A Lightweight Acquisition of Expert Rules for Interoperable Clinical Decision Support Systems

Bernardo Cánovas-Segura[a,*], Antonio Morales[a], Jose M. Juarez[a], Manuel Campos[a], Francisco Palacios[b]

[a] *Computer Science Faculty, University of Murcia, Spain*
[b] *University Hospital of Getafe, Madrid, Spain*

Abstract

Background: The process of adding new knowledge in the form of rules to already running Clinical Decision Support Systems (CDSSs) in hospitals is extremely costly and time consuming. There are two principal limitations: (1) the lack of a broad consensus regarding a uniform representation of clinical rules; and (2) the integration of new rule-based knowledge into hospital information systems.

Objective: To provide a guideline with which to support knowledge acquisition for rule-based CDSSs and to facilitate the integration of that knowledge into hospital datasets using standard clinical terminologies and ontologies as reference elements.

Materials and methods: We have designed a straightforward 4-step methodology with which to incorporate the external knowledge sources and data integration required to run CDSSs in hospitals. This lightweight methodology is based on a reference ontology that integrates standard clinical terminologies and its objective is to effectively acquire procedural knowledge in the form of rules.

Results: We have applied the methodology in the context of antimicrobial stewardship at a hospital. Recommendations from the European Committee on Antimicrobial Susceptibility Testing (EUCAST) were added to WASPSS, a CDSS running at the hospital. The reference ontology combines a subset of ATC terminologies for antibiotics and those of NCBI for microorganisms, including 584 and 1,714 concepts, respectively. A total of 94 new rules were added to the CDSS so as to represent EUCAST knowledge. We also evaluated different implementations in order to study their scalability, during which time we analysed Drools 7.5 as a production rule engine, HermiT as an ontology reasoner and RuQAR as an integration tool. Our experiments show that the combination of a production rule engine and an ontology reasoner in runtime is

*✉Corresponding author

Email addresses: `bernardocs@um.es` (Bernardo Cánovas-Segura), `morales@um.es` (Antonio Morales), `jmjuarez@um.es` (Jose M. Juarez), `manuelcampos@um.es` (Manuel Campos), `francisco.palacios@salud.madrid.org` (Francisco Palacios)

more efficient than using a single rule engine with a knowledge base derived from the reference ontology (1.9 times faster than the next approach when executing 1,000 expert rules on an ontology of 1,000 concepts).

Discussion: The methodology proposed helped to implement the knowledge acquisition process of EUCAST rules in a running CDSS. This methodology is applicable to other clinical domains when knowledge can be modelled with rules. Since it is a lightweight methodology, different implementation strategies are possible. The use of clinical standards also facilitates the future interoperation between CDSSs, particularly when using SNOMED as a reference ontology and employing future rule-sharing standards.

Keywords: Clinical decision support systems, Rule-based systems, Knowledge acquisition, Semantic interoperability, Antimicrobial Susceptibility Testing

1. Introduction

Modern clinical decision support systems (CDSS) have the objective of providing healthcare professionals with relevant knowledge [1, 2]. The CDSSs running in clinical institutions are currently having a growing impact on patients' healthcare and these systems are, therefore, under strict supervision and are strongly constrained. For example, if a CDSS is part of the clinical activity flow, it must be integrated into the hospital's Health Information System and other databases following local regulations. These technical requirements signify that CDSS decisions are often computed using the production rules paradigm, which is considered an efficient, scalable and mature technology.

CDSSs are, at present, specifically adapted to the requirements of each hospital. However, the acquisition of new knowledge may have a positive effect on the quality of the system outcome. Indeed, physicians in daily practice share strategies and protocols published by high quality-tested recommenders, such as international healthcare institutions or national health systems. For example, European health institutions have recently published a catalogue of rules (EUCAST expert rules) in order to assist microbiologists during the tests carried out to evaluate the clinical success of an antibiotic against an infection [3]. Examples of these kinds of rules are (Rule 1.3) *"IF the microorganism belongs to the Enterobacter cloacae species, THEN report as resistant to Amoxicillin-clavulanate"*, or (Rule 13.5) *"IF an Enterobacteriaceae is resistant to ciprofloxacin, THEN report as resistant to all fluoroquinolones"*.

We essentially identify two reasons why new knowledge is acquired by running CDSSs in hospitals: to interoperate with other CDSSs and to incorporate knowledge from specialised literature.

The interoperability between CDSSs, which is understood as the ability of systems to exchange interpretable data, is not a simple issue and, from the computational point of view, knowledge sharing is perhaps the most common bottleneck. In our opinion, there are two general strategies in literature by which to approach the mechanism of knowledge communication between systems: the bottom-up and the top-down strategies.

The first strategy in the process of CDSS interoperation (top-down) is to make all hospitals to share a single ontology, denominate as the global ontology, in order to define concepts and procedural knowledge (such as rules). This is a well-known approach in the Semantic Web community, in which the global ontology provides a homogeneous vocabulary with which to link terms, along with the axioms and relations between them [4]. However, when CDSSs are already in use in hospitals, the top-down approach implies a high-cost design. This approach requires the development of a global ontology in order to cover the terms employed at all hospitals and the redesign of each current knowledge base so as to adapt the rules to this ontology.

