Validating EHR clinical models using ontology patterns

Catalina Martínez-Costa

Institute for Medical Informatics, Statistics and Documentation, Medical University of Graz, Austria

Auenbruggerplatz 2, 8036 Graz Österreich

catalina.martinez@medunigraz.at

Stefan Schulz

Institute for Medical Informatics, Statistics and Documentation, Medical University of Graz, Austria

Auenbruggerplatz 2, 8036 Graz Österreich

stefan.schulz@medunigraz.at

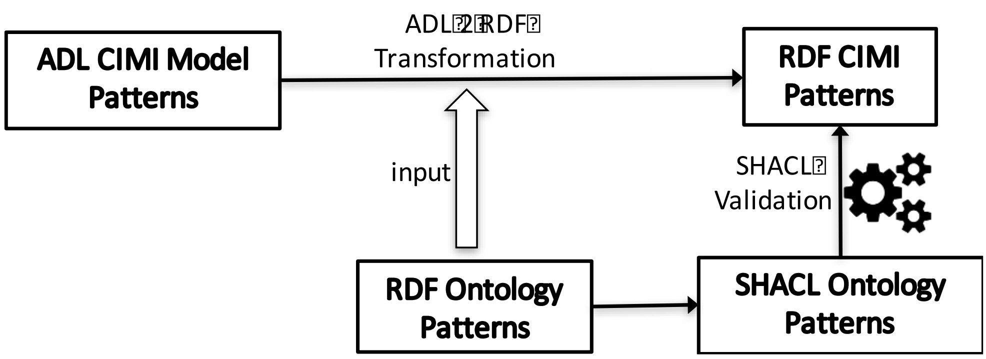
***Corresponding author:** catalina.martinez@medunigraz.at; Tel: +43 316 385 17988

Abstract

Modelling design patterns grounded on formal ontologies are a promising solution to solve current semantic interoperability problems among clinical models from existing Electronic Health Record (EHR) representations. Clinical models focus on gathering the information required at some clinical recording scenario being their modelling a modeler's decision not guided by any formal constraint beyond the constraining of a semantically vague information model. We advocate the use of ontology design patterns to make clinical model's semantics explicit and serve as modelling guideline. This paper focuses on demonstrating the use of ontology design patterns to validate existing clinical models using SHACL. Based on one of the most recent EHR modelling initiatives, the Clinical Information Modelling Initiative (CIMI), we demonstrate how ontology patterns detect both modelling and terminology binding errors (i.e. wrongly used concepts from an external terminology) in CIMI clinical models. SHACL is a W3C constraint language for the validation of RDF graphs. It builds on the concept of "Shape", a data description in terms of expected cardinalities, datatypes and other restrictions. SHACL, as opposite to OWL, subscribes to the Closed World Assumption (CWA) and therefore is more suitable for clinical model validation.

We have demonstrated the feasibility of the approach by manually defining the correspondences between six CIMI clinical models represented in RDF and two SHACL ontology design patterns. Using the available Java based SHACL implementation we found at least eleven modelling errors and binding errors within the CIMI models studied. This research demonstrates the usefulness of ontology design patterns not just as modelling guideline tool but also for validation purposes.

Graphical abstract



Keywords

Semantic interoperability; Clinical Models; Ontology Design Patterns; Data Shapes; SHACL

Highlights

- An approach for checking modelling and terminology binding errors in existing clinical models is proposed
- The proposed approach is based on the use of ontology design patterns described using SHACL
- SHACL data shapes are used to represent ontology design patterns
- The feasibility of the approach is demonstrated by applying it to a subset of CIMI clinical models

1. Introduction

Many approaches that claim to represent electronic health record (EHR) content advertise clinical models as a solution to standardize clinical information for particular use cases [0-3]. Such clinical models (CMs) are based on an information model that they constrain (e.g. HL7 RIM [4], openEHR RM [5], ISO 13606 RM [6]).

Current EHR standards provide mainly means to structuring information items in a form that supports use-case specific data entry and data analysis, but – apart from the optional binding of data items to codes from terminologies – they lack an in-depth rooting in formal ontologies, i.e. global models of meaning. Thus, their focus on syntactic constructs only guarantees the correctness of the models at a syntactic level, but does not really promote semantic interoperability across models, because these would require in-depth semantic representations.

Ontologies formally describe properties and relations of types of entities. Domain-independent categories, relations and axioms are typically provided by top-level ontologies [7], whereas the types of things that make up a domain are represented by domain ontologies. In the former one we find categories like *Process*, *Material entity*, *Quality*, etc., whereas in a clinical domain ontology we find classes for *Diabetes mellitus type 1*, *Left index finger*, or *Diclofenac*, ideally covering all the terms used in clinical documentation and reporting. The terms as such are organized by clinical terminologies. Ontologies have at least a minimal terminology component, consisting in a label or preferred term to make them understandable by humans. SNOMED CT [8], in addition, provides term variants and (quasi-)synonyms as possible values for data entry.

In [9-10] we advocated ontology content design patterns (Content ODPs) [11] as a formal guidance for clinical information modelling in the context of heterogeneous EHR representations. The idea of ODPs is that they bridge between a simple, human-understandable foreground representation and a principled ontology-based background representation, with a strong commitment to an upper-level ontology framework. For instance, both expressions "education to prevent skin cancer" or “education prevents skin cancer” could be translated into the triple "*Education*; `prevents`; *Skin cancer*". Such a representation obscures the fundamental semantic difference between two phrases and is nothing more than providing a seemingly plausible but highly simplified rendering of natural language expressions. The ontological and semantic background representation is complex, only understandable by experts, and possibly computationally expensive[1].

In this work we nevertheless strongly recommend such an ontological framework. Aware of the problems, the complexity and the pitfalls of ontology engineering we study the use of ODPs and demonstrate how they are useful to detect modelling errors in existing CMs. Whereas CMs are usually built from a bottom-up perspective, in a use-case driven way, the proposed formal ODPs are committed to a top-down perspective for building CMs, aiming at a maximum of use-case independence and reusability.

The international semantic interoperability initiative CIMI (Clinical Information Modeling Initiative) [3] has taken the first steps to use modeling patterns to design clinical models and have proposed a set of them represented as CMs. Here, we analyze some of them from a semantic perspective by using ODPs. The CIMI patterns under scrutiny represent observation results as outcome of certain request actions. As ontology patterns we re-use some of the top-level patterns created within the European project SemanticHealthNet [12]. Our validation exercise will check whether the modeling of the CIMI patterns is correct taking our ontological framework as basis. We will also verify whether the terminology bindings are correctly assigned to the data items of the CMs.

Ontology patterns are often represented by using the Web Ontology Language OWL [13]. However, OWL is not intended for typical data validation purposes and does not offer a syntax for patterns. For instance, OWL does not support simple Class – relation - Class triples, due to their fuzzy semantics as shown in the above example[2]. Here we propose to represent them by using SHACL [14]. SHACL is a W3C working draft for validating RDF graphs [15] against a set of conditions (e.g. expected cardinalities, datatypes and other restrictions) provided as shapes. Using this specification, we will propose RDF representations for selected CIMI model patterns, representing the ontology patterns as SHACL shapes with the purpose of error detection.

---

[1] *EducationalMeasure*
    and (**hasParticipant** some *Human*)
    and (**hasOutcome** some (*ReductionProcess*
        and (**hasParticipant** some (*Risk* and **isRealizedBy** only *SkinCancer*))))

[2] We ignore the "punning" mechanism of OWL Full, where classes can be treated as individuals. The reason is that it paves the ground to formulate expressions with fuzzy meaning.

Finally, we demonstrate the feasibility to describe ontology patterns using SHACL. It can be shown that these patterns are useful to detect modeling and terminology binding errors within CIMI models. We also discuss the limitations of this work and devise future research opportunities.

Within the paper we use the following formatting conventions: labels and terms are quoted, for OWL classes we use italics, for OWL relations (i.e. object properties) we use bold face, for RDF predicates we use in `Consolas` font and for individuals we use bold face and end names with the "#" symbol followed by a number.

2. Background

2.1 Electronic Health Record Representations

The last three decades have seen numerous attempts to make electronic health records (EHRs) semantically interoperable [1-3]. The complexity of such a goal is reflected in the vast amount of projects dedicated to it [12,16,17]. Sadly, virtually nothing of this has found its way into commercially available EHR systems so far.

These efforts have positioned clinical models (CMs)[3] as key artefacts. CMs structure clinical information by constraining an underlying, more general information model[4] as a means to define clinical data recording scenarios. Examples of such scenarios are a lab report or a request to administer certain medication. Different communities have proposed different information models, as well as different constraints. EHR information models provide a set of abstract entity types, including their properties and relationships, with the goal to provide a framework that hosts information gathered in healthcare scenarios. Examples of abstract entity types are EVALUATION or ACT in openEHR and HL7 information models respectively.

CMs represent what is done, ordered, seen, concluded or hypothesized in the clinical reality. This suggests a top-level distinction between entities in the clinical reality (like disorders, procedures, drugs, body parts) on the one hand and information (data, statements, orders, …)

[3] Also known as archetypes for certain communities such as openEHR and EN 13606
[4] Also named reference model or reference information model

on the other hand. This distinction follows Ogden and Richards' semantic triangle [18] (see Fig.1). Each corner distinguishes between (1) reference, (2) referent and (3) symbol. EHR information models and their parts correspond to the symbol corner of the triangle. Their constituents, i.e. information entities, have as their referents either (i) clinically relevant individuals that are members of classes (normally seen as the extensions of types, universals or concepts) like *Aspirin*, *Skin cancer*, *Chest pain*, *Chest X-ray*, *Left ey*e, or (ii) to the types themselves (e.g. in our example where the reference to skin cancer does not imply that the patient has skin cancer) [19].

