

# A Methodological Approach for Ontologising and Aligning Health Level Seven (HL7) Applications \*

Ratnesh Sahay<sup>1</sup>, Ronan Fox<sup>1</sup>, Antoine Zimmermann<sup>2</sup>,  
Axel Polleres<sup>1,3</sup>, Manfred Hauswirth<sup>1</sup>

<sup>1</sup> Digital Enterprise Research Institute (DERI), NUIG  
IDA Business Park, Lower Dangan, Galway, Ireland  
`firstname.lastname@deri.org`

<sup>2</sup> INSA-Lyon, LIRIS, UMR5205, F-69621, France  
`firstname.lastname@insa-lyon.fr`

<sup>3</sup> Siemens AG Österreich, Siemensstrasse 90, 1210 Vienna, Austria

**Abstract.** Healthcare applications are complex in the way data and schemas are organised in their internal systems. Widely deployed healthcare standards like Health Level Seven (HL7) V2 are designed using flexible schemas which allow several choices when constructing clinical messages. The recently emerged HL7 V3 has a centrally consistent information model that controls terminologies and concepts shared by V3 applications. V3 information models are arranged in several layers (abstract to concrete layers). V2 and V3 systems raise interoperability challenges: firstly, how to exchange clinical messages between V2 and V3 applications, and secondly, how to integrate globally defined clinical concepts with locally constructed concepts. The use of ontologies for interoperable healthcare applications has been advocated by domain and knowledge representation specialists. This paper addresses two main areas of an ontology-based integration framework: (1) an ontology building methodology for the HL7 standard where ontologies are developed in separated global and local layers; and (2) aligning V2 and V3 ontologies. We propose solutions that: (1) provide a semi-automatic mechanism to build HL7 ontologies; (2) provide a semi-automatic mechanism to align HL7 ontologies and transform underlying clinical messages. The proposed methodology has developed HL7 ontologies of 300 concepts in average for each version. These ontologies and their alignments are deployed and evaluated under a semantically-enabled healthcare integration framework.

**Keywords:** Health Level Seven (HL7), Semantic Interoperability, Ontology Building Methodology, Ontology Alignment

---

\* This work is partly funded by Science Foundation Ireland (SFI) project Lion-2 (SFI/08/CE/I1380).

## 1 Introduction

In a large domain like healthcare, knowledge is represented in information models, clinical repositories (databases), ontologies for terminologies, vocabularies, etc. Considering the impact of this domain, standardisation bodies play a crucial role in defining all entities (*e.g.*, terminologies, codes, vocabularies, information models) related to the construction and exchange of clinical messages. Health Level Seven (HL7)<sup>4</sup> is the most widely deployed healthcare standard, which develops information models and schemas for constructing and exchanging clinical information across healthcare stakeholders. There are two major HL7 versions, HL7 Version 2 and HL7 Version 3, later on called V2 and V3. The majority of HL7 applications comply with V2. V3 is emerging and advocated by medical domain experts for greater consistency and interoperability of healthcare applications. Interoperability of HL7 versions is crucial to bridge the gap between two major types of deployments across the healthcare industry [1]. The lack of interoperability between standards (*e.g.*, HL7, openEHR<sup>5</sup>, CEN<sup>6</sup> TC/251 13606), and also within two versions of the same standard (*e.g.*, V2 and V3) result in a severe interoperability problem in the healthcare domain. Heterogeneity between V3 applications is usually less critical than between V2 applications because of the presence of a centralised information model in V3 that controls all the vocabularies and terminologies shared by the users. The presence of different healthcare standards, large scale applicability, and limitations of syntactic integration solutions, motivated the application of Semantic Web (SW) technologies and ontologies to resolve heterogeneity in a formal and consistent way.

In order to provide semantic interoperability for HL7-based applications, the initial development stage starts with important questions: how to build ontologies for such a vast and complex standard? Is there any methodological support available for building healthcare ontologies? Unfortunately, traditional ontology building methodologies have several limitations in dealing with concrete applications [7]: (i) more emphasis is given to build a central ontology and (ii) greater effort is invested at requirement gathering stage, that is, consensus building. However, the HL7 standard itself is an agreement between HL7 users. Therefore, the priority shifts from requirement gathering to the reuse of HL7 resources published in various formats and arrangements of global and local ontologies.

In addition to building global and local ontologies, the next issue is to resolve ontological heterogeneity using ontology alignment methods. In this paper we propose the Plug and Play Electronic Patient Records (PPEPR) methodology for ontologising the HL7 standard. The PPEPR methodology is based on our experiences in developing and managing the PPEPR framework [16]. The structure of this paper is as follows: first we identify three HL7-specific features for ontology building and introduce the PPEPR methodology. We describe semi-automatic approaches for ontologising HL7 resources. Finally, we introduce our approach for

<sup>4</sup> <http://www.hl7.org/>

<sup>5</sup> <http://www.openehr.org/>

<sup>6</sup> <http://www.cen.eu/>

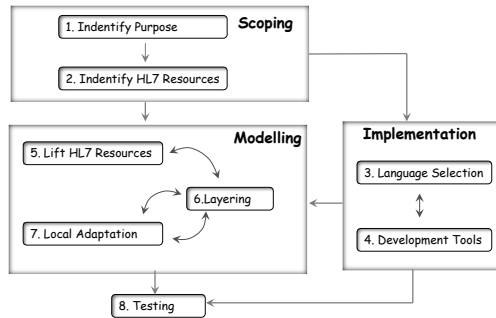
aligning HL7 ontologies from different versions as well as different local information models. The PPEPR methodology is focused around using and arranging global and local ontologies.