The objective of the bottom-up strategy is to minimise changes in the current representation of information in the CDSSs. From the point of view of medical informatics, this approach is considered to be a general interoperability problem. The common solution is to use a standard clinical model (reference model) to enable the use of communication mechanisms between the CDSSs. That is, each system maps its current terminology onto the reference model (e.g. SNOMED) to be shared.

Unlike CDSS interoperability, the acquisition of knowledge extracted from specialised literature does not require the development of technology for information interchange. Traditional approaches require a manual acquisition process carried out by knowledge engineers and validated by specialists. Human intervention is an advantage when different levels of granularity exist between the knowledge available in the sources (literature) and the rules and databases of the CDSS. This approach lacks standard mechanisms. However, the use of automatic or semi-automatic mechanisms could help to save costs and time.

To continue with the EUCAST example, the CDSS knowledge base stores specific treatments (e.g. *ciprofloxacin*, an antimicrobial) while the rule *"IF an Enterobacteriaceae is resistant to ciprofloxacin, THEN report as resistant to all fluoroquinolones"* is also defined over more general terms, such as *Enterobacteriaceae*, which is a large family of bacteria and *fluoroquinolones*, which is the group of antibiotics to which ciprofloxacin belongs.

Despite their differences, knowledge acquisition and CDSS interoperability approaches have to deal with similar problems: (1) the convenience of using ontologies to model new knowledge; (2) the need to extend the reasoning capacities of CDSS to support this new knowledge; and (3) the need to integrate CDSS reasoning into hospital information systems.

Various methodologies with which to extract and acquire knowledge have appeared in recent medical literature [5, 6] and some proposals suggest the use of theoretical models to interchange knowledge between rules. However, little attention is paid to approaching both problems from a holistic perspective in practical scenarios.

In this work, we propose a straightforward-4-step methodology with which to import knowledge from an external source in a rule-based CDSS. This methodology is suitable for acquiring the knowledge extracted from specialised literature but is also an essential step as regards enabling future CDSS interoperability.

The contributions of this paper are:

- A review of the essential principles of clinical production rules and the categorization of recent efforts concerning clinical knowledge representation in perspective (Sections 2 and 6).

- A novel lightweight methodology based on a reference ontology (REO) to support the acquisition of new knowledge in rule-based CDSSs (Section 3).

- The evaluation of our proposal, which was carried out by applying the methodology to a running CDSS for the antimicrobial testing problem and the realization of scalability experiments. This is, to the best of our knowledge, the first methodological approach to incorporate EUCAST rules into a CDSS (Sections 4 and 5).

2. Background

In this section, we review the essential components of production rule systems, the principal approaches related to clinical rule representation, the technologies employed to integrate production rules and semantic technologies and, finally, current standard clinical terminologies.

2.1. Production rules systems

Rule-based systems (RBSs) have been used as the basis for many CDSSs. These systems make it possible to model clinical knowledge by using sets of rules, also called production rules, in the form of *IF-THEN* statements. When the conditions in the *IF* part are met, the system executes the actions indicated by the *THEN* part. These two parts are also called the left-hand side (LHS) and the right-hand side (RHS) of the rule, respectively.

The set of rules contained in an RBS is commonly known as a *knowledge base*. The data currently available in the database that is relevant for the problem to be solved (facts) is loaded into a memory area called the *working memory.* Depending on the facts available there, the RBS selects one rule from the knowledge base to be fired in order to generate new facts or execute the corresponding actions.

Two kind of reasoning mechanisms are available in RBSs. On the one hand, in *forward chaining* the LHS of the rules is checked against the available facts, and the RHS is launched when all the requirements in the LHS are met. On the other hand, in *backward chaining* the reasoning starts with a list of goals or hypothesis, present in the RHS of any rule, and the LHS requirements are searched for among the available facts and the RHS of other rules. In clinical decision support tasks, forward chaining is the most extended mechanisms for prognosis, monitoring and control, while backward chaining is most frequently used in diagnosis problems.

CLIPS [7], Jess [8], and Drools [9] are examples of production rules engines that are widely used in industry and research. CLIPS, written in C, was developed by NASA in 1984 and has been used in many industrial applications,

```
rule "Unique name of the rule"
@...(...) // Rule meta-data
when // IF ...
  // Patterns to find within the working memory (LHS)
  ...
then // THEN ...
  // Actions, written in Java, to perform when the rule is fired (RHS)
  ...
end
```

Figure 1: Example of syntax used by Drools to represent IF-THEN rules.

along with some medical expert systems such as Germwatcher [10]. Jess, developed in 1995 and written in Java, is based on CLIPS and has been used to integrate clinical guidelines into CDSSs [11]. Finally, Drools, also written in Java, was developed in 2001 and acquired by Red Hat in 2006. It has been used in various clinical applications, such as an expert system for the clinical monitoring of anaesthesia [12] and a telecardiology decision support system [13]. These systems interpret different syntaxes in order to code the "IF-THEN" rules used in common language. For example, Figure 1 depicts the syntax followed by Drools to represent this kind of knowledge.