Classes, together with human-readable labels and sometimes formal definitions are typically provided by reference terminologies or ontologies[5] (i.e. REFERENCE in Fig. 1) such as SNOMED CT [8] or ICD-11 [20].[6] Formal definitions, textual scope notes and self-explaining labels avoid undesired interpretations, so that all users share the same meaning, when they refer to the same terminology element.

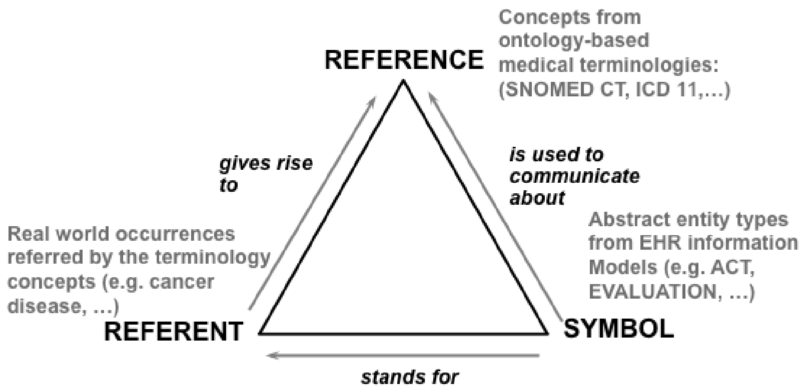


Figure. 1 Ogden and Richards' semantic triangle applied to EHR information models and ontology–based clinical terminologies

CMs focus on gathering the information required at some clinical recording scenario. How many data elements (in the following called nodes) are used to record some information is mainly a designer decision. This leads to the creation of multiple models for gathering the same information (also called isosemantic models by some authors, i.e. heterogeneously structured models but with equivalent meaning). Within CMs, information model entities can be structurally related to representational units (concepts, classes) in terminologies[7]. This relationship is known as terminology binding [21]. Both information models and terminologies are required in order to fully represent electronic patient records. However,

[5] In this paper, (reference) terminology also includes formal ontology
[6] We here exclude interface terminologies, i.e. repositories of human language terms
[7] We refer to ontology based-medical terminologies

there is no agreement about how to create this binding. Moreover, the role of terminologies in CMs is often confused, and medical terminologies are used as information models and vice-versa (cf. boundary problem [21]). Thus, the existence of isosemantic models and the boundary problem are two major barriers to semantic interoperability.

Let us use the Apgar test as example to clarify the distinction between clinical model, information model, reference terminology, representational unit, code and class/type:

- An Apgar test clinical model records a vitality score of a baby at one and five minutes after birth.
- A clinical terminology provides a representational unit such as in SNOMED CT “249228009 |Total apgar score (observable entity)|”
- This representational unit (in SNOMED CT called concept) is constituted by a code "302083008" and a label "Total Apgar score (observable entity)". It could be enriched by a textual and / or formal definition. Both code and label refer to the entity type Apgar score, the extension of which is the class of all individual Apgar scores. The individual Apgar score of the baby #123456 is a member of the class, which is the extension or the referent of "249228009 |Total Apgar score (observable entity)| "
- An information model provides information entities to record the numeric score (e.g. openEHR ELEMENT that allows recording an integer value within the range |0…10|).

In a terminology or ontology we will probably not find that the Apgar score has to be recorded at 1 and 5 minutes after birth, since this may vary according to the context and is not definitional for "Apgar score". However, we may find this information in a clinical model, from which we can derive that it is possible that CMs with multiple Apgar test exist. Neither does the terminology provide any information about numeric score values, which should, again, be provided by the clinical model, as recorded by the corresponding constrained information entity.

The above distinction has not been effectively put in practice by existing EHR representations. Recently the international Clinical Information Modeling Initiative (CIMI) has emerged with the goal “to improve the interoperability of healthcare information systems through shared implementable CMs” [3]. With this purpose, CIMI strives to provide a public CM repository [22] rendered in a common formalism, which can be translated to any other of the current patient record representations (e.g. openEHR, FHIR [23], CEM [24], etc.).

2.2 CIMI Clinical Models / Patterns

CIMI models seek to solve current EHR problems by proposing a set of modeling patterns, which are intended to act as guide for the creation of new CMs (e.g. blood pressure, pathology test, etc.). CIMI clinical patterns [25] pursue to define clinical semantics common to multiple clinical models to increase the design consistency between similar CMs [3]. Thus, each CIMI CM specializes and/or extends at least one CIMI clinical pattern. They are represented as CMs, by constraining the CIMI reference model (mainly derived from openEHR and ISO 13606 information models) and using the Archetype Definition Language 2.0 (ADL) [26]. Additionally, a UML representation of ADL is being developed as the Archetype Modeling Language (AML) [27] under the OMG. However, they are independent of implementation purpose, care setting context, specialty context and use case. They can be organized in hierarchies and related to each other by composition. Besides, for dealing with the existence of multiple iso-semantic models, CIMI proposes to associate each model with a set of isosemantic models, from which one is selected as the preferred one, with mappings established across them. CIMI uses SNOMED CT as reference terminology.

CIMI proposes the use of modelling patterns as a difference to other EHR modelling approaches such as openEHR or ISO 13606, which put more semantics into their reference models. The HL7 clinical statement model (formerly clinical statement pattern) represents a related approach to CIMI within HL7 to provide a pattern that guides RIM based modelling, focused on the HL7 *Act* information model structure [28]. Table 1 compares the four most prominent EHR standards whose modelling approach is based on a reference information model constraining mechanism (i.e. dual-model architecture [29]).

	ISO 13606	openEHR	CIMI	HL7
Most general descriptions of information entity components for clinical data management	Reference model Example: ENTRY, CLUSTER, etc.	Reference model Example: EVALUATION, ACTION, etc.	Reference model Example: CLUSTER, ELEMENT, etc.	Reference model: Example: ACT, ROLE, PARTICIPATION,
Constrained information entity components that describe common information modelling patterns	N/A	N/A	Reference model pattern	N/A
Constrained information entity components that describe common content patterns	N/A	N/A	Clinical model pattern	Clinical statement model
Constrained information entity components that describe clinical data recording scenarios	Archetype	Archetype	Clinical model	Template
Composition and further constraining of archetype nodes for content profiling (narrowing data recording scenarios)	N/A*	Template	N/A*	N/A

Table. 1 Comparison of information artefacts from four prominent EHR interoperability standards. *Template mechanism is possible whenever ADL2 is used [30].

In our opinion, the term "modeling pattern" or "design pattern" lacks a clear meaning within the EHR modelling community. In software engineering [31] design patterns are defined as: "a general reusable solution to a commonly occurring problem within a given context in software design; it is a description or template for how to solve a problem that can be used in many different situations". Applying this definition to EHR modelling we propose to distinguish between (i) modeling patterns that focus on the structure of information and (ii) modelling patterns that focus on representing the content embedded within that structure (what this structure is about or represents). In this paper, we will refer to them as **information modeling patterns** and **ontology content design patterns** (Content ODPs) respectively.

Information modeling patterns provide templates with a standard information structure to be reused in different data recording scenarios. In CIMI, information modeling patterns are named "reference model patterns" [25], examples are "clinical document", "clinical statement", "compound clinical statement", "cluster", etc.

Content ODPs focus on providing templates for recurrent ontology content (here the content embedded within CMs). They are domain-specific and are underpinned by formal-ontological descriptions [32]. SemanticHealthNet proposed examples for such patterns [33], such as the pattern for recording information about a patient's clinical situation or the pattern for describing clinical processes. Within CIMI, the so-called clinical pattern is what comes closest to Content ODP, although not related to any ontology. Examples are “Action”, “Observation set” or “Observation”. In this paper, when referring to CIMI patterns, we mean clinical patterns. Figure 2 depicts the hierarchy of CIMI models (CIMI patterns are also represented as CMs). Note that all elements of this hierarchy are information entities, i.e. labels like "Action" or "Diastolic blood pressure" should be read as "Information about action" and "Information about diastolic blood pressure", respectively.

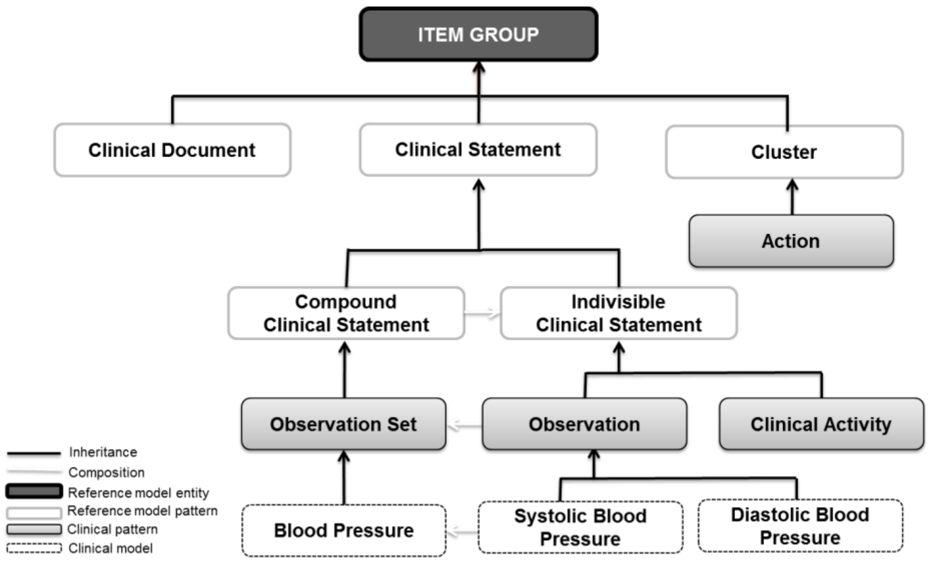


Figure. 2 CIMI hierarchy of models [25]. The root corresponds to a CIMI information model entity (ITEM_GROUP); White rectangles with continuous borders represent reference model patterns; Grey rectangles correspond to clinical patterns; White rectangles with dashed line represent CMs; Black arrows represent taxonomic subsumption; Grey arrows represent composition (composition corresponds to the “use_archetype” construct within CMs). Regarding the labeling, note that e.g. "Action" means information about action, and not action as an ontological category.