## 2 Ontology Building Methodology

The proposed methodology is scenario-based where ontology engineers first identify an application scenario. All resources and entities identified within the application scenario set the guidelines for further development phases. This scenario-based approach makes the proposed methodology application-dependent. While developing the PPEPR methodology, we considered it important to provide guidance to ontology engineers, healthcare engineers and domain experts as fine-grained as possible to make the sequence of development steps concrete and reproducible. We have identified three HL7-specific features/properties lacking within existing ontology building methodologies: (i) reusability of existing non-ontological knowledge sources such as XML schemas specifications, (ii) layering of ontological knowledge bases, for example, V3 has a globally consistent conceptual model as well as locally designed messaging schemas that enable exchange of clinical messages, (iii) adaptation of local knowledge sources with the upper or global conceptual model.

### 2.1 The PPEPR Methodology

The PPEPR methodology is grounded on existing methodologies and domain experiences. We took inspiration where ever possible from existing methodologies (Enterprise Ontology [18], METHONTOLOGY [6], On-To-Knowledge [17], and DILIGENT [13]).



**Fig. 1.** PPEPR Ontology Building Methodology

Additionally, we introduce HL7-specific guidelines for carrying out development phases and steps. The EU project RIDE<sup>7</sup> consortium have suggested a semantic-based roadmap for the interoperability of different healthcare standards and systems. From our RIDE experiences, we obtained a preliminary set

<sup>7</sup> <http://www.srdc.metu.edu.tr/webpage/projects/ride/>

of requirements and guidelines for the use of semantics and making healthcare applications interoperable. Fig. 1 presents the PPEPR methodology which consists of four phases: (i) the **scoping phase** establishes the purpose of ontology building and identifies resources that can support the ontology building process; (ii) the **implementation phase** evaluates Semantic Web languages and supporting tools that can fulfill requirements of the scoping phase; (iii) the **modelling phase** provides detailed guidelines for constructing ontologies; finally (iv) the **testing phase** ensures the consistency and correctness of ontologies with respect to previous phases and requirements. Development steps are allocated to each phase, which indicates the order in which the activities should be performed. In this paper, we primarily focus on the modelling phase responsible for ontologising HL7 specifications, which we present along with the respective development steps.

## 2.2 The Modelling Phase

The modelling phase starts with the task of lifting HL7 resources and ends with the local adaptation of ontologies. The overall goal of this phase is to build HL7 ontologies that have greater conformance with the standard as well as the deployed applications.

**Lifting HL7 Resources** The Lifting step takes as **input:** HL7 resources in XML; and produces as **output:** HL7 global ontologies (by lifting *coreSchemas*<sup>8</sup>) and message ontologies (by lifting local message schemas).

We have categorised HL7 resources in two types, (i) the conceptual model: artefacts that commonly apply to underlying applications and (ii) message schemas: generated from local applications. The V3 conceptual model consists of vocabularies and datatype definitions available as UML and XML Schema specifications, while V2 has datatype, segment and field specifications published as XML Schemas. Ontology engineers can take two possible routes to ontologise V3:

**Route 1:** V3 UML models are stored as Model Interchange Format (MIF)<sup>9</sup> files and HL7 tools can export UML models into message schemas.

- **Conceptual model:** one possibility is to convert MIF repositories to corresponding ontologies. UML and ontology languages (such as OWL [8] or WSML [2]) have similar notions (*e.g.*, Class, subclass, Attribute/Property) for knowledge modelling. However, one major hurdle for this conversion is the presence of extensive system level information (*e.g.*, when model was created, who created it, machine configuration) within MIF files. It is hard to filter out a conceptual hierarchy from MIF files. Attempts to convert HL7 MIF to OWL ontologies were reported as a tedious task [11]. To provide greater automation to this task, MIF to OWL conversion can be defined using XSLT-based transformation rules.

<sup>8</sup> *coreSchemas* are a set of primary schemas containing HL7 message building blocks

<sup>9</sup> <http://wiki.hl7.org/index.php?title=MIF>

- **Message schema:** once the conceptual model is converted, message schemas and instances can be lifted from XML(S) $\leftrightarrow$ Ontology using again XSLT rules.

**Route 2:** the conceptual models of both versions are available as XML Schemas (also known as HL7 *coreSchemas*). These *coreSchemas* are arranged in special ways to represent UML notations (Class/Subclass/Attribute) within the XML structure.

- **Conceptual model:** *coreSchemas* represent conceptual parts of the HL7 messaging framework. Specialised XSLT rules can be defined to lift schema definitions.
- **Message schema:** similar transformation rules can be applied to convert message schemas.

We have taken Route 2 where one set of transformation rules (*i.e.*, XML Schema $\leftrightarrow$  Ontology) is used to lift both the conceptual and the message schemas without the intermediate transformation to UML/MIF. In the case of Route 1, two sets of transformation rules (*i.e.*, XML Schema  $\leftrightarrow$  Ontology and MIF  $\leftrightarrow$  Ontology) are required. However, the advantage of Route 1 is the similarity between UML and ontology languages, which could reduce overall transformation rules. Fig. 2 presents transformation rules between XML Schema  $\leftrightarrow$  OWL  $\leftrightarrow$  WSMML. For the PPEPR framework we have developed ontologies in two ontological formats (OWL and WSMML). Considering the space limitation in this paper, examples are presented using OWL Manchester syntax<sup>10</sup> only. In Fig. 2, italicised schema elements are HL7 specific and other rules can be applied to general XML Schema $\leftrightarrow$ OWL $\leftrightarrow$ WSMML conversions.