2.2. Knowledge exchange and the semantic approach

The problem of exchanging knowledge and rules has been a key topic since the beginning of CDSS research.

A rule-based language, called Arden syntax, was specifically developed in order to share medical knowledge bases [14, 15]. It includes statements such as database queries, IF-THEN blocks, and actions that can provide physicians with alerts. However, the mapping problem persists, since the data elements have to be mapped onto the database employed in each institution before using a shared module.

A milestone as regards sharing interpretable knowledge is the Sematic Web proposal of [4]. Berners-Lee and colleagues proposed to incorporate semantic meaning into the data available in the World Wide Web. This would allow software agents to understand relationships between data and obtain meaningful results in their searches. Ontologies, as a basic component of the Semantic Web, would provide a homogeneous vocabulary with which to link all these data. The language most frequently used to define ontologies is currently OWL [16]. OWL, which is based on description logic, allows the definition of *classes* with which to model the concepts in a specific domain, *individuals* to model the objects in that domain, *object properties* to represent relationships between individuals and *datatype properties* to represent relationships between individuals and literals. Each element has a unique identifier (IRI) within the ontology. Furthermore, OWL identifies two kinds of statements: the *TBox* statements, which contain axioms regarding classes and properties, and the *ABox* statements, which include axioms related to individuals.

The inference of new relationships between individuals and classes is possible owing to the semantics of the available assertions. To continue with the EUCAST example from Section 1, if the *Enterobacteriaceae* family is a subclass of *Gram-negative* and the concept *Escherichia coli* is a subclass of *Enterobacteriaceae*, we can infer that it is also a subclass of *Gram-negative* thanks to the transitivity property of the *isSubclassOf* relationship.

These inferences are computed by specific programs, denominated as *semantic reasoners*, such as Pellet [17] and HermiT [18]. These reasoners can also identify other semantic problems owing to the expressiveness of OWL language, such as inconsistency or undecidability, which may lead to an ontology that is, from a practical point of view, useless.

2.3. Rules and semantics

Since ontologies are conceptual models, they are not intended to be used for the same reasoning processes as production rules. Many approaches with which to incorporate rule-based knowledge into ontologies have, therefore, appeared: The Semantic Web Rule Language (SWRL) [19] extends the OWL axioms to allow the definition of IF-THEN rules that can be stored within the ontology. The Rule Interchange Format (RIF) [20] was developed as part of the Semantic Web infrastructure to provide a core dialect for the definition of rules that could be translated into other rule languages. Finally, the SPARQL Inferencing Notation (SPIN) [21] and the Shapes Constraint Language (SHACL) [22], SPIN's successor, allow the representation of rules and constraints using the SPARQL query language, which is widely used by the Semantic Web community.

Some works are, in contrast, focused on extending the available rule engines to make use of the existing OWL ontologies without incorporating further notations or formats. Some examples of this are: CLIPS-OWL [23], which generates a CLIPS knowledge base from an OWL ontology; Thea [24], which makes it possible to work with OWL ontologies from Prolog; DLEJena, [25] which relies on Pellet to perform TBox reasoning, and on Jena to execute ABox-related rules, and RuQAR [26], which uses HermiT to perform TBox reasoning, generate a set of rules and facts in an intermediate language and translate them into different rule engine languages, such as those used by Jess or Drools.

2.4. Standard Clinical Terminology

There are many ontologies and terminologies with which to formalise clinical concepts. In this work we focus on three of the most relevant ones: SNOMED, ATC and NCBI taxonomy.

The Systematized Nomenclature of Medicine Clinical Terms (SNOMED) is an exhaustive general terminology of clinical terms with multilingual support. SNOMED is principally used as a reference terminology to ease system interoperation and processable clinical content. From the computational point of view, SNOMED is organised in overlapped hierarchies of clinical concepts. SNOMED describes unique clinical concepts (over 30,000 terms), and the relations between them, with several descriptions (over 728,000). SNOMED is mapped with different standard terminologies, such as ICD-10 in its last release (v. 1.37.2 issued in January 2018), and other integrations are currently under development (e.g. LOINC-SNOMED CT cooperation project).

In pharmacy, one of the terminologies most frequently used is the Anatomical Therapeutic Chemical (ATC) classification system [27]. In this ontology, the active substances of drugs are classified in different levels according to their anatomical/pharmacological group (first level), therapeutic/pharmacological groups (second level) and their chemical, pharmacological or therapeutic subgroups (third and fourth levels). The most specific concept level (fifth level) is the chemical substance. Each term has a code, which is obtained by aggregating the codes of its parent terms and a unique identifier within its level. For example, *ciprofloxacin* has the ATC code J01MA02 because it belongs to the groups *Antiinfectives for systemic use* (J), *Antibacterials for systemic use* (01), *Quinolone antibacterials* (M) and *Fluoroquinolones* (A) and its identifier, which is included at the end (02).

