference model. The path is suitable because it points to the container component of the medication coded value, which is the EOI used to obtain the augmentation content. Locating the augmentation content close to the medication coded values enriches medication data with information about drug–drug interactions.

Finally, the augmented archetype may be automatically generated starting from the initial EHR archetype, the augmentation archetype, and the path for the slot reference to the augmentation archetype. Figure 8 shows the resulting augmented archetype in contrast to the initial EHR archetype.



(**a**) (**b**)

Figure 8. Archetypes' visualization from LinkEHR. (**a**) The initial EHR archetype *CEN-EN13606-ENTRY.Medicacion.v1*. (**b**) The augmented archetype *CEN-EN13606-ENTRY.MedicacionAugmented.v1*.

4. Results

Defining and modeling an augmented EHR is a design process that entails performing tasks manually. However, once an augmented EHR is defined for a given initial EHR archetype, it can be applied to any instance of that initial EHR archetype. This section presents the design of a software prototype to run the data exchanges and queries needed to generate augmented EHR extracts from the initial EHR extracts. The workflow (Figure 9) starts from the initial EHR extract and the identifier of the augmented EHR, and finalizes by returning an augmented EHR extract.

The *Extraction* module identifies and obtains the EOI values from the initial EHR extract. The *Augmentation* module takes the EOI values as inputs, and executes the procedure described in the section Materials and Methods (B) to build the EOI binding triples. These triples are injected in the query graph of a SPARQL query to obtain the augmentation content from a LOD dataset, as stated in section Materials and Methods (C). The augmentation content is normalized according to the augmentation archetype (section Materials and Methods (D)). Finally, the *Integration* module generates the augmented EHR extract starting from the initial EHR extract and the normalized augmentation content, as described in section Materials and Methods (E).

A software prototype was implemented using the XQuery scripting language to develop the *Extraction* module, the *Integration* module, and specific steps performed by the *Augmentation* module such as the building of the EOI binding triples and the normalization of the augmentation content. The benefits of using XQuery are, first, to take advantage of the transformation scripts that the LinkEHR platform generates when compiling mapping specifications (i.e., data exchange 1 and 2 at data level in Figure 1); and, second, the fact that XQuery is designed for the extraction and manipulation of XML data eases the tasks of extracting EOI values in the *Extraction* module and the building of EOI binding triples in the *Augmentation* module. Additionally, the SPARQL query language is employed in the *Augmentation* module to obtain the augmentation content, and Java programming language is used in the prototype to orchestrate the workflow and communications between the involved modules.

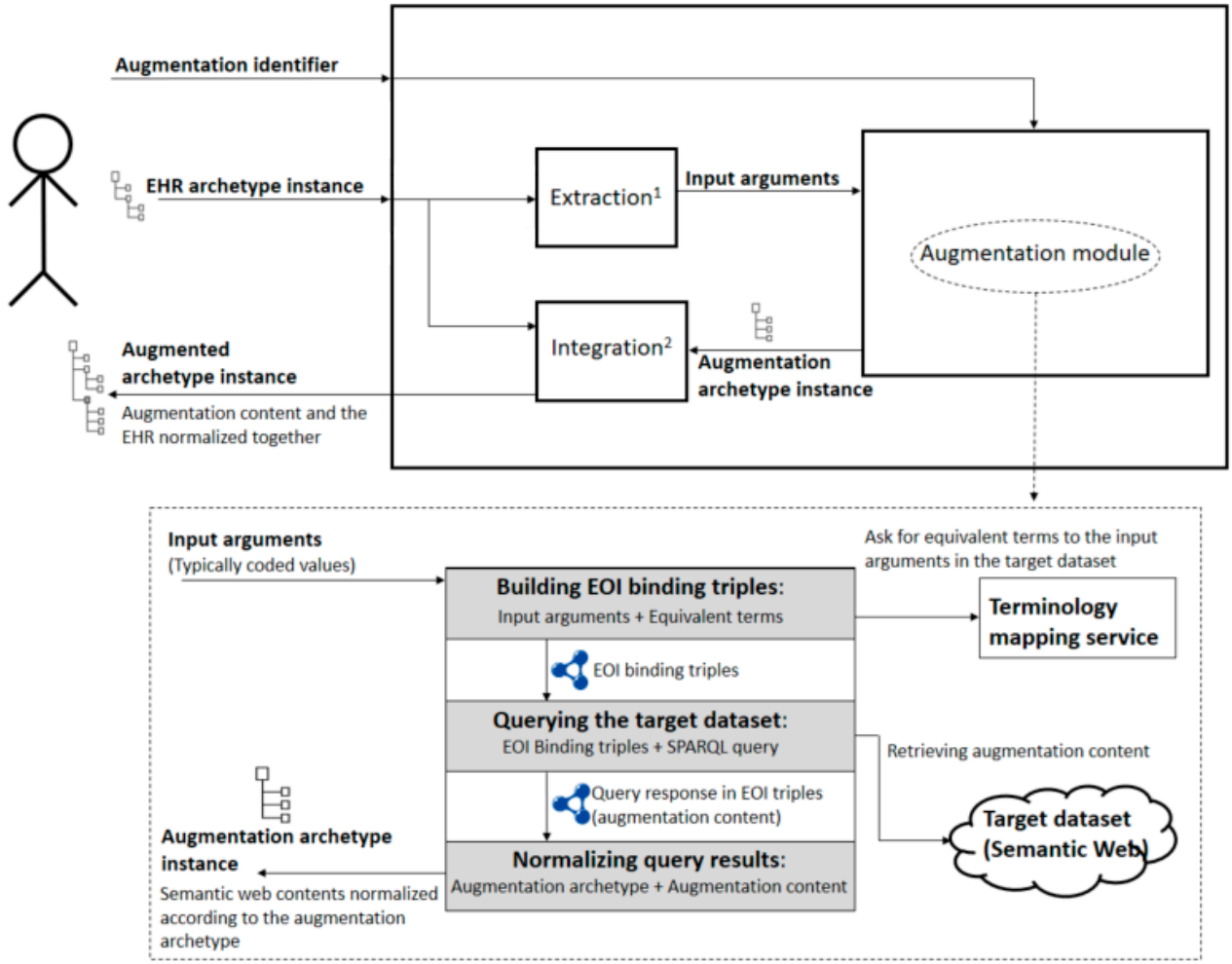


Figure 9. Description of the process to obtain an augmented EHR extract (i.e., execution cycle of the augmented EHR). 1. *Extraction*: Reads the EOIs specified by paths and provides them to the *Augmentation* module. 2. *Integration*: Generates instances of the augmented archetype by combining the EHR archetype instance and the augmentation archetype instance.