XML Schema	WSMML	OWL
<code>element attribute</code>	<code>attribute</code>	<code>ObjectProperty DataProperty</code>
<code>element@substitutionGroup</code>	<code>subAttributeOf</code>	<code>SubPropertyOf</code>
<code>element@type</code>	<code>Range</code>	<code>Range</code>
<code>complexType group attributeGroup</code>	<code>concept</code>	<code>Class</code>
<code>extension@base restriction@base union@memberTypes attribute@classCode type=Class</code>	<code>subConceptOf</code>	<code>SubClassOf</code>
<code>@maxOccurs @minOccurs</code>	<code>maxCard. minCard.</code>	<code>max min</code>
<code>Annotation@ppinfo hl7:LongName hl7:Type</code>	<code>annotations</code>	<code>Annotations@label comment</code>

Fig. 2. XML Schema $\leftrightarrow$ OWL $\leftrightarrow$ WSMML transformation rules

Fig. 3 shows the XML Schema type *AD* from V3 (part of the conceptual model). The *AD* datatype represents mailing, home or office addresses of health-care entities (*e.g.*, patient, physician).

*AD* has a sequence of address parts (*adxp*), such as street or post office box, city, postal code, etc. Similarly, Fig. 3 shows that *AD* extends the base

<sup>10</sup> <http://www.w3.org/TR/owl2-manchester-syntax/>

type *ANY* (HL7 *ANY* is similar to *owl:Thing*) and has sequence of elements describing each address part. *AD* is the most commonly used element in clinical message exchanges.

<pre>&lt;xs:complexType name="AD" mixed="true" &gt; &lt;xs:complexContent&gt; &lt;xs:extension base="ANY" &gt; &lt;xs:sequence&gt; &lt;xs:element name="country" type="adxp.country"/&gt; &lt;xs:element name="state" type="adxp.state"/&gt; &lt;xs:element name="city" type="adxp.city"/&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;</pre>	<pre>&lt;xsd:complexType name="AD" &gt; &lt;xsd:sequence&gt; &lt;xsd:element name="AD.1" type="AD.1.CONTENT"/&gt; &lt;xsd:element name="AD.2" type="AD.2.CONTENT"/&gt; &lt;xsd:element name="AD.3" type="AD.3.CONTENT"/&gt; &lt;/xsd:sequence&gt; &lt;/xsd:complexType&gt;</pre>
--	--

Fig. 3. V3 *AD* Datatype (XSD)Fig. 4. V2 *AD* Datatype (XSD)

Fig. 5 shows an ontology snippet transformed from the example shown in Fig. 3. The transformation rules described in Fig. 2 define XSD base extensions as *subClass/subConcept* relations and attributes as *objectProperty*. According to these rules, we achieve transformations as “*concept AD subConceptOf Datatype#ANY*” and “*ObjectProperty: country domain: AD range: Adxp.country*”.

<pre>Class: AD SubClassOf: ANY ObjectProperty: country Domain: AD Range: Adxp.country ObjectProperty: state Domain: AD Range: Adxp.state ObjectProperty: city Domain: AD Range: Adxp.city</pre>	<pre>Class: AD ObjectProperty: AD.1 Domain: AD Range: AD.1.CONTENT ObjectProperty: AD.2 Domain: AD Range: AD.2.CONTENT ObjectProperty: AD.3 Domain: AD Range: AD.3.CONTENT</pre>
---	--

Fig. 5. V3 *AD* Ontology SnippetFig. 6. V2 *AD* Ontology Snippet

Similarly, Fig. 4 and Fig. 6 show the V2 Schema for *AD* datatype and the corresponding ontology. We notice that the properties of both *AD* ontologies from V2 and V3 have different naming schemes (e.g., *city*, *AD.3*). Such heterogeneity requires alignment of datatype ontologies.

**Layering** The Layering step takes as **input**: HL7 global and message ontologies; and produces as **output**: layered HL7 global and local ontologies. The PPEPR methodology allows for multiple global ontologies so that local ontologies do not have to commit to one overarching standardised ontology. Local ontologies generally originate from different groups (hospitals, countries) where each group commits to different standards/policies. Alignments at different levels (global and local) enable agreement at the upper layer and the resolution of local differences at a separate level.

The layering task arranges ontologies into global (shared by all HL7 users) and local (used within a particular domain) spaces. Global ontologies for both versions can be created from *coreSchemas*. These upper layers are universally applicable to all deployed HL7 applications. To create local ontologies, we suggest using the XML Schemas of locally created clinical messages. The same transformation rules described in Fig. 2 can be used for converting message schemas to message ontologies. Fig. 7 presents a high level view of arrangement of global and local ontologies. In Fig. 7, a circled plus symbol means that the message ontologies are simply merged to form a local ontology, whereas a circled cross symbol represents the task of ontology alignments where similar concepts and relationships are matched to resolve differences.