With regard to terminologies for microorganisms, they are usually based on the common ranks used in zoology, such as *Family*, *Genus* and *Species*, which classify organisms according to their similarities or their capacity to produce fertile offspring. However, these criteria are difficult to observe in the case of microorganisms [28]. The NCBI taxonomy [29] is a curated classification of organisms based on their genetic similarities yet relying on the current taxonomic literature. It is being used to index the genetic sequences available in public databases and is in a state of continuous growth and revision.

3. Methodology Proposed

In this work, we propose a straightforward lightweight methodology with which to acquire new clinical rules for CDSSs from clinical knowledge sources. This methodology assumes that reasoning is based on production rules and that existing CDSSs are already running and integrated into the Health Information Systems (HIS) of the hospital. Changes made to the reasoning mechanisms and knowledge representation must, therefore, be minimised.

This methodology is driven by the reference ontology (REO). This ontology is essentially the reuse of standard clinical terminologies. REO is used to represent the minimal set of concepts and relations from the knowledge sources needed to update the knowledge base with new production rules. The use of REO has to deal with: the lack of ontology inference mechanisms for production rule systems and the mapping with local terms from HIS databases.

The REO methodology consists of 4 steps: REO definition, mapping local terms, extending reasoning properties and implementing new rules. Figure 2 summarises the REO methodology and we explain these steps in what remains of this section.

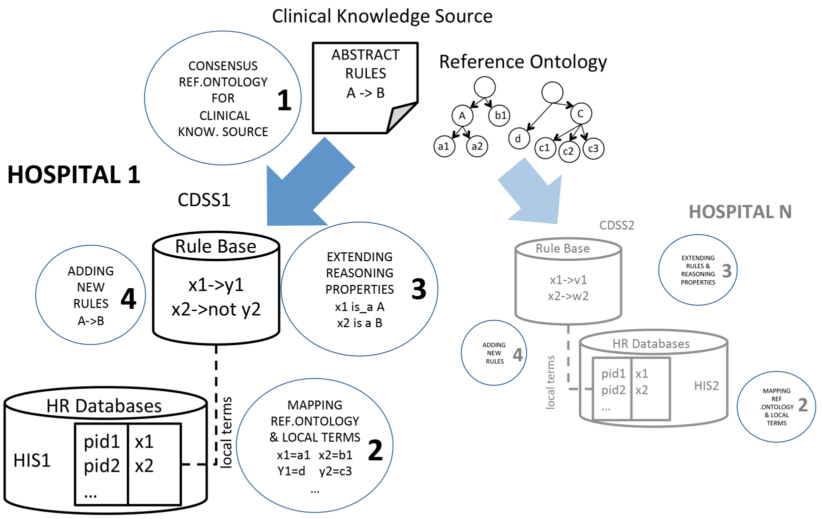


Figure 2: Four-step REO methodology: (1) REO definition, (2) mapping local terms, (3) extending reasoning properties; and (4) implementing new production rules.

3.1. Step 1: Define the Reference Ontology

The first step is to define the Reference Ontology (REO). The purpose of the REO is to represent concepts, relations and properties extracted from the clinical knowledge sources to be used as a reference for rule definition.

The REO has to be designed in accordance with the following criteria:

- Ontology consensus: the REO is obtained by means of consensus between the clinicians and knowledge engineers, who are responsible for validation and technical details, respectively.

- Concept granularity: the REO represents concepts in different levels of abstraction, considering the definitions of the clinical knowledge of sources and the local terminology used in HIS databases.

- Minimal coverage: the REO includes the minimum number of concepts and relations necessary to express the elements of the knowledge sources used in the new rules. This decision is biased by the need to save design time, costs and computational efficiency.

- Standards: the REO is a subset of one or several standard clinical terminologies (e.g. ICD, SNOMED, LOINC, ATC, etc.). Although the definition of a complete new ontology might be considered in reduced clinical scenarios, the reuse of high quality specific terminologies is recommended for the sake of extensibility and interoperability.

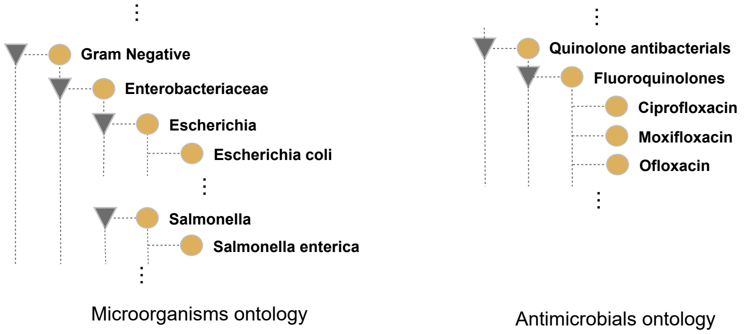


Figure 3: Example of REOs with which to implement EUCAST rules in a CDSS. *Enterobacteriaceae*, *Fluoroquinolones* and *Ciprofloxacin* are mandatory because they are required for rule definition. The other concepts are commonly used in our CDSS and are included to facilitate curation and matching with local codes.