2.3 Ontology content design patterns (Content ODPs)

Building ontologies is a difficult task even for experts [34], and reusing, extending or composing existing ontologies is challenging [35]. Ontology design patterns (ODPs), a special case of software design patterns, were suggested as solutions to recurrent modeling problems [36], which guide and simplify ontology engineering [32] activities. Like software engineering, ontology engineering produces domain models, which suggest that software design pattern principles are reused, adapted and extended for the construction of ontology design patterns [36]. Several categories of ontology design patterns exist depending on their design focus: logical patterns, architectural patterns and content patterns [37].

Here we focus on content patterns, which are content-oriented structures that act as templates to represent modeling use cases (e.g. *Participation*, *Plan*, etc.). They extract a key and reusable fragment of a reference ontology that constitutes its background. Content patterns can be specialized and composed. A pattern p_2 specializes a pattern p_1 when at least one of the classes or properties from p_2 is a subclass or a subproperty of some class or property from p_1, while the remainder of p_1 is identical. Composing patterns allows combining them by linking ontology classes on the background of a common reference ontology [38].

SemanticHealthNet [12] proposed the use of ontology patterns with the main goal of bridging among heterogeneous EHR representations, in order to improve EHR semantic interoperability [10]. Ontology patterns are rooted in a Realist Ontology framework [39], which strictly distinguishes between real world entities, represented by classes like *Lung cancer*, *Blood pressure*, *Hemoglobin*, *Bronchoscopy* etc.) on the one hand, and information entities (e.g. *Lab test result*, *Diagnostic statement*, *Drug order* etc.) on the other hand. Information entities are related to real-world entities via the relation **represents**. This ontological framework encompasses the top-level ontology BTL2 (prefix “btl2”) [40] and SNOMED CT (prefix “sct”). BTL2 is a biomedical top-level ontology, which constrains heavily the freedom of the ontology engineer. SNOMED CT acts as common reference point for representing the clinical content. SNOMED CT subhierarchies and relations are partly harmonized with basic top-level classes and relations of BTL2 [41].

Content patterns have often been formally expressed along the Web Ontology Language OWL [13], which defines the formal semantics and expresses the relationships between the vocabulary items used in the pattern [42]. They usually require highly expressive fragments of the OWL language, making use of inverse properties, (qualified) cardinality restrictions, universal restrictions, etc. [43]. However, there is no standard notation for representing them.
Up to now, ODPs have been much less consolidated and have found much less acceptance compared to software engineering design patterns. One reason may be insufficient support by tools for finding suitable patterns, for pattern generation, integration and visualization [44]. However, there are two main pattern catalogues, the Manchester catalogue of ontology design patterns for bio-ontologies (MBOP) and the “Ontology Design Patterns.org” (ODP-Wiki)

catalogue [45]. Experiences include facilitate ontology lifecycle management [46] for Semantic Web Content, the development of biomedical ontologies [47], and linked data publication [48]. To our knowledge, there are no applications of ontology patterns within the healthcare information-modeling domain.

2.4 SHACL - Shapes constraint language

Shape Expressions (ShEx) [49] and SHACL [50] languages emerged within the W3C Data Shapes WG [51] as a response to the increasing interest in the last few years in creating a language that enables the description and validation of RDF data structures [52-53]. They intend to answer the need of a higher level, concise language for RDF validation and to foster the adoption of RDF-based solutions in scenarios where the data quality is important (e.g. industrial scenarios).

The Semantic Web provides a common framework to share and reuse data by using RDF as common data interchange format and OWL as a more expressive language, based on RDF, which allows inferring new knowledge from existing one by description logics (DL) reasoning [54]. Within the Semantic Web, the information has not to be complete, since it is intended to be shared and extended. As a consequence, OWL subscribes to the Open world assumption, i.e. that any statement that is not known can be true, and it does not subscribe to the Unique name assumption (UNA), which would require that whenever two objects have different identifiers, they are different. This works fine for the context of reusing and sharing within the Web but makes it difficult if the objective is data validation, especially in domains where data quality is important like in the clinical domain.

Although OWL is not intended for typical data validation purposes some workarounds have been proposed [55-57]. By typical data validation tasks we mean text operations, mathematical computations or cardinality constraint checking. E.g., the Open World Assumption obviates that an upper bound of the cardinality of a property at some class can be used to identify whether an individual is a member of that class, as there is no way to know whether the individual has additional properties for that type. This means that no inconsistency will be raised if the cardinality constraint is violated.

In the past, several approaches for data validation were proposed: SPARQL based (e.g. plain SPARQL [58], SPIN [59]), OWL based (e.g. Stardog ICV [60]) and Grammar based (e.g. OSLC Resource Shapes [61]) [62-63]. SHACL and ShEx provide RDF with validation capability in a similar way as XML Schema does regarding XML, although in a more expressive and extensible way. Both serve the same objective although with several differences, especially syntax and inference support. ShEx was designed as a high-level language intended to be human readable, which makes their syntax more intuitive. In contrast, SHACL design has been mainly influenced by the SPIN approach and is less user-friendly [63]. The working group is currently discussing a SHACL surface syntax similar to ShEx. A main reason to use SHACL is its support of basic inference in contrast to ShEx. Both SHACL and ShEx subscribe to the Closed World Assumption. However, SHACL provide several mechanisms that may interact with inference to support both open and closed world contexts.

Clinical data management and in general database engineering requires, as a first step, to check whether they follow certain data schema and comply with its constraints. In a second step, we may want to infer additional knowledge from data, which justifies the use of inferencing systems such as DL reasoners. That DL reasoning might be still a bottleneck due to the inherent complexity of most OWL dialects is an issue not discussed here.

SHACL is defined as an RDF vocabulary and supports the serialization formats from RDF. A key concept of SHACL is a “Shape”. An RDF Shape can be used to constrain and describe data. OWL ontologies are good for describing terms and their use but lack a mechanism for describing what kinds of things can be said with those terms, in contrast to RDF Shapes [64]. RDF Shapes declare constraints about RDF nodes in terms of expected cardinalities, datatypes and other restrictions. Figure 3 shows an excerpt of a SHACL Shape for describing the Observation result pattern described later within Section 3.2.

```
:ObservationResultShape rdf:type sh:Shape ;
        sh:scopeClass smp:ObservationResult ;
        sh:property[
                sh:predicate smp:IsResultOfProcess ;
                sh:minCount 1 ;
                sh:shape :ProcessShape ;
        ];
        sh:property[
                sh:predicate smp:isAbout ;
                sh:minCount 1 ;
                sh:classIn (btl2:Quality btl2:Situation) ;
        ];
        sh:property[
                sh:predicate smp:hasObservedValue ;
                sh:minCount 1 ;
                sh:class btl2:ValueRegion ;
        ];
        sh:property[
                sh:predicate smp:hasInformationPart ;
                sh:class btl2:InformationObject ;
        ].
```

Figure 3. RDF Shape for Observation Result pattern. It operates on RDF predicates that wrap more complex OWL expressions (see Section 3.2). It is described using Turtle syntax [65].

Within the above pattern we find the scope class (sh:scopeClass), which specifies which data instances denoted by a data graph are in the validation scope (here all with rdf:type smp:ObservationResult or its subclasses). The node validated against the shape is named focus node. Then, a list of constraints on the properties of the focus node is provided (sh:property). In this case they are cardinality (sh:minCount) and range constraints for the RDF predicates `smp:IsResultOfProcess`, `smp:isAbout` and `smp:hasInformationPart`. When no cardinality constraint is provided, zero or unbounded is assumed by default. The range constraints verify that each value node is of a certain type, which in this case might be another shape (sh:shape), a class (sh:class) or a list of classes (sh:classIn). Therefore, under a closed world interpretation, for an instance to match this shape it must be of the type smp:ObservationResult and must have minimum values for the three predicates constrained with minimum cardinality 1 and within the value ranges shown in the Figure. This shape allows data instances with other properties not to be constrained by the shape, although it is possible to restrict this shape to the listed ones only, by using sh:closed.

Although W3C's SHACL specification is not finished yet, most of the syntax can be considered stable and already suits the representation of clinical models.

3. Methods

3.1 Description of the approach

We have reused some of the ontology design patterns created within SemanticHealthNet and represented them in SHACL, with the aim to detect possible flaws in CIMI patterns. We focus on those CIMI patterns that represent observation results as outcome of certain request actions. The CIMI patterns under investigation are represented in RDF for validation. Figure 4 depicts the steps involved in this validation process. Here, the focus is set on checking (i) whether the CIMI patterns are correctly built from the point of view of a top-level or foundational ontology, and (ii) whether the terminology codes are correctly bound to the nodes of these patterns.