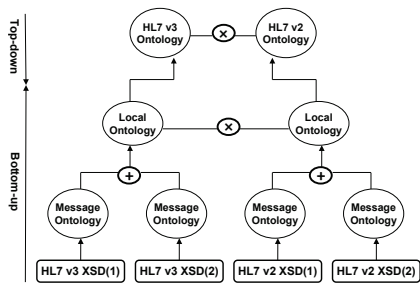


Fig. 7. PPEPR Methodology: Layering



Fig. 8. Lab observation from xsd(1)

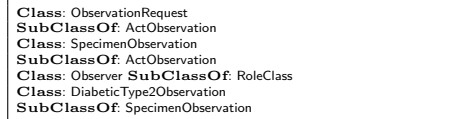


Fig. 9. Lab observation from xsd(2)

The PPEPR methodology aims to reduce the bilateral mappings between local ontologies by using global ontologies: considering the practicalities of HL7 applications and the flexibility that HL7 allows to include/create local vocabularies and codes, it is impossible to completely avoid bi-lateral mappings between local ontologies. Local ontologies are created from message ontologies in a bottom-up way whereas global ontologies are created in a top-down fashion. Two conceptual ambiguities exist between message ontologies: (i) semantically similar concepts are named differently (ii) corresponding concepts are represented at different structural levels. These ambiguities arise because local systems have the flexibility and choices to design clinical messages.

Fig. 8 and Fig. 9 show snippets of message ontologies transformed from *Lab Observation* schemas of two V3 applications. For this paper, we used example message schemas from HL7 ballots (Jan 2007<sup>11</sup> and Jan 2011<sup>12</sup>) and schemas provided by the PPEPR project partners. The different time-line (Jan 2007 and Jan 2011) has been chosen to show heterogeneity between two standard schemas. For example, one difference is the use of codes for *complexType* names (transformed to classes `ObservationOrder.POOB.MT210000UV`, `Observer.POOB.MT210000UV`). Codes describe a unique identification of schemas within the HL7 messaging space and they carry a specific meaning. For example, `ObservationOrder` is a label for the `POOB.MT210000UV` message where `POOB` indicates Observation Order, `MT` stands for Message Type, the number shows the order of message exchanges, and `UV` is used to denote universal scope. There are country-specific codes as well, in that case, `UV` changes to `GB` for Great Britain. Our experience suggests that developers use local conventions to name *complexType*, e.g., some use only labels, others use codes, and many use a combination of both like `ObservationOrder.POOB.MT210000UV`. Another difference between the two schemas is the hierarchy of observation classes. Fig. 8 describes `HemoglobinObservation.POOB.MT210000UV` as a direct specialisation of `Ac-`

<sup>11</sup> [http://www.hl7.org/v3ballot/html/welcome/downloads/v3ballot\\_schemasexamples\\\_2007JAN.zip](http://www.hl7.org/v3ballot/html/welcome/downloads/v3ballot_schemasexamples\_2007JAN.zip)

<sup>12</sup> [http://www.hl7.org/v3ballot2011jan/html/welcome/downloads/v3ballot\\_schemasexamples\\\_2011JAN.zip](http://www.hl7.org/v3ballot2011jan/html/welcome/downloads/v3ballot_schemasexamples\_2011JAN.zip)

tObservation, while in Fig. 9 SpecimenObservation is a subclass of ActObservation. Diabetic tests require hemoglobin test, however, DiabeticType2Observation is a special test for Diabetic Type 2 patients. It is up to local practices and clinicians in their local contexts (*e.g.*, hospital, lab, country) how they prefer to model any observation.

<p><b>Class:</b> ObservationRequest<sub>s</sub> <b>EquivalentTo:</b> ObservationOrder. POOB.MT210000UV<sub>t</sub></p> <p><b>Class:</b> Observer<sub>s</sub> <b>EquivalentTo:</b> Observer.POOB.MT210000UV<sub>t</sub></p> <p><b>Class:</b> DiabeticType2Observation<sub>s</sub> <b>SubClassOf:</b> HemoglobinObservation.POOB.MT210000UV<sub>t</sub></p>	<p><b>Class:</b> ObservationRequest <b>SubClassOf:</b> ActObservation</p> <p><b>Class:</b> Observer <b>SubClassOf:</b> RoleClass</p> <p><b>Class:</b> SpecimenObservation <b>SubClassOf:</b> ActObservation</p> <p><b>Class:</b> HemoglobinObservation.POOB.MT210000UV <b>SubClassOf:</b> ActObservation</p> <p><b>Class:</b> DiabeticType2Observation <b>SubClassOf:</b> SpecimenObservation HemoglobinObservation.POOB.MT210000UV</p>
---	---

**Fig. 10.** Alignment of Fig. 8 and Fig. 9

**Fig. 11.** Merged Local Ontology

A correspondence describing the relationship between two concepts (DiabeticType2Observation, HemoglobinObservation.POOB.MT210000UV) is required to exchange messages. To deal with these two conceptual ambiguities, one option is to provide alignments between message ontologies. However, the maintenance of alignments would be a significant burden. Considering the problem of several small message ontologies and the maintenance of their alignments, we propose a two-step process: (i) merge all message ontologies and alignments into a single local ontology for a domain, which means that each domain is represented by a local ontology; (ii) use alignments to merge semantically similar concepts and properties into a single concept or property. This way, ontologies could be maintained at a reasonable size. This is possible as local systems have more control over their message ontologies than on the global ontology. An additional task is to update references within message instances according to the newly created local ontology.