- T-Box: The reference ontology contains TBox elements only. That is, it must have concepts and relationships, but it must be empty of axioms regarding individuals (concept instances).

In practice, the clinical context or the health-care stakeholders may suggest this REO ontology. For instance, healthcare organisms might encourage the use of SNOMED-CT [30, 31] in order to facilitate data gathering and interoperability with external systems. In other situations, there are many public repositories (e.g. OBOFoundry [32]) with available ontologies that can be reused, pruned and/or adapted to the context required. Finally, several tools with which to extract ontological relationships from a database schema and/or data are available [33–35], whose results can be used as a starting point for the definitive ontology.

Example: According to the EUCAST guiding example, the general concepts of *Enterobacteriaceae* and *Fluoroquinolones* must exist in our reference ontology in order to allow the definition of Rule 13.5. Other more specific concepts must also be present, since they are part of other rules or in order to facilitate subsequent matching with the concepts used by the system. For example, *Ciprofloxain* is a kind of *Fluoroquinolones* that must be present because it is needed to define Rule 13.5.*Escherichia coli* and *Salmonella enterica* are, meanwhile, present in the ontology as child concepts of *Enterobateriaceae* because they are common results in microbiology tests, yet no rule refers to them specifically. For similar reasons, the antimicrobials *Moxifloxacin* and *Ofloxacin* are present as children of the *Fluoroquinolones* concept. A graphical representation of this example of a REO is shown in Figure 3.

3.2. Step 2: Mapping the local terms

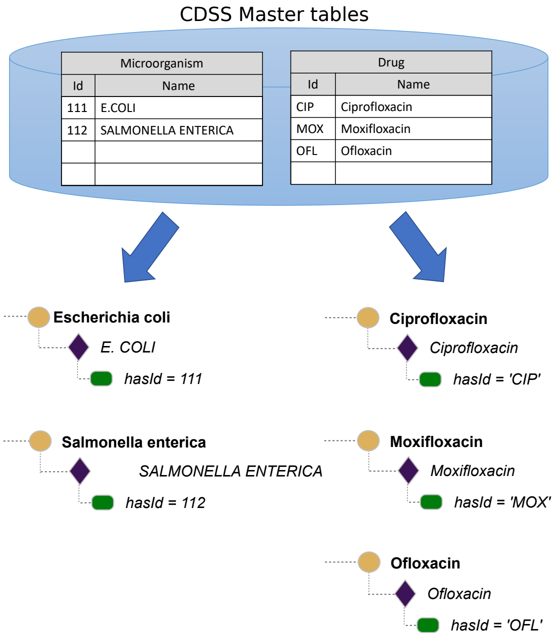
The goal of the second step is to allow the REO individuals to be added to the knowledge base of the CDSS. If a new CDSS is created, this step means the development of an ontology-based CDSS. However, in this methodology we assume that a rule-based CDSS is already in use. In this case, CDSS rules are often linked to terms in hospital databases and a mapping process is, therefore, needed between the REO concepts and the local database terms. This methodology has the following criteria:

- Standard local terminologies: If the local databases use standard clinical terminologies, it might be convenient to incorporate these terminologies into the REO (Step 1).
- Specific mapping: The terms from the local databases are mapped onto the most specific concept of the REO, that is, the lowest term in the hierarchy.
- ABox: Mapped terms are asserted, as individuals, in the ABox of the REO.
- Partial mapping: Not all concepts and terms are mapped. Those concepts of the highest level of abstraction are not mapped onto database elements, but they might be inferred in ontology subsumption reasoning. Database terms might not be mapped onto REO concepts when they are not directly involved in the topic of the clinical knowledge source.

From the practical point of view, concept-term mapping is not an immediate issue. Moreover, the ontology instances that are asserted should include the primary key values used to link the database terms. This can be achieved by using different approaches, such as employing data type properties or including these data in their IRI. The goal is to facilitate their retrieval during the next steps of this methodology.

It may occur that some local terms cannot be mapped with any concept. In these cases, we might consider going back to Step 1 and revising the ontology in order to include the missing concepts. However, when the terms without a match are of no interest as regards our problem, we can ignore them or map them with a generic concept to ensure the entire mapping between data and ontology. For instance, if our global rules are focused on bacterial resistance, we can ignore terms regarding other kinds of organisms (viruses, fungi, etc.) or simply map them with the *Other organisms* concept.