In the first task, the ontology patterns are used to check whether within each CIMI pattern the nodes are semantically related by analyzing whether it makes sense to include them within the same pattern or whether they should be part of a different one. This is possible because the ontology patterns are embedded into a highly axiomatized top-level ontology (i.e. BioTopLite 2 [39]). In this particular case we use BTL2 to guide the modeling of clinical information. Thus, we interpret each CIMI pattern as a small ontology and check whether it is built consistently regarding the ontological framework and the ontology patterns proposed.
For (2) we check the correctness of SNOMED CT bindings in CIMI patterns. This is possible since the ontological framework underpinning the ontology patterns includes SNOMED CT and most of their classes have been harmonized with basic top-level classes and relations of BTL2 [41].

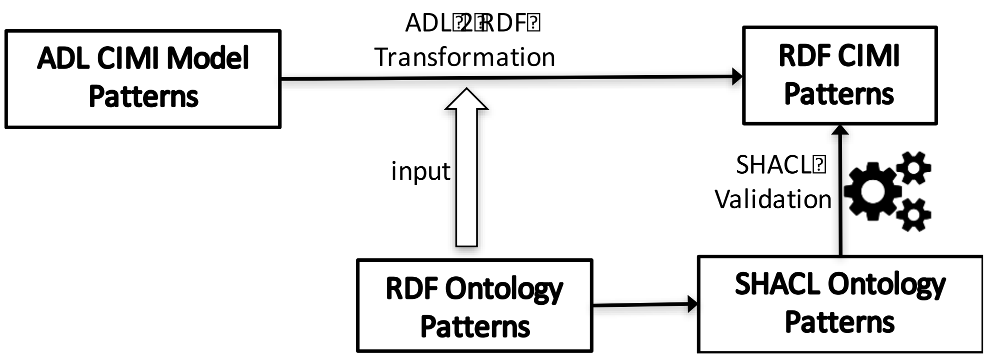


Figure 4. Steps involved in the ontology-based validation of CIMI patterns

In the following we describe the subset of CIMI clinical patterns and ontology content patterns selected.

3.1 Observation related CIMI clinical patterns

Figure 5 depicts the CIMI models used for validation. They represent observation results as outcome of a request action. Examples of clinical models based on them are laboratory test results or physical examination results.

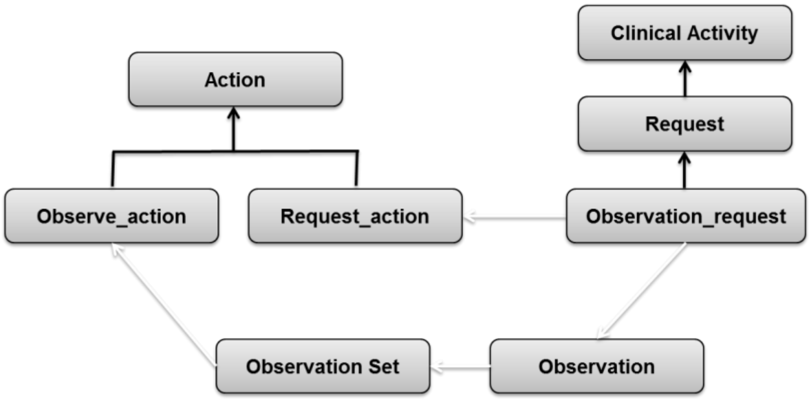


Figure. 5 CIMI clinical patterns to represent observation results as outcome of request actions; Grey rectangles correspond to clinical patterns; Black arrows represent inheritance; Grey arrows represent composition (composition corresponds to the “use_archetype” construct within CMs).

Figure 6 provides an excerpt of the observation clinical pattern represented in ADL [26]. Three main sections of the model are shown: (i) the archetype section, which univocally identifies the model; (ii) the definition section which provides the data entry points in a tree-like structure (as constrained information model entities) required within a clinical recording scenario; (iii) the terminology section, which provides textual descriptions and external terminology bindings of the data entry points at the definition section.

```
archetype (adl_version=1.5.1; rm_release=2.0.2; generated)
    CIMI-CORE-ITEM_GROUP.observation.v1.0.0
definition
    ITEM_GROUP[id1.1.1] matches {-- Observation
item matches {
   ELEMENT[id5.0.2] matches { -- Code
     value matches {CODED_TEXT[id0.0.102]}}
    ELEMENT[id5.0.3] matches { -- Reason
     value matches {TEXT[id0.0.103]}}
    ELEMENT[id5.0.4] matches { -- Method
     value matches {TEXT[id0.0.104]}}
    ELEMENT[id5.0.5] occurrences matches {0..1} matches { -- Status
     value matches {TEXT[id0.0.105]}}
    ELEMENT[id5.0.6] occurrences matches {0..1} matches { -- Priority
     value matches {TEXT[id0.0.106]}}
    ELEMENT[id5.0.7] occurrences matches {0..1} matches { -- Interpretation category
     value matches {TEXT[id0.0.107]}}
    ELEMENT[id5.0.8] occurrences matches {0..1} -- Result
    use_archetype
        ITEM_GROUP [id4.0.1, CIMI-CORE-ITEM_GROUP.observation_action_set.v1] -- Action set
}}
terminology
   term_bindings = < ["snomed-ct"] = < items = <
        ["id5.0.3"] = <http://snomed.info/id/363702006> -- has focus
        ["id5.0.4"] = <http://snomed.info/id/260686004> -- method
        ["id5.0.5"] = <http://snomed.info/id/408730004> -- procedure context
        ["id5.0.6"] = <http://snomed.info/id/260870009> -- priority
        ["id5.0.7"] = <http://snomed.info/id/363713009> -- has interpretation
  >>>
```

Figure. 6 Excerpt of the observation clinical pattern. Note that all terminology bindings target SNOMED CT relations, i.e. the meaning of "Reason", "Method" etc. is relational.

The model example above shows that information model entities (e.g. ELEMENT or ITEM_GROUP) are abstract information entities without formalized meaning. Terminology bindings assign them a meaning through external terminology codes, such as SNOMED CT in this example. Note that terminology binding is a kind of denotation (e.g. like the string "Catalina Martínez-Costa" denotes the first author of this paper) and therefore fundamentally different from terminology mapping). For this process of selecting the appropriate terminology code, no clear constraints and criteria have been formulated so far. In this example all codes, interestingly represent SNOMED CT semantic relations (aka *linkage concepts in* SNOMED parlance), i.e. such as **priority**, **method**, **has focus**, etc. Table 2 shows the SNOMED CT domain and range concept values for each relation. These constraints were taken from the SNOMED CT constraint expression language [66] and should be interpreted like domain and range restrictions of OWL object properties.

Not all SNOMED CT relations shown in Table 2 share the same domain. Whereas the relations **has focus**, **method**, and **priority** have their domain restricted by SNOMED CT *Procedure* concepts, the relation **procedure context** must have a *Situation with explicit context* as domain; and for **has interpretation** the domain is restricted to *Clinical finding* concepts.

However, all these CM nodes are children of the same root node (ITEM_GROUP[id1.1.1]), which corresponds to an *Observable entity* concept[8] in SNOMED CT (e.g. *Systolic blood pressure*).

Intuitively we could think that the nodes under the root should share the same domain and therefore be allowed relations of an *Observable entity* concept as domain (e.g. what is observed, the observation result, the observation result interpretation, etc.). However, by looking at the clinical model nodes *Priority*, *Status*, *Method* and *Reason*, they better describe the observation process rather than the observation target.

CM node	**Corresponding SNOMED CT relation**	**Domain restriction concept**	**Range restriction concept**	**Constraint Tag**
Reason	**has focus**	*Procedure*	*Procedure* *Clinical Finding*	(<<) (<<)
Method	**method**	*Procedure*	*Action*	(<<)
Status	**procedure context**	*Situation with explicit context*	*Context values for actions*	(<=)(< Q)
Priority	**priority**	*Procedure*	*Priorities*	(<=)(< Q)
Interpretation category	**has interpretation**	*Clinical finding*	*Findings values*	(<<)

Table. 2 Domain and range values for the SNOMED CT relations (aka "linkage concepts") bound to the observation clinical pattern model, according to the SNOMED CT concept model; in addition, the following tags from the SNOMED CT constraint expression language [66] are used: (<<) this code and descendants; (<=) descendants only (because the code itself is a grouper, not meant for coding); (< Q) means descendants only

[8] In SNOMED CT observables are entities that could be used to code elements where a value can be assigned

when in a qualifying relationship (a relationship not part of the concept's definition, which indicates a possible attribution used to refine a post-coordinated expression that refers to the source concept)

3.2 Observation related ontology content design patterns

In this work we will reuse the patterns proposed within the SemanticHealthNet project for modeling the observation-related CIMI models described in Section 3.1 (see Table 3). They are top-level patterns, which means that they provide generic classes and properties and can be specialized and composed in order to model specific clinical information modeling use cases.

Content pattern name	**Pattern description**
Observation Result	Record entry that represents the result of observing or assessing some quality related with the health status of a patient (e.g. record of: body weight, hemoglobin concentration in blood, etc.)
Clinical Process	Clinical process description (e.g. observation, assessment, history taking, request process, physical examination, etc.)
Planned Clinical Process	Record entry about the intent to perform some healthcare related process (e.g. request to administer some drug, plan to reach some target body measurement (e.g. weight), request to perform some healthcare service (e.g. check potassium level, etc.)