Finally, this prototype also includes a repository of available EHR augmentations, which is omitted from Figure 9 for the sake of clarity. For each EHR augmentation, this repository holds:

- (I) The XQuery script to build the EOI binding triples starting from the EOI coded values
- (II) The SPARQL query to collect the augmentation content
- (III) The augmentation archetype
- (IV) The XQuery script to normalize the augmentation content according to the augmentation archetype
- (V) A list of initial EHR archetypes this augmentation may enrich. For each initial EHR archetype, the repository contains:
    - o Archetype paths pointing to the EOI elements in the initial EHR archetype, required to extract EOI coded values from initial EHR extracts
    - o The augmented archetype
    - o The XQuery script to generate augmented EHR extracts starting from initial EHR extracts and normalized augmentation contents

To set up new EHR augmentations in the prototype requires manually adding this information (I through V) to the repository. The prototype is intended to be integrated with an EHR system in compliance with EHR archetype-based standards. The integration

must be carried out by IT professionals. Once integrated, it allows users (e.g., physicians or other healthcare professionals) to automatically access augmented EHR extracts as they usually access patients' EHRs. The system checks the repository for appropriate EHR augmentations, by searching for initial EHR archetypes in (V) matching the initial EHR extract. Then, the process starts as described in Figure 9 for the identified EHR augmentations. Results comprise augmentation contents formatted as EHR extracts, which are sent to the EHR system to be rendered in the EHR viewer, providing users augmented EHR extracts with specific EHR augmentations on demand.

5. Use Case

The summary EHR archetype defined by the Spanish Ministry of Health (*CEN-EN13606-COMPOSITION.HistoriaClinicaResumida.v1*) for the communication of health data among regional health institutions [23] is an appropriate initial EHR archetype to build a large use case of an augmented EHR. It encompasses elements concerning the patients' EHRs, which can be EOIs to enrich the archetype with several EHR augmentations. This section shows the design process of the augmented summary EHR archetype (*CEN-EN13606-COMPOSITION.HistoriaClinicaResumidaAugmented.v1*) from scratch, as a result of enriching the summary EHR archetype with EHR augmentations related to drug–drug interactions and disease-related treatments.

The EOIs selected from the summary EHR archetype are patients' active medication and active episodes of patients' diseases, located at paths

/content[archetype_id="at0043"]/members[archetype_id="at0024"]/members[archetype_id="at0053"]/members[archetype_id="at0054"]/items[archetype_id="at0172"]/parts[archetype_id="at0181"]/value

and

/content[archetype_id="at0043"]/members[archetype_id="at0042"]/members[archetype_id="at0046"]/items[archetype_id="at0158"]/value

respectively. These EOIs are intended to hold coded values from SNOMED-CT terminology, which can be augmented with potentially interesting additional information from public Semantic Web datasets. Figure 10 shows the initial EHR archetype and the initial EHR extract showing the selected EOIs.

Drug information to enrich a patient's active medication with drug–drug interactions can be found in the DrugBank database. Information on treatments for specific diseases to enrich the active episodes of patients' diseases with existing treatments and their effects is available on the NDFRT ontology. Equivalencies between coded terms from source (SNOMED-CT) and target (DrugBank, NDFRT) terminologies are identified using the BioPortal terminology mapping service. The EOI binding triples to establish conceptual mappings between source and target terminologies are built by requesting such a service for mappings between source and target terminologies for the EOI coded terms. A LOD representation of the target terminologies is required to obtain the augmentation content through SPARQL queries. The LOD cloud contains a LOD representation of the DrugBank database (Bio2RDF DrugBank), whereas the NDFRT terminology expressed as LOD is located at the BioPortal ontology repository.

The automatic procedure described in section Materials and Methods (B) is used to build the EOI binding triples taking several arguments as inputs, where only the initial EHR extract must be provided at runtime. The other arguments are defined as part of the EHR augmentation at design time. Tables 1 and 2 summarize the required arguments for both augmentations. In Table 1, note that the URL prefix of data items in the target dataset differs from the identifier URL of the target terminology returned by the mapping service. In this case, the URL prefix of data items in the target dataset can be identified by querying VoID descriptors as explained in section Materials and Methods (A).