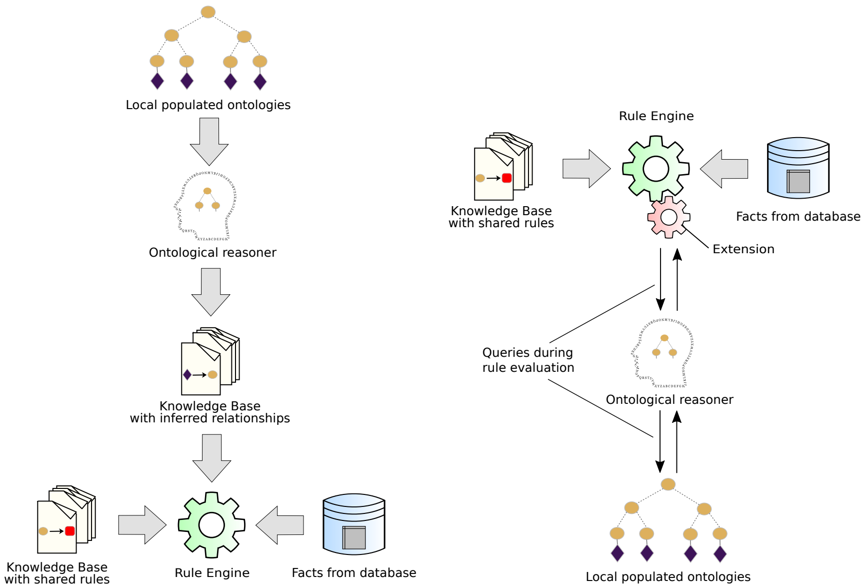
Example: To continue with the guiding example, hospital databases include a list of local terms, denominated as master tables. If the master table of microorganisms in the HIS contains a term entry *E.COLI*, with an identifier "111", an individual with the name *E.COLI* belonging to the concept *Escherichia coli* is added to the ontology. The identifier can be included as the value of a data property (e.g. *hasId*). The same approach can be employed to map antimicrobials, as depicted in Figure 4.



Microorganism	
Id	Name
111	E.COLI
112	SALMONELLA ENTERICA

Drug	
Id	Name
CIP	Ciprofloxacin
MOX	Moxifloxacin
OFL	Ofloxacin

Figure 4: Example of mapping between local concepts and REO terms by asserting individuals into the ontology. The local id is preserved in a data property in order to find the concept easily.



(a) (b)

Figure 5: Strategies employed to extend reasoning properties: (a) Use an ontological reasoner to generate a knowledge base with all the inferred types and rules; or (b) extend the production rules engine to query the ontological reasoner during rule evaluation.

3.3. Step 3: Extending reasoning properties

Once the REO has been defined (Step 1) and its instances have been mapped and inserted into the knowledge base (Step 2), the aim of Step 3 is to extend the reasoning properties of the production rule engine of the CDSS with ontology reasoning.

This issue can be addressed by following two strategies: simulating ontology reasoning with rules or using an external ontological reasoner.

3.3.1. Step 3.a: Ontology reasoning simulation

When employing the first strategy, new production rules are added to the knowledge base. In this case, ontology reasoning mechanisms must be identified beforehand (see Figure 5.a). In practice, this strategy assumes essential abduction methods, such as: A is_a B , so if x_A is of type A, then x_A is also of type B. In these cases, new rules are added to the knowledge base, simulating this type of inferences.

The ontological reasoner is first used to infer all the concepts and relationships related to individuals. A knowledge base for the target rule engine is then created, which contains all the inferred data. This knowledge base will be used as a basis on which to define new rules that require ontological concepts.

For each individual in the REO, a rule is generated to assign the fact that the individual represents to all the inferred concepts to which it belongs, which are also declared in the knowledge base. This process can be automatised, since all the information required (facts IDs, etc.) is available in the REO.

This knowledge base can be used as a starting point to define any rule querying for the ontological concepts while working with the local terminology.

A similar approach is followed by some tools that translate OWL ontologies into knowledge bases for specific production rule systems, such as CLIPS-OWL [23], which generates knowledge bases in the COOL language for CLIPS, or RuQAR [26], which performs a similar task for Jess and Drools.

Example: According to the EUCAST guiding example, we can generate Drools types and rules to link each concept with all its ontological inferred types. As shown in Figure 6, for each concept in the REO, we can declare a *trait*, which is a type that can be dynamically assigned to any fact. In its declaration, the *extends* clause allows us to indicate the antecedents of each concept. Moreover, a rule is created for each individual in order to assign the aforementioned types to the local concept (by using the *don* operator) when it is present in the working memory. Since the id of the local concept is included within the ontology, this code can be generated automatically.

3.3.2. Step 3.b: External ontological reasoner

When using the second strategy, the rule engine is extended to query an ontological reasoner concerning the relationships and concepts available in the REO (see Figure 5.b).

When the evaluation of the LHS of a rule makes it necessary to check whether a specific fact is related to a concept, the ontological reasoner looks for that fact within the ABox and infers all the related concepts. If the concepts found contain that which is required, the rule can be fired.