Table 3. Ontology top-level design patterns used for modeling the observation related CIMI models

As mentioned in section 2.3 there is no standardized representation for ontology design patterns, although they have been often represented in OWL DL. However, the use of OWL for describing patterns is an approximation since as previously described it aims to describe a domain, not a model of that domain. Figure 7 provides a graphical representation of the Observation result pattern and its rendering using OWL 2 Manchester syntax [67] is shown in Figure 8. Both have been selected for didactic reasons.

The *ObservationResult* class is described by three OWL relations (object properties), which indicate that an observation result (i) might have as part other information entities (e.g. some comment); (ii) must be the result of performing at least one process such as an observation procedure; and (iii) must represent at least a quality or a certain situation of a patient (e.g. normal blood pressure).

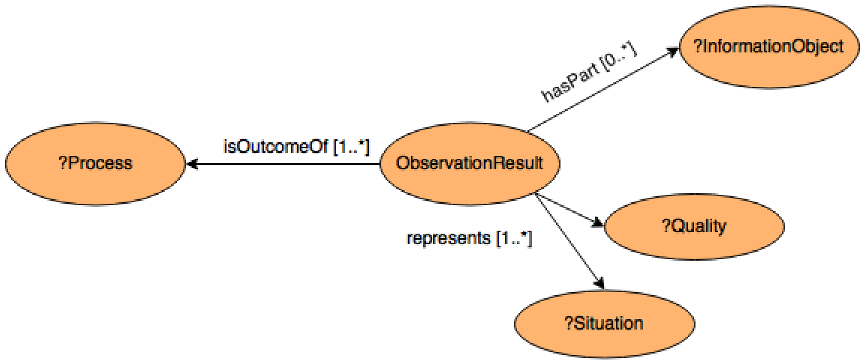


Diagram elements	**Ontology elements**
Observation result	*smp:ObservationResult* **rdf:type** *owl:Class*; **rdfs:subClassOf** *btl2:InformationObject*
InformationObject	*btl2:InformationObject* **rdf:type** *owl:Class*
Quality	*btl2:Quality* **rdf:type** *owl:Class*
Situation	*btl2:Situation* **rdf:type** *owl:Class*
Process	*btl2:Process* **rdf:type** *owl:Class*
isOutcomeOf	**btl2:isOutcomeOf rdf:type** *owl:ObjectProperty*
hasPart	**btl2:hasPart rdf:type** *owl:ObjectProperty*
represents	**btl2:represents rdf:type** *owl:ObjectProperty*

Figure 7. Graphical representation of the Observation Result Pattern and description of the corresponding ontology elements. A question mark in the name of an OWL class represents a variable within a pattern.

Class: *smp:ObservationResult*
SubClassOf: *btl2:InformationObject*,
btl2:hasPart some *?btl2:InformationObject*,
btl2:isOutcomeOf some *?btl2:Process*,
btl2:represents only (*?btl2:Quality* or *?btl2:Situation*)

Figure 8. OWL DL Manchester syntax representation of the Observation Result Pattern. Using OWL for patterns is an approximation, for which we recommend a kind of simple OWL meta-language using placeholders. A question mark in the name of an OWL class represents a variable within a pattern. In English: "Every observation result is an information object. It might have information object entities as part and be the outcome of some process. Observation results represent anything than it is of the type quality or situation."

According to the above, an instance of the observation result class that describes “normal blood pressure” would be represented as follows:

Individual: Observation_result_about_blood_pressure#12345
Types: *btl2:ObservationResult*
Facts:
btl2:hasPart This_is_an_example_of_comment#12345**,**
btl2:isOutcomeOf Observation_Process#12345,
btl2:represents Blood_pressure_observable#12345 ,
btl2:represents Normal_Blood_Pressure#12345

Individual: This_is_an_example_of_comment#12345
Types: *btl2:InformationObject*

Individual: Observation_Process#12345
Types: *sct:ObservationalAssessment*

Individual: Blood_pressure_observable#12345
Types: *sct:BloodPressure*
Facts:
btl2:projectsOnto Normal_Value#12345

Individual: Normal_Value#12345
Types: *sct:Normal*

Individual: Normal_Blood_Pressure#12345
Types: *sct:NormalBloodPressure*

Figure 9. Example instance of the class Observation Result "normal blood pressure" using Manchester syntax

Although ODPs aim at simplifying the task of building ontologies, their use by non-ontology experts might still be tedious, even more if they are based on a foundational ontology, which requires the use predefined classes and relations which are only understandable for who is familiar with this ontology (e.g., *Quality*, **isOutcomeOf)**. In SemanticHealthNet we had proposed to simplify and represent them in RDF, in order to facilitate their use without the need of understanding the underlying ontology. This simplification consists in expressing OWL expressions as RDF predications with close-to-user labels. The complexity of an OWL axiom is thus hidden behind an RDF predicate, with the subject of the triple being the definiendum in the OWL axiom and the object being one of the class symbols occurring in the definiens. In the easiest case a predicate `smp:p` is a substitutent for the OWL expression fragment "subclassOf **r** some". (e.g., the expression fragment "subclassOf **btl2:isOutcomeOf** some" corresponds to the predicate `smp:isResultOfProcess`). An example of a more complex shortcut is shown in Table 4.

OWL DL	**RDF**
Class: *smp:ObservationResult*	
SubClassOf: *btl2:InformationObject* and **btl2:represents** only (?*btl2:Quality* and **btl2:projectsOnTo** some ?*btl2:ValueRegion*)	*smp:ObservationResult* `smp:hasObservedValue` *btl2:ValueRegion.*

Table 4. Example of correspondence between the simplified RDF and the OWL DL representation. The underlined fragment of the OWL code corresponds to the predicate `smp:hasObservedValue`.

Thus, patterns are easier to implement and to use, because correspondences are defined regarding the OWL DL representation, both of which can be interchangeably used. The meaning of these RDF representations around these predicates is mostly intuitive. In case of doubt, the underlying OWL formalization reveals the intended meaning. Figure 10 shows the RDF representation of the Observation Result Pattern.

smp:ObservationResult `smp:hasInformationPart` btl2:InformationObject .

smp:ObservationResult `smp:isResultOfProcess` *btl2:Process* .

smp:ObservationResult `smp:isAbout` *btl2:Quality* .

smp:ObservationResult `smp:isAbout` *btl2:Situation* .

smp:ObservationResult `smp:hasObservedValue` *btl2:ValueRegion* .

Figure 10. RDF triples [67] representation of the Observation Result Pattern. OWL Object properties correspond now to RDF predicates (prefix "smp")

4. Results

4.1 Representation of CIMI Patterns in RDF

Since we aim at validating CIMI patterns by using RDF Shapes, we need them to be represented as RDF instances and by following the ontology patterns proposed. To this end we have manually defined the correspondences between them and the ontology patterns. This mapping process consists in first selecting a single or a list of ontology patterns that best represent the CIMI pattern. Secondly, establishing the correspondences between the CIMI model nodes and the ontology pattern triples.

In total we have created RDF instances for six CIMI patterns whose purpose is the representation of observation results (see Table 5). The CIMI patterns are freely accessible at their model repository via web [22] and represented in the ADL language. As ontology patterns we have used the two described in Table 3.

Selected CIMI CMs
CIMI-CORE-ITEM_GROUP.observation.v1.0.0
CIMI-CORE-ITEM_GROUP.observation_action_set.v1.0.0
CIMI-CORE-ITEM_GROUP.observe_action.v1.0.0
CIMI-CORE-ITEM_GROUP.observation_request.v2.0.0
CIMI-CORE-ITEM_GROUP.request_action.v1.0.0
CIMI-CORE-ITEM_GROUP.observation_set.v1.0.0

Table 5. CIMI clinical patterns related to the modelling of observation results

As an example, Tables 6 and 7 show the correspondences between the nodes of the CIMI model CIMI-CORE-ITEM_GROUP.observation.v1.0.0 and the ontology pattern triples. We used two ontology patterns in order to represent the meaning of the CIMI model, *viz.* the Observation result and the Clinical process patterns, related to each other by composition via the RDF predicate `smp:isResultOfProcess`. The definition of the correspondences between the CIMI models and the ontology patterns and their application to build the RDF CIMI representation are at the moment manual tasks that should ideally be assisted by a dedicated tool.

CIMI Model Node path	RDF ontology pattern triple
[id5.0.2] – Code	*smp:ObservationResult* `smp:isAbout` *btl2:Quality*
item[id5.0.7] – Interpretation category	*smp:ObservationResult* `smp:isAbout` *btl2:Situation*
item[id5.0.8] – Result	*smp:ObservationResult* `smp:hasObservedValue` *btl2:ValueRegion*
item[id4.0.1] – Action set	*smp:ObservationResult* `smp:isResultOfProcess` *btl2:Process*

Table 6. Correspondences between the CIMI model CIMI-CORE-ITEM_GROUP.observation.v1.0.0 and the Observation result ontology pattern

CIMI Model Node path	RDF ontology pattern triple
item[id5.0.3] – Reason	*smp:Process* smp:isDueTo *btl2:Process*
item[id5.0.4] – Method	*smp:Process* smp:hasQuality *btl2:ValueRegion*
item[id5.0.5] – Status	*smp:Process* smp:hasQuality *btl2:ValueRegion*
item[id4.0.6] – Priority	*smp:Process* smp:hasQuality *btl2:ValueRegion*

Table 7. Correspondences between the CIMI model CIMI-CORE-ITEM_GROUP.observation.v1.0.0 and the Clinical Process ontology pattern

Once the correspondences have been defined, we have substituted the object part of each triple for the corresponding terminology binding as provided for each of the CIMI model nodes (see Table 2). As an example, the node that describes the status of the observation (i.e. item[id5.0.5]) is bound to the SNOMED CT relation **procedure context**, which, according to the SNOMED CT concept model has as domain the concept *Situation with explicit context* and as range all descendant concepts of *Context values for actions*. The resulting triple to describe the status of the observation is the following: cimi:ObservationAction#12345 cimi:hasStatus cimi:ActionValues#12345. The cimi:ObservationAction#12345 instance is of the type *smp:Process* and cimi:ActionValues#12345 is of the type *sct:ProcedureContext* (i.e. the SNOMED CT relation **procedure context**). The predicate cimi:hasStatus is subpredicate of smp:hasQuality. Table 8 shows the OWL DL representation of the RDF triple *smp:Process* smp:hasQuality *btl2:ValueRegion*.