Fig. 10 shows an alignment between concepts from the message ontologies. The merge can be achieved using general merging tool such as Prompt Suite [10]. When merging is applied to the message ontologies and their alignments, then Fig. 11 shows the output local ontology.

We observe in Fig. 11 that concepts ObservationRequest and ObservationOrder.POOB.MT210000UV are merged into a single concept ObservationRequest. Similarly, concepts Observer and Observer.POOB.MT210000UV are merged to the Observer concept. In case of structural differences, concept DiabeticType2Observation specialises from concepts SpecimenObservation and Hemoglobin Test, such that DiabeticType2Observation is represented at the appropriate level of hierarchy.

**Local adaptation** The final Local adaptation step takes as **input:** HL7 global and local ontologies; and has as **output:** Extended global and local ontologies meeting local requirements.

The notion of local adaptation was first proposed by DILIGENT [13]. DILIGENT local users adapt the global ontology by introducing local changes and a central board controls the contents of the global and local ontologies. In fact, the HL7 standard works on a similar paradigm where country-specific changes are maintained by respective HL7 groups. A central harmonisation committee

(as in the DILIGENT central board) is responsible for introducing local changes within the V2 and V3 conceptual models. Along similar lines, PPEPR global and local ontologies originate from multiple sources, from HL7 standard to locally deployed XML Schemas. The local adaptation step is motivated by normal practices where HL7 applications diverge from the standard guidelines by modifying local XML Schemas and by introducing local vocabularies. The local adaptation phase ensures that: (i) local ontologies are generalised enough to resemble the concepts defined in the global ontology, and (ii) global ontologies are specialised enough to resemble the concepts defined in the local ontologies. Local concepts generally represent the consensus of a local group. Consequently, their understanding is limited to the local environment.

```

Class: ActObservation SubClassOf: Act
Class: ActSpecimenObservation SubClassOf: ActObservation
Class: ActGenomicObservation SubClassOf: ActObservation
Class: ActClinicalTrial SubClassOf: ActObservation
Class: ActCondition SubClassOf: ActObservation

```

**Fig. 12.** Snippet of Global Ontology (V3)

Fig. 12 shows a snippet of the global ontology (V3). The axioms describe top level concepts of the local ontologies shown in Fig. 8 and Fig. 9. For example, “ActObservation” class represents all types of clinical observations possible within a clinical environment and “ActGenomicObservation”, “ActClinicalTrial”, “ActSpecimenObservation”, etc. are specialisations of “ActObservation”. Each of these specialisations requires “lab test order” in different clinical scenarios. The local concept “ObservationRequest” (see Fig. 9) is eligible to inherit from many of these specialisations. In addition to the local adaptation proposed by DILIGENT, the PPEPR methodology suggests three approaches for the adaptation of the local concept “ObservationRequest”. These three approaches are refinements of approaches proposed in the Enterprise Ontology [18]:

*Top-Down* extend the global ontology with more specialised concepts that resemble the concepts defined in local ontologies. For example, each of the concepts “ActGenomicObservation” and “ActClinicalTrial” could further be extended with a lab specific concept like “ActClinicalTrialLabObservation” or “ActGenomicLabObservation”, to represent that all lab tests specific to clinical trials are denoted by “ActClinicalTrialLabObservation”.

*Bottom-Up* extend the local ontology with more generalised concepts that resemble the concepts defined in the global ontology. For example, a super class “ActClinicalTrialLabObservation” for the local concept “ObservationRequest” can be added to the local ontology, so appropriate inheritance like “ActClinicalTrialLabObservation” subClassOf “ActClinicalTrial” could be established in the local ontology. For instance, DiabeticType2Observation in Fig. 9 is a local concept and inherited from super class “ActObservation” via “SpecimenObservation”.

*Middle-Out* concepts in the global ontology (“ActGenomicObservation” or “ActClinicalTrial”) are defined at a higher level of abstraction, which allows local

concepts to inherit from all of them. Therefore, instead of specialising or generalising global or local concepts, another approach is to add a specialised class for “Lab Test” on the same level as the above two concepts. For example, Alpha fetoprotein (AFP) is an observation for detecting liver cancer, and V3 (Jan 2011 Ballot) does not include any specialised classes for cancer diseases. It may be appropriate to add “CancerObservation” on a similar level to “ActGenomicObservation” or “ActClinicalTrial”, and then extending “CancerObservation” with an “AlphaFetoproteinObservation” concept.

These three approaches could be applied independently or in combinations depending on requirements from different clinical scenarios. Enterprise Ontology suggested the middle-out approach. However, considering the heterogeneities of different clinical scenarios, we intentionally avoid any “fit-for-all” suggestion.

### 3 Aligning HL7 Ontologies

The integration and alignment of ontologies is a well-investigated fundamental problem in the development of Semantic Web techniques. Obviously, just using ontologies instead of XML, does not defeat heterogeneity: it just raises heterogeneity problems to a higher level with a hope for easier integration. Analogous to schema matching and mapping techniques, ontology matching methods have been proposed to deal with ontological heterogeneity. Ontology matching consists of finding the correspondences (*e.g.*, equivalence, subsumption) between elements of ontologies (classes, properties). This is an important task because it helps establishing actual interoperability. It is also a difficult task because (i) independently built ontologies can vary significantly in the terminology they use and the way they model the same entities; and (ii) domain ontologies focus on a particular domain and use terms in a sense that is relevant to the domain and which are not related to similar concepts in other domains.