Unlike the simulation strategy, the use of an ontological reasoner has a minor impact on rule definitions but requires the availability of a CDSS in order to integrate external reasoner engines. That is, the CDSS implementation requires the capacity to create plugins or extensions to the rules engine, which is not possible in all cases. Examples of implementations exploiting this approach have been developed for Drools [36], which allows the definition of new LHS operators, and for Prolog [37].

Example: A new operator, called *isOfType* can be declared in Drools as a binary operator. The first argument will be a *Microorganism* or *Drug* type fact, and the second argument will be a string representing the name of the concept in the ontology. The operator will perform the following steps: i) Recover the Id from the local term passed as first parameter; ii) Look for the individual within the ontology with this Id; iii) Query the ontological reasoner to infer all the types for the individual; iv) Check whether the string passed as a second argument coincides with the description of any of the inferred types. This signifies that if the operator is called passing a microorganism with *Id=111* and the string *"Enterobacteriaceae"* as arguments, it will recover the *E.COLI* individual, infer all its types, find the *Enterobacteriaceae* concept among them and return *TRUE*.

```
declare trait Enterobacteriaceae  extends Gram_Negative end
declare trait Escherichia_coli    extends Enterobacteriaceae end
declare trait Salmonella_enterica extends Enterobacteriaceae end

. . .

declare trait Fluoroquinolones extends Quinilone_antibacterials end
declare trait Ciprofloxacin    extends Fluoroquinolones end
declare trait Ofloxacin        extends Fluoroquinolones end
declare trait Moxifloxacin     extends Fluoroquinolones end

. . .

rule "Classify microorganism 111-E. COLI"
when
  $micro : Microorganism (id == 111)
then
  don( $micro, Escherichia_coli.class);
end

rule "Classify microorganism 112-SALMONELLA ENTERICA"
when
  $micro : Microorganism (id == 112)
then
  don( $micro, Salmonella_enterica.class);
end
. . .
rule "Classify antimicrobial CIP-Ciprofloxacin"
when
  $drug : Drug (id == "CIP")
then
  don( $drug, Ciprofloxacin.class);
end

rule "Classify antimicrobial OFL-Ofloxacin"
when
  $drug : Drug (id == "OFL")
then
  don( $drug, Ofloxacin.class);
end

rule "Classify antimicrobial MOX-Moxifloxacin"
when
  $drug : Drug (id == "MOX")
then
  don( $drug, Moxifloxacin.class);
end
```

Figure 6: Example of types and rules generated from the REO elements in Figures 3 and 4 so as to be able to use them in Drools.

```
rule "Interpretive EUCAST Table 13 Rule 13.5"
  @id ("RPEUT13.R5")
  @altName("Interpretive rules for quinolones - Rule 13.5")
  @note1 ("Acquisition of at least two target mutations ...")
  @evidenceGrade("B")
when
  $c:     Culture(microorganism isOfType "Enterobacteriaceae")
  $astR1: TestResult (culture == $c,
    antimicrobial isOfType "Ciprofloxacin" , isResistant)
  $drug:  Drug(this isOfType "Fluoroquinolones")
then
  EucastInference r = new EucastInference(
    $c, 'RESISTANT', $drug, drools.getRule(),
    new Explanation($c.getMicroorganism(), "is a Enterobacteriaceae"),
    new Explanation($astR1, "indicates resistance to Ciprofloxacin"),
    new Explanation($drug, "is a fluoroquinolone antibacterial")
  );
  insert(r);
end
```

Figure 7: Example of EUCAST rule implemented thanks to the use of the custom operator *isOfType* and the concepts available in the REO ontology.

3.4. Step 4: Implementing new rules

Once the CDSS is able to reason with the REO instances, the knowledge extracted from the clinical sources in the form of production rules can be added to the knowledge base.

In this step, new concepts that could be ignored during Step 1 may be required when coding the final rules. In this case, it might be necessary to iterate in the methodology. Fortunately, thanks to the easy incorporation of knowledge, ontologies and RBSs, it does not overload the design process.

Example: According to the EUCAST example, Rule 13.5 is eventually implemented in the CDSS knowledge base using Drools, as shown in Figure 7. First, the rule includes meta-data coded after the symbol '@'. These data are used to code the explanation and evidence grade as indicated in the source document, thus making it accessible to the CDSS when required. The LHS searches for cultures that confirm the presence of an *Enterobacteriaceae* and its resistance to *ciprofloxacin* (in a linked *TestResult* fact). Moreover, the LHS matches with any drug belonging to the *fluoroquinolones* group. When fired, the rule creates a new fact indicating that the microorganism might also be resistant to the *fluoroquinolones* member retrieved. Additionally, the new fact includes an explanation which contains the EUCAST rule that has been launched along with its meta-data, and a textual explanation of the inferences performed for validation.

3.5. Running Example

In this section, we illustrate the effects of and changes to the REO methodology in a CDSS by means of the running example.