OWL DL	**RDF**
Class: *smp:Process* SubClassOf: *btl2:Process*, **btl2:isBearerOf** only (btl2:*Quality* and **btl2:projectsOnto** some btl2:*ValueRegion*)	*smp:Process* smp:hasQuality *btl2:ValueRegion*

Table 8. Example of correspondence between the RDF simplified and the OWL DL representation

```
cimi:ObservationResult#12345
        rdf:type smp:ObservationResult ;
        cimi:resultsFromProcess cimi:ObservationAction#12345 ;
        cimi:isAbout cimi:ObservableEntity#12345 ;
        cimi:hasInterpretation cimi:ClinicalFinding#12345 ;
        cimi:hasObservedValue cimi:ObservedValueObservationResult#12345 .
cimi:ObservationAction#12345
        rdf:type smp:Process ;
        cimi:IsDueTo cimi:DueToObservationAction#12345 ;
        cimi:hasMethod cimi:Technique#12345 ;
        cimi:hasStatus cimi:ActionValues#12345 ;
        cimi:hasPriority cimi:PriorityValues#12345 .
cimi:resultsFromProcess rdf:Property; rdfs:subPropertyOf smp:IsResultOfProcess .
cimi:isAbout rdf:Property; rdfs:subPropertyOf smp:isAbout .
cimi:hasInterpretation rdf:Property; rdfs:subPropertyOf smp:isAbout .
cimi:hasObservedValue rdf:Property; rdfs:subPropertyOf smp:hasObservedValue .
cimi:IsDueTo rdf:Property; rdfs:subPropertyOf smp:isDueTo .
cimi:hasMethod rdf:Property; rdfs:subPropertyOf smp:hasQuality .
cimi:hasStatus rdf:Property; rdfs:subPropertyOf smp:hasQuality .
cimi:hasPriority rdf:Property; rdfs:subPropertyOf smp:hasQuality .

cimi:DueToObservationAction#12345 rdf:type btl2:Situation .
cimi:ObservableEntity#12345 rdf:type sct:ObservableEntity.
cimi:ClinicalFinding#12345 rdf:type sct:ClinicalFinding .
cimi:Technique#12345 rdf:type sct:Technique .
cimi:ActionValues#12345 rdf:type sct:ProcedureContext .
cimi:PriorityValues#12345 rdf:type sct:Priority .
cimi:ObservedValueObservationResult#12345 rdf:type btl2:ValueRegion .
```

Figure 11. RDF Turtle representation of the CIMI pattern CIMI-CORE-ITEM_GROUP.observation.v1.0.0 based on the Observation result and Clinical process ontology patterns

By representing the CIMI patterns in RDF according to the ontology patterns, we can find modeling errors by checking whether they comply with the proposed formal ontological framework. For the observation CIMI pattern, we found some of these errors. As already foreseen in section 3.1 after examining the model terminology bindings, not all nodes under the CIMI root model node share the same domain concept and this is confirmed after defining the correspondences with the observation result pattern and not finding any for some of the nodes (i.e. reason, method, priority, status). Instead these nodes can be mapped to certain triples of the Clinical process ontology pattern. Based on this, we can conclude that part of the content modelled by the CIMI observation result pattern should be rather described in a pattern intended to describe processes, which already exists (CIMI-CORE-ITEM_GROUP.action.v1). Within the CIMI models analyzed, we found at least eleven modeling errors according to our proposed ontological framework. In the next section we will see how to detect both modelling and terminology binding errors automatically by using SHACL ontology patterns.

4.2 Validation of CIMI Patterns using SHACL

We have represented the ontology patterns in SHACL as described in Section 2.4. Figure 12 shows an excerpt of the RDF Shape for the Clinical Process pattern. This shape has as scope all instances of the type *smp:Process* or any of its subtypes and provides a list of constraints which specify their valid predicates with their corresponding ranges and cardinality constraints.

```
:ProcessShape rdf:type sh:Shape ;
        sh:scopeClass smp:Process;
        sh:property[
                sh:predicate smp:hasTemporalValue;
                sh:class btl2:TemporalRegion;
        ];
        sh:property[
                sh:predicate smp:hasResult;
                sh:class btl2:InformationObject;
        ];
        sh:property[
                sh:predicate smp:hasParticipant;
                sh:classIn (btl2:MaterialObject btl2:InformationObject) ;
        ];
        sh:property[
                sh:predicate smp:hasQuality;
                sh:class btl2:ValueRegion ;
        ];
        sh:property[
                sh:predicate smp:isDueTo;
                sh:classIn (btl2:MaterialObject btl2:InformationObject btl2:Process btl2:ImmaterialObject
btl2:Disposition btl2:Situation);
        ].
```

Figure 12. Excerpt of the RDF Shape for the Clinical Process pattern

After representing CIMI models as RDF instances we can check if they are correctly modelled by looking that they comply with the pattern description given by the constraints of the shape. For that we have used the Java open source implementation of SHACL [68]. Here, since CIMI RDF instances have been created by manually defining the correspondences with the proposed ontology patterns, we have already detected the modelling errors and provided a valid CIMI RDF representation from the modelling perspective (see Section 4.1). However, we have not checked if the terminology bindings have been correctly assigned to the CIMI models. Since the ontological framework underpinning the ontology patterns includes SNOMED CT, and most of SNOMED CT classes have been harmonized with basic top-level classes and relations of BTL2 we will be able to check them.

Figure 13 shows the example of a violation message stating that the range of the predicate `smp:hasPriority` should be of the RDF type *btl2:ValueRegion* instead of *cimi:PriorityValues* (which is of the type **priority** in SNOMED). The SNOMED **priority** relation belongs to the attribute hierarchy according to the concept model and in BTL2 it is a subproperty of the BTL2 object property **btl2:isBearerOf**. Therefore in this case there is a binding error, since the value should be of the type *btl2:ValueRegion*. According to the SNOMED CT concept model the right binding should be of the type *Priorities*, which belong to the qualifier value SNOMED hierarchy.

Shape: :ProcessShape
Focus node: cimi:#ObservationAction#12345
Error Message: Error in predicate smp:hasPriority –
Value does not have class btl2:ValueRegion –
Actual value is cimi:PriorityValues

Figure 13. Violation message triggered by the execution of the SHACL ontology top-level patterns on the RDF instance shown in Figure 11

Most of existing CMs, as in the case of CIMI models, have terminology bindings assigned to their information structures. However, terminology bindings are usually not present for all information structures within a model, and due to the high flexibility provided by CMs and the lack of explicit relationships between their elements, there is no unique way of doing the binding, despite several strategies proposed in recent years [69-73]. This lack of criteria prevents semantic interoperability, making possible to link an information structure to a concept or attribute indistinctly as we have noticed in the CIMI-CORE-ITEM_GROUP.action.v1.0.0. and CIMI-CORE-ITEM_GROUP.observation.v1.0.0 models, where bindings of semantically equal information structures have been done in a heterogeneous manner.

Another example in which a violation is raised is shown in Table 9. In this case, within a hierarchy of CIMI models we find terminology binding inconsistencies for certain information structures. The CIMI model *CIMI-CORE-ITEM_GROUP.observe_action.v1.0.0* specializes the model *CIMI-CORE-ITEM_GROUP.action.v1*. Both model patterns provide information structures for recording the action timing offset (i.e. [id2.6] and [id2.6.1]) and the action timing origin (i.e. [id2.7] and [id2.7.1]). For the parent model, they are bound to the SNOMED CT concept *Temporal observable*. However, the terminology bindings of the specialized model are wrong (i.e. bound to *Person* and *Environment or geographical location*) and a violation will be raised after checking the models against the SHACL ontology patterns.

CIMI-CORE-ITEM_GROUP.action.v1		CIMI-CORE-ITEM_GROUP.observe_action.v1.0.0	
CIMI Model Node path	SNOMED CT terminology binding	CIMI Model Node path	SNOMED CT terminology binding
[id1.1]	*Action*	[id1.1.1]	*Action*
item[id2.6]	*Temporal observable*	item[id2.6.1]	*Person*
item[id2.7]	*Temporal observable*	item[id2.7.1]	*Environment or geographical location*

Table 9. Excerpts of two CIMI model patterns hierarchically related. (the model at the right is specialization of the one at the left)

In the examples described, no violation regarding cardinality constraints were raised since we have instantiated every node within the CIMI pattern due to our purposes (i.e. to detect modeling and terminology binding errors). A typical validation scenario not focused on detecting modelling errors would consist on representing CIMI patterns in SHACL, as specialization of the top-level ontology patterns and checking if a concrete patient CIMI RDF instance complies with the CIMI model.