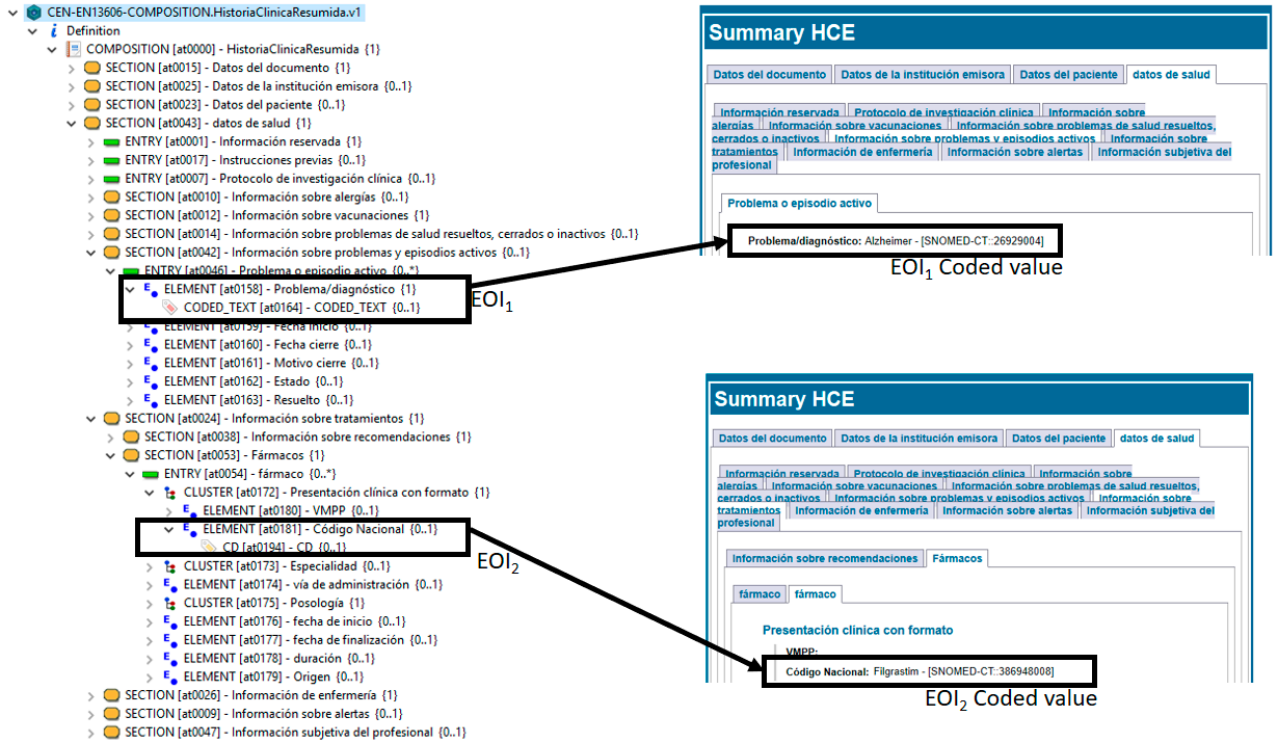


Figure 10. On the left, a LinkEHR visualization of the initial EHR archetype *CEN-EN13606-COMPOSITION. HistoriaClinicaResumida.v1*. On the right, two views of the initial EHR extract presented in HTML. The EOIs' active episode of patients' diseases (EOI_1) and patients' active medication (EOI_2) and their coded values are highlighted on both the initial EHR archetype and the initial EHR extract.

Table 1. Inputs and output of the automatic procedure to build the EOI binding triples for the drug–drug interactions' EHR augmentation.

	Variable	Value
Inputs	Initial EHR Extract	An instance of the summary EHR archetype provided at runtime
	Path to the EOI in the initial EHR archetype	*/content[archetype_id="at0043"] /members[archetype_id="at0024"] /members[archetype_id="at0053"] /members[archetype_id="at0054"] /items[archetype_id="at0172"] /parts[archetype_id="at0181"]/value*
	The predicate URL	*http://www.w3.org/2004/02/skos/core#exactMatch*
	The name of the source terminology employed by the EOI coded values	*SNOMEDCT*
	The identifier URL of the target terminology/ontology returned by the terminology mapping service	*http://www.drugbank.ca/drugs*
	The URL prefix of data items in the target dataset	*http://bio2rdf.org/drugbank:*
Output	The EOI binding triples	RDF triples that relates EOIs from the initial EHR extract with data items in the target dataset

Table 2. Inputs and output of the automatic procedure to build the EOI binding triples for the disease-related treatments' EHR augmentation.

	Variable	Value
Inputs	Initial EHR Extract	An instance of the summary EHR archetype provided at runtime
	Path to the EOI in the initial EHR archetype	*/content[archetype_id="at0043"] /members[archetype_id="at0042"] /members[archetype_id="at0046"] /items[archetype_id="at0158"]/value*
	The predicate URL	*http://www.w3.org/2004/02/skos/core#exactMatch*
	The name of the source terminology employed by the EOI coded values	*SNOMEDCT*
	The identifier URL of the target terminology/ontology returned by the terminology mapping service	*http://purl.bioontology.org/ontology/NDFRT*
	The URL prefix of data items in the target dataset	*http://purl.bioontology.org/ontology/NDFRT*
Output	The EOI binding triples	RDF triples that relates EOIs from the initial EHR extract with data items in the target dataset

The resulting EOI binding triples for the drug–drug interactions' EHR augmentation are generated using inputs from Table 1 and the initial EHR extract introduced in Figure 10, whose EOI (patient's active medication) coded value is Filgrastim-[SNOMED-CT::386948008]. The EOI binding triples for the disease-related treatments EHR augmentation are built using inputs from Table 2 and the initial EHR extract introduced in Figure 10, whose EOI (active episodes of the patient's diseases) coded value is Alzheimer-[SNOMED-CT::26929004]. Table 3 summarizes information related to the EOI binding triples obtained for this use case.

Table 3. EOI binding triples generated for each EHR augmentation taking the initial EHR extract introduced in Figure 10 and the arguments described in Tables 1 and 2, respectively.

EHR Augmentation	EOI	EOI Coded Value	EOI Binding Triple
Drug-drug interactions	patient's active medication	*Filgrastim-[SNOMED-CT::386948008]*	<*http://purl.bioontology.org/ontology/SNOMEDCT/386948008*> <*http://www.w3.org/2004/02/skos/core#exactMatch*> <*http://bio2rdf.org/drugbank:DB00099*>
Disease-related treatments	active episodes of patient's diseases	*Alzheimer-[SNOMED-CT::26929004]*	<*http://purl.bioontology.org/ontology/SNOMEDCT/26929004*> <*http://www.w3.org/2004/02/skos/core#exactMatch*> <*http://purl.bioontology.org/ontology/NDFRT/N0000000363*>

The EOI binding triples are injected at runtime in a SPARQL query as part of the query graph to collect the augmentation content. Figure 11 shows the SPARQL queries for the EHR augmentations' drug–drug interactions and disease-related treatments.

The augmentation archetypes to normalize the augmentation content are modeled at design time. Figure 12 shows the drug–drug interactions' augmentation archetype (*CEN-EN13606-CLUSTER.DrugDrugInteraction.v1*) and the disease-related treatments' augmentation archetype (*CEN-EN13606-CLUSTER.DiseaseTreatment.v1*). These augmentation archetypes were modeled using LinkEHR. In both augmentation archetypes, the root element is a cluster containing at least one element for each variable in the SPARQL query to hold the augmentation content.

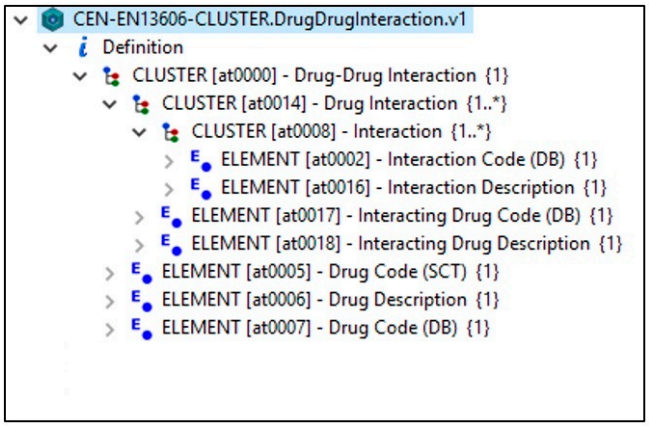
```
PREFIX sct: <http://purl.bioontology.org/ontology/SNOMEDCT/>
PREFIX sch: <http://schema.org/>
PREFIX db: <http://bio2rdf.org/drugbank:>
PREFIX dbv: <http://bio2rdf.org/drugbank_vocabulary:>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

SELECT DISTINCT
?reference_drug_name
?reference_drug_code_sct
?reference_drug_code_db
?target_drug_name
?target_drug_code_db
?interaction_description
?interaction_code_db

WHERE {

?reference_drug_code_sct skos:exactMatch ?reference_drug_code_db .
SERVICE <http://bio2rdf.org/sparql> {
        ?interaction_code_db rdf:type dbv:Drug-Drug-Interaction .
             ?reference_drug_code_db dbv:ddi-interactor-in ?interaction_code_db .
        ?target_drug_code_db dbv:ddi-interactor-in ?interaction_code_db .
        ?target_drug_code_db dcterms:title ?target_drug_name .
        ?reference_drug_code_db dcterms:title ?reference_drug_name .
        ?interaction_code_db dcterms:title ?interaction_description .
        FILTER(?reference_drug_code_db != ?target_drug_code_db)
  }
}
```