Most research in the ontology matching [5] area has focused on (semi-) automatically aligning ontologies using approaches based on combinations of syntactic similarity, graph similarity, and using a third upper level ontology. In many cases, linguistic and structural matchings are used in conjunction with an upper level ontology (*e.g.*, CYC<sup>13</sup>) as a common layer for two ontologies to be matched. The (semi-)automated approaches of linguistic and structural matchings, typically give an incomplete or incorrect set of correspondences between terms. A human must align the remaining terms and check the machine-built alignments to truly complete the alignment process. Our work places emphasis on exploiting alignment methods to provide valuable insight to the alignment process and improve accuracy. In particular, we want to investigate how “fit” generic ontology alignment methods are for our relatively straightforwardly created HL7 V2 and V3 ontologies: as described in Fig. 8 and Fig. 9, *Lab Observation* concepts are modelled differently within V3. Similarly, Fig. 5 and Fig. 6 show differences in the *AD* datatype of V2 and V3. Fig. 13 shows XSD for the range class

<sup>13</sup> <http://www.cyc.com/>

(AD.3.CONTENT) of the AD.3 *objectProperty*. The class AD.3.CONTENT is *equivalent* to the class Adxp.city of Fig. 5.

---

```

<xsd:complexType name="AD.3.CONTENT">
  <xsd:annotation>
    <xsd:appinfo>
      <hl7:Type>ST</hl7:Type>
      <hl7:LongName>City</hl7:LongName>
    </xsd:appinfo>
  </xsd:annotation>
  ...

```

---

**Fig. 13.** V2 XSD Annotations

We notice that V3 class names (*e.g.*, Adxp.city, Adxp.country, ObservationRequest) are self-descriptive, while V2 class names are coded (*e.g.*, AD.1, AD.1.CONTENT) with descriptions attached as annotations (*e.g.*, hl7:Type or hl7:LongName of Fig. 13). We evaluated three ontology matching tools, namely, Anchor-Prompt [10], Falcon-AO [9], and H-Match [3]. We have selected these three matching tools because we had experience with them and they offer stable and easy testing environments. Matching evaluations are conducted in two categories: (i) finding matches within the same version (*e.g.*, Fig. 8 and Fig. 9 of V3), (ii) finding matches between two versions (*e.g.*, Fig. 5 and Fig. 6 of V3 and V2). Evaluation tests are conducted on full ontologies as well as subparts of ontologies to analyse performance vs. result output.

## 4 Ontology Matching Evaluation

**Anchor-PROMPT** is an ontology mapper that attempts to automatically find semantically similar terms between two ontologies. Anchor-PROMPT uses a graph-based approach where an ontology is taken as a directed labelled graph. Anchor-PROMPT takes as input a set of pairs of related terms, called “anchors”, from the source ontologies. These anchors are identified before triggering the matching task; either the user enters them or the system automatically generates them. Based on this set of anchors, Anchor-PROMPT produces a set of pairs of semantically close terms.

**Falcon-AO** is an ontology matching system for finding, aligning and learning ontologies. Falcon-AO is a similarity-based generic ontology mapping system. It consists of two elementary matchers: one is a matcher based on linguistic matching for ontologies, called Linguistic Matching for Ontologies (LMO); the other is a matcher based on graph matching for ontologies, called Graph Matching for Ontologies (GMO). GMO takes the alignments generated by LMO as external input and outputs additional alignments. Linguistic similarity between two entities relies on their names, labels, comments and other descriptions.

**H-Match** performs ontology matching at different levels of depth by deploying four different matching models, namely, *surface*, *shallow*, *deep*, and *intensive*. In H-Match a threshold-based mechanism is enforced to set the minimum level of semantic affinity required to consider two concepts as matching concepts. The linguistic affinity function of H-Match provides a measure of similarity between two ontology concepts computed on the basis of their linguistic features (*e.g.*,

concept names). For the linguistic affinity evaluation, H-Match relies on a thesaurus of terms and terminological relationships automatically extracted from the *WordNet*<sup>14</sup> lexical system. The contextual affinity function of H-Match provides a measure of similarity by taking into account the contextual features of the ontology concepts. The context of a concept can include properties, relations with other concepts, and property values.

Fig. 14 shows matching results of V2 and V3 ontologies. The second column describes matching between V3 ontologies and the fourth column shows the accuracy of V2 and V3 ontology matching. In the second column, matching ontologies (*i.e.*, local message ontologies for V3) include “common concepts” inherited from the global reference ontology. In the fourth column, global ontologies of different versions (V2 and V3) are matched, which means “common concepts” are irrelevant and concepts are different in terms of their names or structure. The percentage denotes a *precision* ( $p$ ) and a *recall* ( $r$ ) measurements for evaluating accuracy of matches discovered. Precision indicates the “correctness” and recall measures the “completeness” of matching results. Both parameters are measured against the reference alignment  $R$  with alignment  $A$  returned by the matching tools, where  $p = \frac{|R \cap A|}{|A|}$  and  $r = \frac{|R \cap A|}{|R|}$ . In our case the two input ontologies with 310 and 290 concepts have 200 prospective matching concepts (*i.e.*, reference alignment  $R$ ). The third and last columns show the size (*i.e.*, total number of concepts excluding properties) of ontologies and their subparts.