5. Discussion and conclusions

Clinical models are the core of current EHR standards, and they are advertised with the argument that by sharing the same set of clinical models, semantic interoperability between clinical data from different sources should be granted. In practice, however, most clinical models have been built from a bottom-up perspective, strongly driven by specific data recording use cases, with data recording needs hardly ever being the same across different healthcare providers. This results a situation in which the interoperability desideratum is not met although the same modelling principles are shared.

Aware of the above limitation, initiatives and projects like CIMI or SemanticHealthNet have proposed the use of patterns, but by following different approaches.

CIMI focuses on the use of modelling patterns for guiding the building of clinical models. In order to share meaning between multiple clinical models and to increase the design consistency between them, they propose two kinds of patterns: (i) reference model patterns and (ii) clinical patterns. While (i) focus on reusing standard information structures, (ii) focus on modelling repetitive content aspects. Both are represented as clinical models.

SemanticHealthNet proposed to dissect the meaning of clinical content using a formal-ontological approach, using a highly constrained ontology framework under a top-level ontology. Its constraints are not arbitrary but grounded on formal theories and prevent users from modelling decisions that contradict the way upper-level categories like *Material object*, *Information object*, or *Process* are defined and how they are related to each other. The use of this highly complex axiomatic framework requires a deep understanding of the underlying top-level ontology which requires a learning process towards "ontological thinking". This effort is largely reduced when ontology content design patterns are used, which hide modelers from that complexity and keep the axiomatic theory "under the hood". As the underlying ontology accounts for all aspects of information models, i.e. both ontological and epistemic content, they are no longer dependent of any particular EHR representation.

The decision to select one or another semantic framework to represent clinical information is arguable. BTL2 has been chosen as a top-level ontology given its compromise between degree of formalization (i.e. constraining axioms) and complexity, contributing to a quick learning curve. BTL2 was not intended to compete with established upper level ontologies, but rather to integrate with them. BTL2 bridges with other ontologies such as BFO, DOLCE and RO [39]. SNOMED CT as a domain ontology was mainly chosen by its formal grounding, its high clinical content coverage and its increasing global acceptance. In addition, the most important SNOMED CT top-level categories have been harmonized with BTL2, which makes them seamlessly fit into an ontology framework that represents both clinical and information entities. Other terminologies can be used in clinical domain representations, using SNOMED CT as reference terminology.

In this work we have demonstrated how ontology content design patterns can be used to detect modelling errors in existing clinical models. As a use case we have selected a set of CIMI patterns, represented as clinical models. Other EHR modelling options could have been selected [9,74]. However, CIMI was chosen, because this approach is equally pattern-based, however without ontological rooting.

For the validation, both ontology patterns and CIMI models have been represented in RDF. RDF has been recently proposed as the universal healthcare exchange language by the Yosemite project, which pursues semantic interoperability of structured healthcare information. As discussed in the paper, OWL subscribes to the Open World Assumption and is therefore neither intended nor suited for typical data validation tasks. However, OWL is useful when we want to infer additional knowledge from data. We have represented ontology content design patterns as RDF Shapes using SHACL. An RDF Shape is used to constrain and describe data, which fits well with the purpose of clinical models to describe data recording scenarios by constraining information model structures. RDF Shapes declare constraints about RDF nodes in terms of expected cardinalities, datatypes and other restrictions.

We have reused a set of ontology patterns from SemanticHealthNet and represented them in SHACL in order to check whether (i) CIMI patterns are correctly built from the perspective of a top-level ontology and (ii) whether terminology codes are correctly bound to the nodes of the patterns.

Six CIMI clinical models were manually connected to two SHACL ontology patterns. CIMI models have then been represented as RDF triples accordingly. Several modelling and binding errors could be detected after the application of the ontology patterns represented as RDF Shapes. In this case, no cardinality violations were raised due to the nature of our study, which focused on modelling and binding errors. A typical validation scenario not focused on detecting modelling errors would consist on representing CIMI patterns in SHACL, as specialization of the top-level ontology patterns and checking if a concrete patient CIMI RDF instance complies with the CIMI model.

Although the surface syntax of SHACL may be considered inelegant, SHACL provides mechanisms to support inference, allowing, e.g., to apply the ontology pattern used to describe clinical processes to any subclass of the *Process* class in BTL2. Providing it with a simpler syntax is being study under the W3C group.

The representation of ontology patterns as RDF Shapes in SHACL provides other potential benefits. One of these benefits is to provide with more tooling possibilities, which is one of the limitations of using ontology patterns. SHACL can also be used for code generation, with the purpose to generate complex OWL DL axioms out of the RDF representation or even for generating natural language descriptions from the axioms. It serves also as integration tool and might be used to integrate heterogeneous clinical information representations. Representing ontology patterns in SHACL provides an outline to query and can be used for facilitating the interface building. Besides, SHACL is compatible with linked data principles [75].

Conflict of interest

The authors have no conflict of interest

Acknowledgments

Daniel Karlsson (University of Linköping) participated in the initial discussions regarding the use of ontology design patterns for modelling CIMI Observation related clinical models. Thomas Beale (Intermountain Healthcare) reviewed an early version of the paper. Both provided valuable input.

Funding

Initial work was supported by the EU Network of Excellence project SemanticHealthNet (Semantic Interoperability for Health Network, FP7-ICT-2011-7, Grant agreement no.: 288408).

References

1. Beale, T., & Heard, S. Archetype Definitions and Principles versión 1.0.2. http://www.openehr.org/releases/1.0.2/architecture/am/archetype_principles.pdf (accessed May 2017).
2. HL7 Implementation Guide for CDA® Release 2: IHE Health Story Consolidation, Release 1.1 - US Realmhttp://www.hl7.org/implement/standards/product_brief.cfm?product_id=258 (accessed May 2017).
3. Clinical Information Modeling Initiative (CIMI). http://informatics.mayo.edu/CIMI/index.php/Main_Page (accessed May 2017)
4. HL7 Version 3 Reference Information Model (RIM). http://www.hl7.org/implement/standards/product_brief.cfm?product_id=77 (accessed May 2017)
5. Beale T, Lloyd D, Heard S, et al. (editors). The openEHR Reference Model version 1.0.2. http://www.openehr.org/programs/specification/releases/1.0.2 (accessed May 2017).
6. ISO/TC 215, Health informatics (2008). Electronic Health Record Communication Part 1: Reference Model (ISO 13606-1:2008)
7. Schulz S, Jansen L. Formal ontologies in biomedical knowledge representation. Yearbook of Medical Informatics 2013;8(1):132-46
8. SNOMED International. http://www.ihtsdo.org/ (accessed May 2017).
9. Martínez-Costa, C., Martínez-Costa, C., & Schulz, S. (2014). Ontology content patterns as bridge for the semantic representation of clinical information. Applied clinical informatics, 5(3), 660.
10. Martínez-Costa, C., Cornet, R., Karlsson, D., Schulz, S., & Kalra, D. (2015). Semantic enrichment of clinical models towards semantic interoperability. The heart failure summary use case. Journal of the American Medical Informatics Association, 22(3), 565-576.
11. Falbo, R. A., Guizzardi, G., Gangemi, A., & Presutti, V. (2013, October). Ontology patterns: clarifying concepts and terminology. In Proceedings of the 4th International Conference on Ontology and Semantic Web Patterns-Volume 1188 (pp. 14-26). CEUR-WS. org.
12. Semantic Interoperability for Health Network (SHN). http://www.semantichealthnet.eu/ (accessed May 2017).
13. W3C OWL working group. OWL 2 Web Ontology Language, Document Overview. W3C Recommendation 11 December 2012. http: //www.w3.org/TR/owl2-overview (accessed May 2017).
14. W3C Shapes Constraint Language (SHACL). Candidate recommendation 11 April 2017. http://www.w3.org/TR/shacl/ (accessed May 2017)
15. W3C Resource Description Framework (RDF). Published 25 February 2014. https://www.w3.org/RDF/ (accessed May 2017)
16. Semantic interoperability Yosemite project. http://yosemiteproject.org (accessed May 2017)
17. Electronic Health Records for Clinical Research project. http://www.ehr4cr.eu (accessed May 2017)
18. Ogden, C. K., Richards, I. A., Malinowski, B., & Crookshank, F. G. (1946). The meaning of meaning (pp. 109-38). New York: Harcourt, Brace & World.
19. Schulz, S., Martínez-Costa, C., Karlsson, D., Cornet, R., Brochhausen, M., & Rector, A. L. (2014, September). An Ontological Analysis of Reference in Health Record Statements. In FOIS (pp. 289-302).
20. International Classification of Diseases, Eleventh Revision (2017). http://www.who.int/classifications/icd/revision/en/ (accessed May 2017)