(**a**)

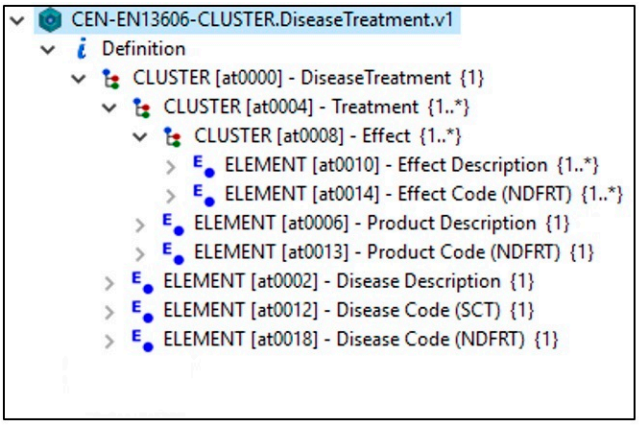
```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ndfrt: <http://purl.bioontology.org/ontology/NDFRT/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX sct: <http://purl.bioontology.org/ontology/SNOMEDCT/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

SELECT DISTINCT
?disease_code_sct
?disease_code_ndfrt
?disease_description
?treatment_code_ndfrt
?treatment_description
?effect_code_ndfrt
?effect_description
WHERE
{
    ?disease_code_sct skos:exactMatch ?disease_code_ndfrt .
    SERVICE <http://sparql.bioontology.org/sparql?apikey=[APIKEY]> {
        ?disease_code_ndfrt ndfrt:may_be_treated_by ?treatment_code_ndfrt .
        ?treatment_code_ndfrt ndfrt:has_product_component ?prod .
        ?treatment_code_ndfrt skos:prefLabel ?treatment_description .
        ?treatment_code_ndfrt ndfrt:has_physiologic_effect ?effect_code_ndfrt .
        ?effect_code_ndfrt skos:prefLabel ?effect_description .
        ?disease_code_ndfrt skos:prefLabel ?disease_description .
    }
}
```

(**b**)

Figure 11. SPARQL queries to obtain the augmentation contents. (**a**) The query to collect the drug–drug interactions. This retrieves interaction descriptions, codes, and names of the drugs involved in each drug–drug interaction. (**b**) The query to retrieve disease-related treatments. This obtains treatments and descriptions related to a given disease. In both cases, queries are parametrized using the EOI binding triples as part of the query graph, and the *skos:exactMatch* predicate allows filtering the augmentation content by retrieving only the information related to the coded values of the EOI binding triples.



(**a**)



(**b**)

Figure 12. Augmentation archetypes' visualization from LinkEHR. (**a**) The *CEN-EN13606-CLUSTER. DrugDrug-Interaction.v1* augmentation archetype for the drug–drug interactions EHR augmentation. (**b**) The *CEN-EN13606-CLUSTER.DiseaseTreatment.v1* augmentation archetype for the disease-related treatments' EHR augmentation.

Figure 13 shows the augmented archetype *CEN-EN13606-COMPOSITION. Historia-ClinicaResumidaAugmented.v1*, which results from combining the augmentation archetypes and the initial EHR archetype. LinkEHR was used to model the required archetypes, and to define the mapping relationships for these archetypes and their respective sources. Table 4 establishes the mappings carried out to define the augmented summary EHR. Note that the only archetype existing before the definition of an augmented EHR is the initial EHR archetype (*CEN-EN13606-COMPOSITION.HistoriaClinicaResumida.v1*). The

augmentation archetypes and the augmented archetype must be modeled as part of the process of designing an augmented EHR.

The design stage concludes once the required resources have been identified and/or defined: EOI paths, target dataset, terminological mapping service, SPARQL queries, arguments to build EOI binding triples, augmentation archetypes, and augmented archetype, in addition to their corresponding mappings. At runtime, only an instance of the initial EHR archetype is required to execute the workflow and generate an instance of the augmented archetype. Figure 14 shows the augmentation contents for the use case, normalized according to the augmentation archetypes. Figure 15 illustrates the instance of the augmented archetype generated for the use case.