Method/Tool	HL7 (V3 -V3) precision-recall	Ontology/Subpart HL7V3	HL7 (V2-V3) precision-recall	Ontology/Subpart HL7 (V2-V3)	Threshold Value
Anchor-Prompt	90%-30%	310/50	90%-10%	290/50-310/50	Varies
Falcon-AO	70%-50%	310/50	30%-20%	290/50-310/50	0.01
HMatch	80%-90%	310/50	40%-20%	290/50-310/50	0.5
SPARQL Recipes	80%-90%	310/50	50%-60%	290/50-310/50	

Fig. 14. Matching Evaluation

We observe in Fig. 14 that matching results are significantly different between second and fourth columns. For example, Anchor-Prompt has higher  $p=90\%$  and lower  $r = 30\%$ , which indicates that the completeness of alignments is lower. Precision is higher due to obvious matches (the common concepts) from the reference ontology and negligible *false positives* in the alignment between local concepts. Similarly, in the fourth column, Anchor-Prompt recall value is much lower as obvious matches between V2 and V3 are negligible. Precision is higher because of few exact matches between datatypes (*e.g.*, AD of V2 and V3) concepts. The fifth column shows the threshold values for deciding concept equivalent: Anchor-Prompt defines it at run-time, depending on path length and set of anchors; Falcon-AO value (0.01) is fixed by the tool itself; H-Match allows one to select a value between 0 and 1, we opted the default value (0.5). Falcon-AO and H-Match have significantly improved recall for the second column (*i.e.*, 50% and 90%), however, precision falls because of higher *false positives*. H-Match precision and recall ratios are best because it has different levels of matching and

<sup>14</sup> <http://wordnet.princeton.edu/>

we noted the best matches (*intensive* matching). However, *intensive* matching shows greater delays with ontologies of size greater than 50 concepts. This is the reason we have used smaller subparts of ontologies to determine the overall performance and matching results. Unfortunately, the recall measurement for the fourth column is quite low. This is due to the naming scheme (coded for V2 vs. self-descriptive names V3 ontologies). For example, AD of V3 matches with AD, AD.1, AD.1.CONTENT, AD.2.CONTENT concepts of V2, which makes *false positives* within the alignment higher, thus, precision lower.

## 5 SPARQL Recipes

Instead of relying on automated alignments, a simple query-based method for expressing customised and complex alignments between ontologies has been proposed in [14, 4] using the SPARQL query language. We have employed this approach specifically to manually match V2 and V3 ontologies.

```
CONSTRUCT { ?v3 owl:equivalentClass ?v2 } WHERE { ?v3 rdf:type owl:Class . ?v2 rdf:type owl:Class .
?v2 rdfs:label ?LongName . { FILTER regex(str(?v3), str(?LongName), "i") }
```

**Fig. 15.** SPARQL Recipe for Ontology Matching

Surprisingly, a single, generic SPARQL query that matches concept names of the V3 ontology against *LongName* (the label annotations) of the V2 ontology outperforms all automatic mapping attempts. We notice in Fig. 13 that the annotation of *hl7:LongName* (“City”) is a substring of the concept named *Adxp.city* from Fig. 5; this mapping from the annotation in onto the concept name by substrings is a recurring pattern in mappings between V2 and V3. Similarly, HL7 ontologies contain several other “simple patterns” that can guide us in determining correspondences between ontological elements. A domain expert is required to analyse and identify such “simple patterns”. Fig. 15 shows a “simple pattern” created as a “SPARQL recipe”:

By employing this simple “SPARQL recipe”, we observed significant successful matching results for classes that share similarity but are described and modelled differently. SPARQL recipes matching results are similar to H-Match (*precision* 80%, *recall* 90%) for V3 ontologies (second column of Fig. 14) and significantly improved *precision (p)* 50% and *recall(r)* 60% for V2 and V3 ontologies. However, a major limitation of “recipes” is lack of standard matching measures, like threshold, similarity coefficient, etc. Recipes can be created by domain experts after analysing similarities between ontological elements in HL7. As opposed to alignment algorithms that usually just search for simpler alignments, such recipes can express complex correspondences between ontology instances, involving built-in function calls, etc. For example, it allows one to express complex correspondences such as concatenating attributes (*e.g.*, first/last names, dates) and express complex interactions between knowledge of various sources. Similarly, complex correspondences could be expressed in the Rule Interchange Format (RIF)<sup>15</sup>, which offers a rich set of built-in functions (*e.g.*, string manip-

<sup>15</sup> <http://www.w3.org/TR/rif-core/>

ulations), as well as a formal semantics for interoperating with RDF and OWL knowledge bases. RIF has been recently approved as a W3C standard recommendation and we consider it as a prominent candidate to express alignments over ontologies.

## 6 Related Works

The first release of the ANSI normative V3<sup>16</sup> was published in 2003. Since then, ontology engineers have taken initiatives to ontologise V3. To the best of our knowledge, the EU Project Artemis [1] is the only significant contribution for ontologising both V3 and V2. In 2003, Bhavana Orgun has developed V3 ontology using the RDFS language [12]. The main contribution of Orgun’s work is to identify important sets of ontological concepts and properties from the V3 artefacts. In 2008, Helen Chan from W3C HCLS IG<sup>17</sup> published V3 ontology. Chan’s work has improved Orgun’s ontology by covering a broader range of artefacts from the V3. In 2009, Alan Rector [15] proposed to align V3 artefacts with other standard medical vocabularies such as SNOMED<sup>18</sup>. The focus of this work is different from the issue of interoperability between two versions of the HL7 standard and the presence of local applications.