21. Rector A, Qamar R, Marley T. Binding Ontologies & Coding systems to Electronic Health Records and Messages, Appl Ontol 2009;4:51-69.
22. CIMI Model Browser. http://opencimi.org/model-browser (accessed May 2017)
23. HL7 FHIR. https://www.hl7.org/fhir/ (Accessed May 2017)
24. Clinical Element Model (CEM). http://www.clinicalelement.com/docs/CEReference20081114.pdf / (Accessed May 2017)
25. CIMI Modelling Taskforce Progress Report. November 2014. http://www.opencimi.org/document/CIMI%20Modelling%20Taskforce%20Progress%20Report (Accessed May 2017)
26. Archetype Definition Language (ADL 2). http://www.openehr.org/releases/trunk/architecture/am/adl2.pdf (Accessed May 2017)
27. AML – OMG Archetype Modeling Language. http://www.omg.org/spec/AML/ (Accessed May 2017)
28. Benson, T. (2012). Principles of health interoperability HL7 and SNOMED. Springer Science & Business Media.
29. Beale, T. (1999, November). The GEHR software architecture for a reliable EHR. In Toward an Electronic Health Record Europe (Vol. 99, pp. 328-339).
30. Working with Templates: http://www.openehr.org/downloads/ADLworkbench/working_with_templates (Accessed May 2017)
31. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design patterns: elements of reusable object-oriented software. 1994. ISBN- 0-201-63361-2
32. Falbo, R. A., Guizzardi, G., Gangemi, A., & Presutti, V. (2013, October). Ontology patterns: clarifying concepts and terminology. In Proceedings of the 4th International Conference on Ontology and Semantic Web Patterns-Volume 1188 (pp. 14-26). CEUR-WS. org.
33. SemanticHealthNet Year 2, deliverable 4.2: Ontology/Information models covering the HF use case, 2013. http://www.semantichealthnet.eu/index.cfm/deliverables/ (Accessed May 2017)
34. Noppens, O., Liebig, T., Ontology Patterns and Beyond - Towards a Universal Pattern Language. In: Proceedings of the Workshop on Ontology Patterns (WOP 2009), Washing- ton D.C., USA, 2009.
35. Gangemi,A.,Presutti,V.,OntologyDesignPatterns.In:HandbookonOntologies,Second edition, Staab, S., Studer, R. (Eds.), Springer, 2009, pp. 221 - 243.
36. Ontology Design Patterns . org (ODP). http:// ontologydesignpatterns.org (Accessed May 2017)
37. Gangemi, A., Gómez-Pérez, A., Presutti, V., & Suárez-Figueroa, M. C. (2007). Towards a catalog of owl-based ontology design patterns.
38. ISWC 2007 Introduction to the Semantic Web tutorial. Ontology Engineering Design Patterns. Aldo Gangemi http://kmi.open.ac.uk/events/iswc07-semantic-web-intro/pdf/5.%20Ontology%20Design.pdf (Accessed May 2017)
39. Schulz, S., & Martínez-Costa, C. (2013). How ontologies can improve semantic interoperability in health care. In Process Support and Knowledge Representation in Health Care (pp. 1-10). Springer International Publishing.
40. Schulz, S., Boeker, M., & Martinez-Costa, C. (2017). The BioTop Family of Upper Level Ontological Resources for Biomedicine. Studies in health technology and informatics, 235, 441.
41. Schulz, S., & Martínez-Costa, C. (2015). Harmonizing SNOMED CT with BioTopLite: An Exercise in Principled Ontology Alignment. Studies in health technology and informatics, 216, 832.
42. Blomqvist, E., Hitzler, P., Janowicz, K., Krisnadhi, A., Narock, T., & Solanki, M. (2016). Considerations regarding ontology design patterns. Semant. Web, 7(1), 1-7.

43. Horridge, M., Aranguren, M. E., Mortensen, J., Musen, M., & Noy, N. F. (2012, November). Ontology design pattern language expressivity requirements. In Proceedings of the 3rd International Conference on Ontology Patterns-Volume 929 (pp. 25-36). CEUR-WS. org.

44. Ontology patterns. A survey into their use. Technical report March 2014 http://kmi.open.ac.uk/publications/pdf/kmi-14-02.pdf (Accessed May 2017)

45. Daga E, Presutti V, Gangemi A, Salvati A. http://ontologydesignpatterns.org (Accessed May 2017)

46. Gangemi, A. (2005, November). Ontology design patterns for semantic web content. In International semantic web conference (pp. 262-276). Springer Berlin Heidelberg.

47. Aranguren, M. E., Antezana, E., Kuiper, M., & Stevens, R. (2008). Ontology Design Patterns for bio-ontologies: a case study on the Cell Cycle Ontology. BMC bioinformatics, 9(5), 1.

48. Rodrıguez-Doncel, V., Krisnadhi, A. A., Hitzler, P., Cheatham, M., Karima, N., & Amini, R. (2015, October). Pattern-based linked data publication: The linked chess dataset case. In Proceedings of the 6th International Workshop on Consuming Linked Data co-located with 14th International Semantic Web Conference (ISWC 2105), Bethlehem, Pennsylvania, US.

49. Prud'hommeaux, E., Labra Gayo, J. E., & Solbrig, H. (2014, September). Shape expressions: an RDF validation and transformation language. In Proceedings of the 10th International Conference on Semantic Systems (pp. 32-40). ACM.

50. Shapes Constraint Language (SHACL) W3C Candidate Recommendation 11 April 2017 http://www.w3.org/TR/shacl/#ClosedConstraintComponent (Accessed May 2017)

51. RDF Data Shapes Working Group Charter http://www.w3.org/2014/data-shapes/charter (Accessed May 2017)

52. Bosch, T., Nolle, A., Acar, E., & Eckert, K. RDF Validation Requirements-Evaluation and Logical Underpinning. Computing Research Repository (CoRR), abs/1501.03933, 2015.

53. Bosch, T., & Eckert, K. (2014, October). Requirements on RDF constraint formulation and validation. In: Proceedings of the 2014 International Conference on Dublin Core and Metadata Applications (pp. 95-108). Dublin Core Metadata Initiative.

54. Baader, F.; Calvanese, D.; McGuinness, D. L.; Nardi, D.; and Patel-Schneider, P. F., eds. 2010. The Description Logic Handbook: Theory, implementation, and applications. Cambridge University Press, 2nd edition.

55. J. Tao, E. Sirin, J. Bao, and D. L. McGuinness. Integrity constraints in OWL. In Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI-10). AAAI, 2010.

56. Ait-Ameur, Y., Baron, M., Bellatreche, L., Jean, S., & Sardet, E. (2015). Ontologies in engineering: the OntoDB/OntoQL platform. Soft Computing, 1-21.

57. Patel-Schneider, P. F. (2014). Using description logics for RDF constraint checking and closed-world recognition. arXiv preprint arXiv:1411.4156.

58. Gayo, J. E. L., & Rodrıguez, J. M. A. (2013, September). Validating statistical index data represented in RDF using SPARQL queries. In RDF Validation Workshop. Practical Assurances for Quality RDF Data, Cambridge, Ma, Boston.

59. Knublauch. H. SPIN - Modeling Vocabulary. http: //www.w3.org/Submission/spin-modeling/, 2011. (Accessed May 2017)

60. Clark&Parsia. 2014. Stardog integrity constraint validation. http://docs.stardog.com/icv. (Accessed May 2017)

61. A. G. Ryman, A. L. Hors, and S. Speicher. OSLC resource shape: A language for defining constraints on linked data. In C. Bizer, T. Heath, T. Berners-Lee, M. Hausenblas, and S. Auer, editors, Linked data on the Web, volume 996 of CEUR Workshop Proceedings. CEUR-WS.org, 2013.

62. Gayo, J. E. L., Prud'hommeaux, E., Boneva, I., & Solbrig, H. Validating and Describing Linked Data Portals using Shapes.
63. RDF and Linked Data Validation - ESWC'16 Tutorial http://es.slideshare.net/jelabra/rdf-validation-tutorial?next_slideshow=1 (Accessed May 2017)
64. What's exciting about SHACL - RDF Data Shapes. Semantic Arts. http://www.semanticarts.com/blog/rdf-shapes/ (Accessed May 2017)
65. Turtle – Terse RDF Triple Language W3C 2011. http://www.w3.org/TeamSubmission/turtle/ (Accessed May 2017)
66. SNOMED CT Expression Constraint Language Specification and Guide v1. 2015. http://doc.ihtsdo.org/download/doc_ExpressionConstraintLanguageSpecificationAndGuide_Current-en-US_INT_20150820.pdf (Accessed May 2017)
67. OWL 2 Web Ontology Language Manchester Syntax (Second Edition) 2012. http://www.w3.org/TR/owl2-manchester-syntax/ (Accessed May 2017)
68. TopBraid SHACL API: https://github.com/TopQuadrant/shacl (Accessed May 2017)
69. Sundvall, E., Qamar, R., Nyström, M., Forss, M., Petersson, H., Karlsson, D., ... & Rector, A. (2008). Integration of tools for binding archetypes to SNOMED CT. BMC Medical Informatics and Decision Making, 8(1), S7.
70. Qamar, R., & Rector, A. (2007, March). Semantic mapping of clinical model data to biomedical terminologies to facilitate data interoperability. In HealthCare Computing Conference.
71. Yu, S., Berry, D., Bisbal, J.:An Investigation of Semantic Links to Archetypes in an External Clinical Terminology through the Construction of Terminological "Shadows". IADIS 2010 Freiburg, Germany
72. Markwell D, Sato L, Cheetham E: Representing clinical information using SNOMED Clinical Terms with different structural information models. [http://www.kr-med.org/] proceedings of KR-MED May 31 – June 2 2008; Phoenix, Arizona, USA 2008.
73. IHTSDO terminology binding proposal. http://snomed.elearningserver.com/ContentRegistration.aspx?DocumentID=d18e65da-c42a-470e-8384-f6ddfc47fc04&Cust=34601&ReturnUrl=/p/3460133013 (Accessed May 2017)
74. Martinez-Costa, C., & Schulz, S. (2017). HL7 FHIR: Ontological Reinterpretation of Medication Resources. Studies in health technology and informatics, 235, 451.
75. SHACL Use Cases and Requirements: https://w3c.github.io/data-shapes/data-shapes-ucr/#uc28-self-describing-linked-data-resources. (Accessed May 2017)