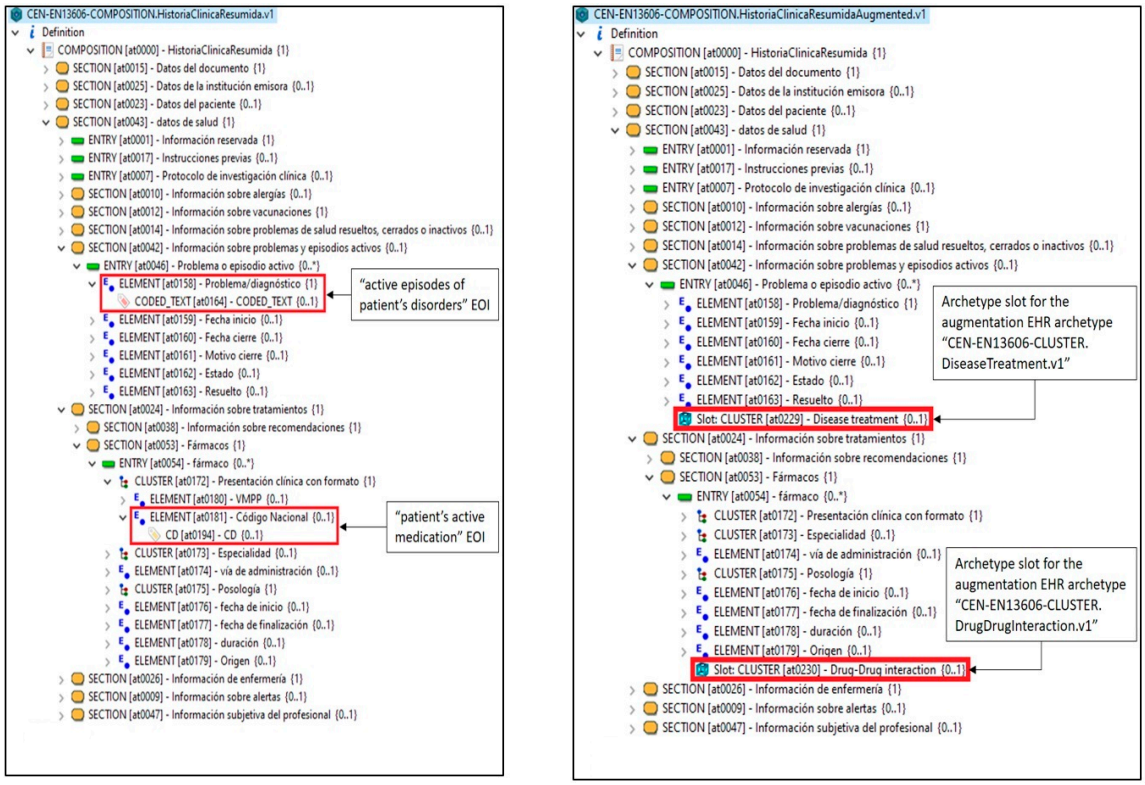
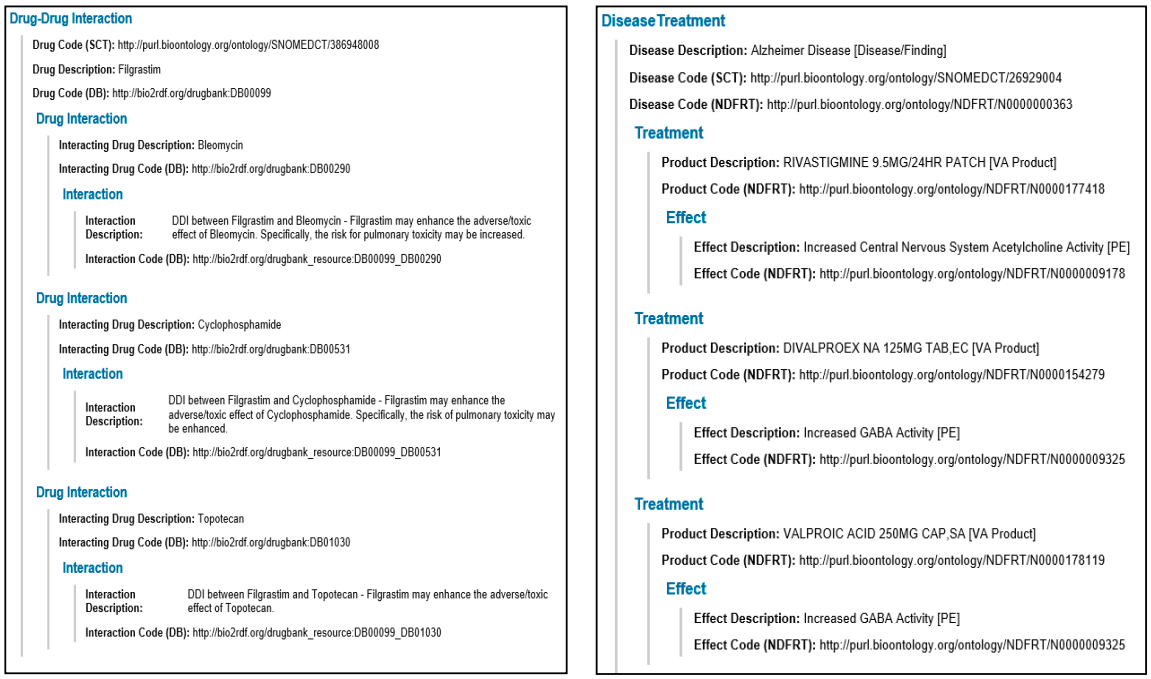


Figure 13. Archetypes' visualization from LinkEHR. (**a**) The initial EHR archetype *CEN-EN13606-COMPOSITION.HistoriaClinicaResumida.v1*. The EOIs used in this use case are highlighted. (**b**) The augmented archetype *CEN-EN13606-COMPOSITION.HistoriaClinicaResumidaAugmented.v1*, which is similar to the initial EHR archetype but includes archetype slots for the augmentation archetypes.

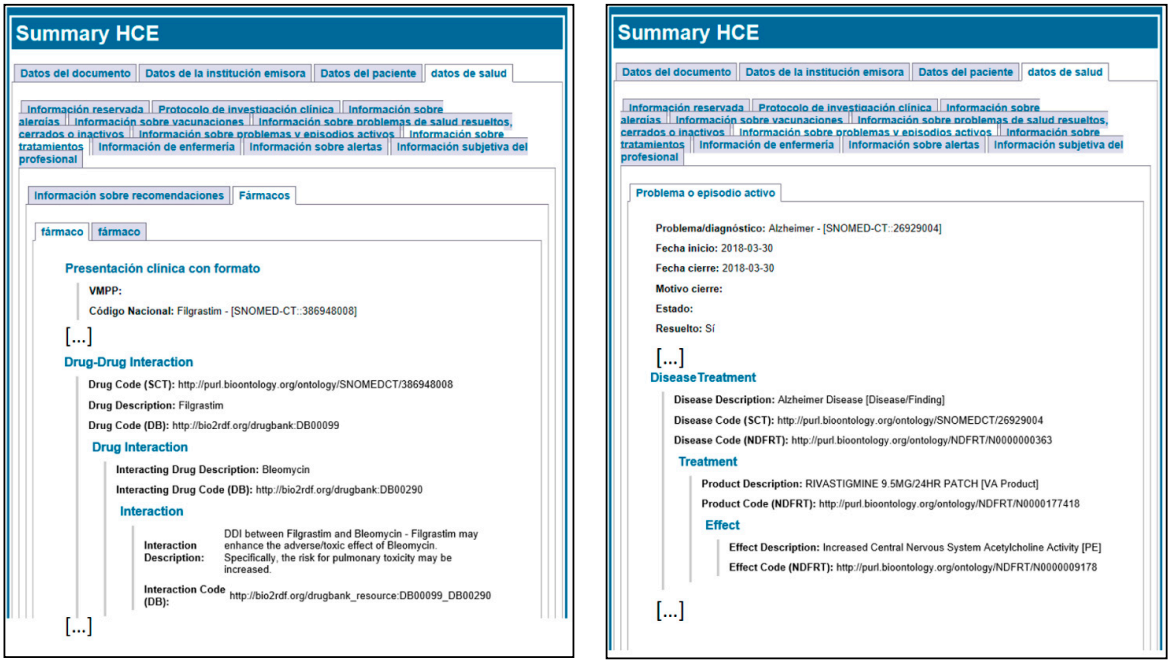
Table 4. List of mapping specifications defined to obtain the augmented summary EHR. The mappings were stated using the LinkEHR platform. For each mapping specification, LinkEHR built a XQuery script whose execution converts data instances of the source schema to data instances of the target schema.