All the approaches mentioned above focus on ontologising upper conceptual models of HL7 standard. None of them consider local applications and related issues. The PPEPR methodology has two development steps that deal with the problem of layered knowledge spaces and how local resources could be adapted with an upper conceptual model. Similarly, none of the related work aimed for (semi-)automatic alignment of HL7 ontologies. Above all, the works mentioned above lack detailed methodology for ontologising HL7 applications.

## 7 Conclusion

We have identified three HL7-specific ontology building features (reusability of non-ontological resources, ontology layering, and local adaptation) missing from traditional ontology building methodologies. We propose the PPEPR ontology building methodology for the HL7 standard. The PPEPR methodology extends, refines, and inherits from existing methodologies. We address a key requirement, (semi-)automatic ontology alignment, to resolve ontological heterogeneity. We have tried three ontology matching systems for alignments between different ontologised versions of the HL7, but came to the conclusion that simple manual mapping “recipes” worked better for us. In the future, we plan to extend the evaluation with other matching tools, e.g. from the “anatomy track” of the Ontology Alignment Evaluation Initiative (OAEI)<sup>19</sup>. Our future work also is to extend the

<sup>16</sup> <http://www.hl7.org/implement/standards/rim.cfm>

<sup>17</sup> <http://www.w3.org/wiki/HCLS/ClinicalObservationsInteroperability/HL7CDA2OWL.html>

<sup>18</sup> <http://www.ihtsdo.org/snomed-ct/>

<sup>19</sup> <http://oaei.ontologymatching.org/2010/results/anatomy/index.html>

PPEPR methodology by introducing mechanisms that may allow interoperation with other prominent healthcare standards and medical vocabularies.

## References

1. Bicer, V., Laleci, G.B., Dogac, A., Kabak, Y.: Artemis Message Exchange Framework: Semantic Interoperability of Exchanged Messages in the Healthcare Domain. SIGMOD Record (ACM Special Interest Group on Management of Data) (2005)
2. de Bruijn, J., Fensel, D., Keller, U., Kifer, M., Kruppenacher, R., Lausen, H., Polleres, A., Predoiu, L.: Web Service Modeling Language (WSML), w3C member submission, June 2005
3. Castano, S., Ferrara, A., Montanelli, S.: Matching Ontologies in Open Networked Systems: Techniques and Applications. *J. on Data Semantics* 3870(V):25–63 (2006)
4. Euzenat, J., Polleres, A., Scharffe, F.: SPARQL Extensions for Processing Alignments. *IEEE Intelligent Systems* 23(6), 82–84 (Nov 2008)
5. Euzenat, J., Shvaiko, P.: *Ontology Matching*. Springer-Verlag, Heidelberg (DE) (2007)
6. Fernández-López, M., Gómez-Pérez, A., Juristo, N.: Methontology: from ontological art towards ontological engineering. In: *Proc. Symposium on Ontological Engineering of AAAI* (1997)
7. Fernández-López, M., Gómez-Pérez, A.: Overview and analysis of methodologies for building ontologies. *Knowl. Eng. Rev.* (2002)
8. Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P., Rudolph, S.: *OWL 2 Web Ontology Language Primer W3C Recommendation*. Tech. rep., World Wide Web Consortium (W3C) (Oct 2009)
9. Hu, W., Qu, Y.: Falcon-AO: A Practical Ontology Matching System. *Journal of Web Semantics* 6(3), 237–239 (2008)
10. Noy, N.F., Musen, M.A.: The PROMPT suite: interactive tools for ontology merging and mapping. *Int'l Journal of Human-Computer Studies* 59(6), 983–1024 (2003)
11. Oemig, F., Blobel, B.: An Ontology Architecture for HL7 V3: Pitfalls and Outcomes. In: *World Congress on Medical Physics and Biomedical Engineering* (2009)
12. Orgun, B.: *INTEROPERABILITY IN HETEROGENEOUS MEDICAL INFORMATION SYSTEMS USING SMART MOBILE AGENTS AND HL7 (EMAGS)*. Master's thesis, Macquarie University, Australia (2003)
13. Pinto, H.S., Staab, S., Tempich, C.: DILIGENT: Towards a fine-grained methodology for DIstributed, Loosely-controlled and evolvInG Engineering of oNTologies. In: *16th European Conf. on AI (ECAI)*. pp. 393–397. IOS Press (2004)
14. Polleres, A., Scharffe, F., Schindlauer, R.: SPARQL++ for mapping between RDF vocabularies. In: *ODBASE*. pp. 878–896. OTM'07, Springer (2007)
15. Rector, A.L., Qamar, R., Marley, T.: Binding ontologies and coding systems to electronic health records and messages. *Applied Ontologies* 4(1), 51–69 (2009)
16. Sahay, R., Akhtar, W., Fox, R.: PPEPR: Plug and Play Electronic Patient Records. In: *Proceedings of the 23rd Annual ACM Symposium on Applied Computing, the Semantic Web and Applications (SWA) track*. ACM, Fortaleza, Brazil (3 2008)
17. Staab, S., Studer, R., Schnurr, H.P., Sure, Y.: Knowledge Processes and Ontologies. *IEEE Intelligent Systems* 16, 26–34 (January 2001)
18. Uschold, M.: Building Ontologies: Towards a Unified Methodology. In: *16th Annual Conf. of the British Computer Society Specialist Group on Expert Systems*. pp. 16–18 (1996)