Resource	Source Schema(s)	Target Schema
Drug-drug interactions augmentation content	XML Schema for SPARQL query results	*CEN-EN13606-CLUSTER.DrugDrugInteraction.v1*
Disease-related treatments augmentation content	XML Schema for SPARQL query results	*CEN-EN13606-CLUSTER.DiseaseTreatment.v1*
Augmented Summary EHR	The summary EHR archetype *CEN-EN13606-COMPOSITION.HistoriaClinicaResumida.v1* The drug-drug interactions archetype *CEN-EN13606-CLUSTER.DrugDrugInteraction.v1* The disease-related treatments archetype *CEN-EN13606-CLUSTER.DiseaseTreatment.v1*	*CEN-EN13606-COMPOSITION.HistoriaClinicaResumidaAugmented.v1*



(**a**) (**b**)

Figure 14. HTML representation of augmentation contents. (**a**) The augmentation content of the drug–drug interactions' EHR augmentation, formatted according to the augmentation archetype *CEN-EN13606-CLUSTER. DrugDrugInteraction.v1*. (**b**) The augmentation content of the disease-related treatments' EHR augmentation, formatted according to the augmentation archetype *CEN-EN13606-CLUSTER. DiseaseTreatment.v1*.



(**a**) (**b**)

Figure 15. An HTML representation of the augmented summary EHR extract, an instance of the augmented archetype *CEN-EN13606-COMPOSITION.HistoriaClinicaResumidaAugmented.v1* including the augmented content from (**a**) drug–drug interactions' and (**b**) disease-related treatments' EHR augmentations.

6. Discussion and Conclusions

This work introduces the augmented EHR concept as the enrichment of an existing EHR with additional contents (i.e., information and/or knowledge) derived from different external information sources. Furthermore, a preliminary approach to build the augmented EHR using Semantic Web sources is proposed by enriching EHRs with patient-related data (at the instance level) and/or data related to medical concepts (at the information model level). Additionally, practical foundations to express Semantic Web data as normalized EHR extracts are defined. This solution provides an alternative means to bridge the gap between healthcare IT standards and the Semantic Web, by transforming Semantic Web contents into EHR extracts. Moreover, note that the approach can be extended to enrich the EHR with additional contents derived from external sources other than those from the Semantic Web, by designing intermediate transformations to build a semantic representation of such sources, and then applying this method taking such a semantic representation as an external source.

Several previous works have addressed combining the EHR with other information sources [11–14,16,30–32], most of which have focused on converting EHR data to Semantic Web formats. These approaches aim at reasoning with the data, and thus infer new facts from the patient's EHR. However, Semantic Web formats are not natively supported by EHR systems. The presented work introduces an approach in the opposite direction, aiming to bring additional contents from Semantic Web sources to the EHR of a given patient, in such a way that normalized healthcare IT systems and services can leverage these contents. Other works address methods to incorporate additional potentially relevant information into the EHR [17–19,33]. Nevertheless, this information (i.e., the augmentation content) is not normalized according to an EHR standard, leading to a limited degree of interoperability, in contrast to the augmented EHR approach described in this paper, in which the content from external sources is normalized and integrated as a part of the EHR. For example, this allows processing of the augmented EHR in subsequent transformations as a normalized EHR, processing both the augmentation content and the patient EHR data as the same entity (i.e., the EHR), which may be especially useful when feeding the EHR with data from other information systems, such as third-party EHR systems, clinical decision support systems (CDS), or computerized clinical guidelines (CCG). Special mention is necessary of Infobuttons [24], incorporated as part of the HL7 standard (HL7's Context-Aware Knowledge Retrieval), which is able to retrieve external information from the point of care (i.e., EHR or PHR). However, our work improves such an approach in two ways. First, our methodology allows retrieving relevant information from the Semantic Web, which offers significant advantages in contrast with other kinds of information sources, such as reasoning over data and dealing with heterogeneous data sources [34]. The Infobuttons approach does not specify how to integrate contents from the Semantic Web into the EHR. In addition, our solution establishes a method to retrieve additional relevant information as EHR extracts compliant with EHR archetypes, enhancing interoperability with other EHR systems compliant with EHR archetype-based standards. This is in contrast to the Infobuttons approach, which does not state methods to retrieve the additional data in compliance with EHR archetypes, limiting the interoperability of the retrieved information with other EHR systems compliant with EHR archetype-based standards.

Works dealing with EHR data and CDS integration highlight the lack of a standard API to access EHR data [35,36]. Despite the current lack of a standard API, the inputs of CDS systems, and an approach to extract these inputs from the EHR data, can be defined through standard EHR models [12,13,37,38]. Thus, combining this idea with the augmented EHR approach may lead to a more accurate and efficient exploitation of EHR data for secondary uses, given that data from the original EHR and from external sources can be processed starting from the augmented EHR (or an abstraction thereof).

In this work, EHR augmentations of potential interest were identified in the context of standard EHRs using archetypes and available Semantic Web resources. Starting from these augmentations, a method to obtain an implementation that serves for such augmentations

was designed and implemented. Although this method was designed in close connection with the EHR schema, the EHR augmentations can be modeled regardless of the EHR extract that will use them. Therefore, a varied set of EHR augmentations could potentially be provided for users, who would be able to choose the most convenient EHR augmentation to enrich specific EHR extracts with additional contents. Furthermore, augmentations were designed to be reused not only to enrich existing EHR archetypes, but also results from CDS systems.

At present, most of the process of designing and implementing an EHR augmentation is manual. Existing Semantic Web tools to discover equivalent relationships between EHR coded values and terms from the LOD cloud were explored for this work. However, most schema definitions are not necessarily well formed in the domain of LOD [39] and cannot be seamlessly crawled. In addition, the quality of these LOD datasets in terms of completeness, consistency, conciseness, and interlinking remains a challenge [40]. VoID descriptors allow identification of public LOD datasets when not provided by the organization supporting the ontology. Nevertheless, not all of the published datasets provide VoID descriptors, and those providing VoID descriptors show the relevant information (e.g., the URL prefix of data items in the target dataset) located in different properties for each VoID descriptor. To promote the automatization of the augmented EHR design process, we proposed an algorithm to suggest where to place the slot for the augmentation archetype in the augmented EHR archetype. The algorithm is based on the syntactic structure of the initial EHR archetype, and considers semantic relationships by trying to identify a valid position near to the EOI (e.g., a sibling of the EOI, under the same parent component in the archetype).

Although the proposed solution requires some manual interventions, it provides the advantage that at execution time it is completely automatic. Therefore, once an augmented EHR is defined, specific patient EHR extracts can be automatically augmented without restrictions or manual interventions. The presented method was designed aiming to be as automatic as possible, considering the current limitations of RDF schemas and dataset descriptions in LOD. The main aim of the process requiring manual intervention is the identification of potentially relevant augmentation content. This is a portion of linked data from the Semantic Web whose potential relevance must be established by clinical experts. However, there are tools that can assist in the process of designing augmented EHRs; for example, linked data crawlers such as LDSpider [41] could be used to discover potentially interesting augmentation contents starting from the EOI. In addition, visual SPARQL query editors such as Gruff [42] may assist in the exploration of the augmentation content and can be helpful to build the corresponding SPARQL queries. Artificial intelligence methods can help to automatize this process and reduce manual interventions, but we consider this would be part of a different work. In addition, future research should examine how this method can be applied to enrich the EHR with information and/or knowledge from heterogeneous sources other than Semantic Web sources, and to define new approaches to automatize the design of augmented EHRs; for example, improved tools to discover augmentations of potential interest could automatically explore terminological annotations of an EHR schema, and determine the candidate EHR augmentations to enrich a given EHR extract.

Author Contributions: Conceptualization, A.M.-G., J.A.M., M.M.; methodology, A.M.-G., J.A.M., D.B.; software, A.M.-G., D.B.; validation, A.M.-G., J.A.M., D.B.; formal analysis, A.M.-G.; investigation, A.M.-G.; writing—original draft preparation, A.M.-G.; writing—review and editing, A.M.-G., J.A.M., M.M., M.R.; supervision, J.A.M., M.M., M.R. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by the Spanish Ministry of Economy and Competitiveness and the FEDER program through grant TIN2014-53749-C2-1-R, and by the Spanish Ministry of Science, Innovation and Universities through grant PTQ2018-009924.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. De Boer, D.; Delnoij, D.; Rademakers, J. The Importance of Patient-Centered Care for Various Patient Groups. *Patient Educ. Couns.* **2013**, *90*, 405–410. [CrossRef] [PubMed]
2. Bowman, S. Impact of Electronic Health Record Systems on Information Integrity: Quality and Safety Implications. *Perspect. Health Inf. Manag.* **2013**, *10*, 1.
3. Boytcheva, S.; Angelova, G.; Angelov, Z.; Tcharaktchiev, D.; Vodenicharov, V. Enrichment of EHR with Linked Open Data for Risk Factors Identification. In Proceedings of the ACM International Conference Proceeding Series, Ruse, Bulgaria, 21–22 June 2019; pp. 84–90.
4. Weiskopf, N.G.; Cohen, A.M.; Hannan, J.; Jarmon, T.; Dorr, D.A. Towards Augmenting Structured EHR Data: A Comparison of Manual Chart Review and Patient Self-Report. *AMIA Annu. Symp. Proc. AMIA Symp.* **2020**, *2019*, 903–912.
5. Bizer, C.; Heath, T.; Idehen, K.; Berners-Lee, T. Linked Data on the Web (LDOW2008). In Proceedings of the 17th international conference on World Wide Web-WWW '08, Beijing, China, 21–25 April 2008; pp. 1265–1266.
6. The Linking Open Data Cloud Diagram. Available online: http://lod-cloud.net/ (accessed on 19 September 2020).
7. Wishart, D.S. DrugBank: A Comprehensive Resource for in Silico Drug Discovery and Exploration. *Nucleic Acids Res.* **2006**, *34*, D668–D672. [CrossRef]
8. National Drug File-Reference Terminology Source Information. Available online: https://www.nlm.nih.gov/research/umls/sourcereleasedocs/current/NDFRT/index.html (accessed on 9 September 2020).
9. Boscá, D.; Maldonado, J.A.; Moner, D.; Robles, M. Automatic Generation of Computable Implementation Guides from Clinical Information Models. *J. Biomed. Inform.* **2015**, *55*, 143–152. [CrossRef] [PubMed]
10. Shah, U.; Finin, T.; Joshi, A.; Cost, R.S.; Matfield, J. Information Retrieval on the Semantic Web. In Proceedings of the International Conference on Information and Knowledge Management (CIKM'02), McLean, VA, USA, 4–9 November 2002; pp. 461–468.
11. Lezcano, L.; Sicilia, M.-A.; Rodriguez-Solano, C. Integrating Reasoning and Clinical Archetypes using OWL Ontologies and SWRL Rules. *J. Biomed. Inform.* **2011**, *44*, 343–353. [CrossRef]
12. Tao, C.; Jiang, G.A.; Oniki, T.; Freimuth, R.R.; Zhu, Q.; Sharma, D.; Pathak, J.; Huff, S.M.; Chute, C.G. A Semantic-Web Oriented Representation of the Clinical Element Model for Secondary Use of Electronic Health Records Data. *J. Am. Med. Inform. Assoc.* **2012**, *20*, 554–562. [CrossRef]
13. Fernández-Breis, J.T.; Maldonado, J.A.; Marcos, M.; Legaz-García, M.D.C.; Moner, D.; Torres-Sospedra, J.; Esteban-Gil, A.; Martínez-Salvador, B.; Robles, M. Leveraging Electronic Healthcare Record Standards and Semantic Web Technologies for the Identification of Patient Cohorts. *J. Am. Med. Inform. Assoc.* **2013**, *20*, e288–e296. [CrossRef] [PubMed]
14. Odgers, D.J.; Dumontier, M. Mining Electronic Health Records using Linked Data. *AMIA Jt. Summits Transl. Sci. Proc.* **2015**, *2015*, 217–221.
15. Mozaffarian, D.; Benjamin, E.J.; Go, A.S.; Arnett, D.K.; Blaha, M.J.; Cushman, M.; De Ferranti, S.; Després, J.-P.; Fullerton, H.J.; Howard, V.J.; et al. Heart Disease and Stroke Statistics—2015 Update. *Circulation* **2015**, *131*, e29-322. [CrossRef]
16. Kilintzis, V.; Chouvarda, I.; Beredimas, N.; Natsiavas, P.; Maglaveras, N. Supporting Integrated Care with a Flexible Data Management Framework Built Upon Linked Data, HL7 FHIR and Ontologies. *J. Biomed. Inform.* **2019**, *94*, 103179. [CrossRef]
17. Cimino, J.J.; Del Fiol, G. Infobuttons and Point of Care Access to Knowledge. In *Clinical Decision Support*; Elsevier BV: Amsterdam, The Netherlands, 2007; pp. 345–371.
18. Alfano, M.; Lenzitti, B.; Bosco, G.L.; Muriana, C.; Piazza, T.; Vizzini, G. Design, Development and Validation of a System for Automatic Help to Medical Text Understanding. *Int. J. Med. Inform.* **2020**, *138*, 104109. [CrossRef] [PubMed]
19. Chetta, A.; Carrington, J.M.; Forbes, A.G. Augmenting EHR Interfaces for Enhanced Nurse Communication and Decision Making. In Proceedings of the ACM International Conference Proceeding Series, Chicago, IL, USA, 25 October 2015; Volume 4, pp. 1–6.
20. SPARQL Query Results XML Format (Second Edition). Available online: https://www.w3.org/TR/rdf-sparql-XMLres/ (accessed on 9 September 2020).
21. Polytechnic University of Valencia, Biomedical Informatics Group. LinkEHR Platform. Available online: http://www.linkehr.com (accessed on 21 September 2020).
22. Describing Linked Datasets with the VoID Vocabulary. Available online: https://www.w3.org/TR/void/ (accessed on 9 September 2020).
23. Spanish National Health System, Resources for Clinical Modeling (Archetypes). Available online: https://www.msssi.gob.es/en/profesionales/hcdsns/areaRecursosSem/Rec_mod_clinico_arquetipos.htm (accessed on 21 September 2020).
24. SNOMED International. Available online: http://www.snomed.org/snomed-ct (accessed on 21 September 2020).
25. Whetzel, P.L.; Noy, N.F.; Shah, N.H.; Alexander, P.R.; Nyulas, C.; Tudorache, T.; Musen, M.A. BioPortal: Enhanced Functionality via New Web Services from the National Center for Biomedical Ontology to Access and use Ontologies in Software Applications. *Nucleic Acids Res.* **2011**, *39*, W541–W545. [CrossRef] [PubMed]

26. NCBO BioPortal-Repository of Biomedical Ontologies. Available online: http://bioportal.bioontology.org/ (accessed on 21 September 2020).
27. SKOS Simple Knowledge Organization System Namespace Document-HTML Variant. Available online: https://www.w3.org/2009/08/skos-reference/skos.html#exactMatch (accessed on 27 April 2021).
28. Bio2RDF. Linked Data for the Life Sciences. Available online: https://bio2rdf.org/ (accessed on 27 April 2021).
29. DrugBank. A Pharmaceutical Knowledge Base. Available online: http://www.drugbank.ca (accessed on 27 April 2021).
30. Miñarro-Gimenez, J.A.; Madrid, M.; Fernandez-Breis, J.T. OGO: An Ontological Approach for Integrating Knowledge about Orthology. *BMC Bioinform.* **2009**, *10*, S13. [CrossRef]
31. Pathak, J.; Kiefer, R.C.; Chute, C.G. Applying Linked Data Principles to Represent Patient's Electronic Health Records at Mayo clinic. In Proceedings of the 2nd ACM SIGHIT Symposium on International Health Informatics-IHI '12, Miami, FL, USA, 28–30 January 2012; pp. 455–464.
32. Maldonado, J.A.; Marcos, M.; Fernández-Breis, J.T.; Giménez-Solano, V.M.; Legaz-García, M.D.C.; Martínez-Salvador, B. CLIN-IK-LINKS: A Platform for the Design and Execution of Clinical Data Transformation and Reasoning Workflows. *Comput. Methods Programs Biomed.* **2020**, *197*, 105616. [CrossRef]
33. Konstantinidis, S.T.; Kummervold, P.E.; Luque, L.F.; Vognild, L.K. A Proposed Framework to Enrich Norwegian EHR System with Health-Trusted Information for Patients and Professionals. *Stud. Health Technol. Inform.* **2015**, *213*, 149–152.
34. Chung, D.S.; Tai, D.; O'Sullivan, P.B. The Mouse Approach: Mapping Ontologies using UML for System Engineers. *Comput. Rev. J.* **2018**, *1*, 8–29.
35. González-Ferrer, A.; Peleg, M. Understanding Requirements of Clinical Data Standards for Developing Interoperable Knowledge-based DSS: A Case Study. *Comput. Stand. Interfaces* **2015**, *42*, 125–136. [CrossRef]
36. Zhang, M.; Velasco, F.T.; Musser, R.C.; Kawamoto, K. Enabling Cross-Platform Clinical Decision Support through Web-Based Decision Support in Commercial Electronic Health Record Systems: Proposal and Evaluation of Initial Prototype Implementations. *AMIA Annu. Symp. Proc. AMIA Symp.* **2013**, *2013*, 1558–1567.
37. Marcos, M.; Maldonado, J.A.; Martínez-Salvador, B.; Boscá, D.; Robles, M. Interoperability of Clinical Decision-Support Systems and Electronic Health Records using Archetypes: A Case Study in Clinical Trial Eligibility. *J. Biomed. Inform.* **2013**, *46*, 676–689. [CrossRef]
38. Khalique, F.; Khan, S.A. An FHIR-based Framework for Consolidation of Augmented EHR from Hospitals for Public Health Analysis. In Proceedings of the 2017 IEEE 11th International Conference on Application of Information and Communication Technologies (AICT), Moscow, Russia, 20–22 September 2017; pp. 1–4. [CrossRef]
39. Zhang, Z.; Gentile, A.L.; Blomqvist, E.; Augenstein, I.; Ciravegna, F. An Unsupervised Data-Driven Method to Discover Equivalent Relations in Large Linked Datasets. *Semant. Web* **2016**, *8*, 197–223. [CrossRef]
40. Issa, S.; Hamdi, F.; Cherfi, S.S.-S. Enhancing the Conciseness of Linked Data by Discovering Synonym Predicates. In *Knowledge Science, Engineering and Management. KSEM 2019*; Douligeris, C., Karagiannis, D., Apostolou, D., Eds.; Lecture Notes in Computer Science; Springer: Charm, Switzerland, 2019; Volume 11775, pp. 739–750. [CrossRef]
41. Isele, R.; Umbrich, J.; Bizer, C.; Harth, A. LDspider: An Open-Source Crawling Framework for the Web of Linked Data. In Proceedings of the 2010 International Semantic Web Conference (ISWC 2010), Shanghai, China, 7–11 November 2010; Volume 658, pp. 29–32.
42. Gruff-New Browser Based Version AllegroGraph. Available online: https://allegrograph.com/products/gruff/ (accessed on 9 September 2020).