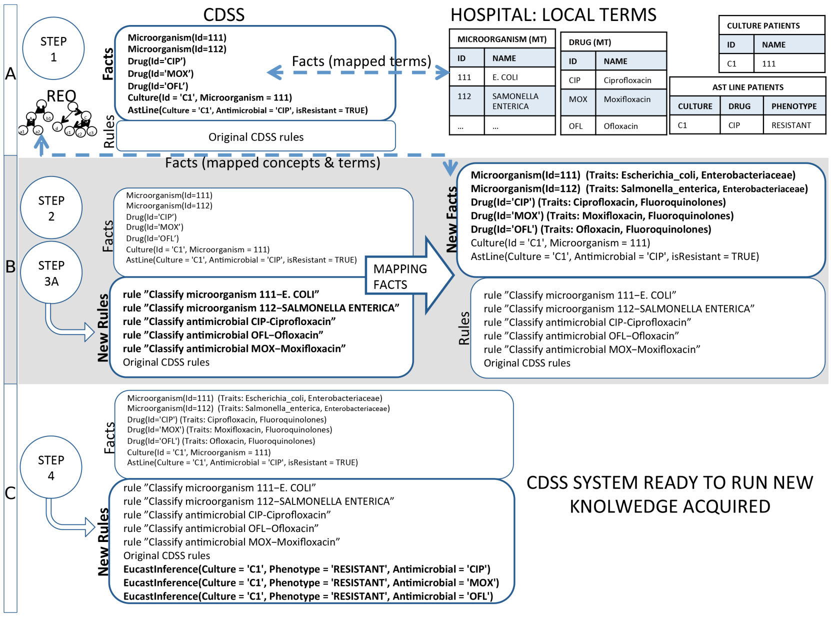

A
B
C
STEP 1
REO
CDSS
Facts
Microorganism(Id=111)
Microorganism(Id=112)
Drug(Id='CIP')
Drug(Id='MOX')
Drug(Id='OFL')
Culture(Id = 'C1', Microorganism = 111)
AstLine(Culture = 'C1', Antimicrobial = 'CIP', isResistant = TRUE)
Rules
Original CDSS rules
Facts (mapped terms)
HOSPITAL: LOCAL TERMS
MICROORGANISM (MT)
ID | NAME
111 | E. COLI
112 | SAMONELLA ENTERICA
... | ...
DRUG (MT)
ID | NAME
CIP | Ciprofloxacin
MOX | Moxifloxacin
OFL | Ofloxacin
CULTURE PATIENTS
ID | NAME
C1 | 111
AST LINE PATIENTS
CULTURE | DRUG | PHENOTYPE
C1 | CIP | RESISTANT
Facts (mapped concepts & terms)
STEP 2
STEP 3A
Facts
Microorganism(Id=111)
Microorganism(Id=112)
Drug(Id='CIP')
Drug(Id='MOX')
Drug(Id='OFL')
Culture(Id = 'C1', Microorganism = 111)
AstLine(Culture = 'C1', Antimicrobial = 'CIP', isResistant = TRUE)
New Rules
rule "Classify microorganism 111–E. COLI"
rule "Classify microorganism 112–SALMONELLA ENTERICA"
rule "Classify antimicrobial CIP-Ciprofloxacin"
rule "Classify antimicrobial OFL–Ofloxacin"
rule "Classify antimicrobial MOX–Moxifloxacin"
Original CDSS rules
MAPPING FACTS
New Facts
Microorganism(Id=111) (Traits: Escherichia_coli, Enterobacteriaceae)
Microorganism(Id=112) (Traits: Salmonella_enterica, Enterobacteriaceae)
Drug(Id='CIP') (Traits: Ciprofloxacin, Fluoroquinolones)
Drug(Id='MOX') (Traits: Moxifloxacin, Fluoroquinolones)
Drug(Id='OFL') (Traits: Ofloxacin, Fluoroquinolones)
Culture(Id = 'C1', Microorganism = 111)
AstLine(Culture = 'C1', Antimicrobial = 'CIP', isResistant = TRUE)
Rules
rule "Classify microorganism 111–E. COLI"
rule "Classify microorganism 112–SALMONELLA ENTERICA"
rule "Classify antimicrobial CIP-Ciprofloxacin"
rule "Classify antimicrobial OFL–Ofloxacin"
rule "Classify antimicrobial MOX–Moxifloxacin"
Original CDSS rules
STEP 4
Facts
Microorganism(Id=111) (Traits: Escherichia_coli, Enterobacteriaceae)
Microorganism(Id=112) (Traits: Salmonella_enterica, Enterobacteriaceae)
Drug(Id='CIP') (Traits: Ciprofloxacin, Fluoroquinolones)
Drug(Id='MOX') (Traits: Moxifloxacin, Fluoroquinolones)
Drug(Id='OFL') (Traits: Ofloxacin, Fluoroquinolones)
Culture(Id = 'C1', Microorganism = 111)
AstLine(Culture = 'C1', Antimicrobial = 'CIP', isResistant = TRUE)
New Rules
rule "Classify microorganism 111–E. COLI"
rule "Classify microorganism 112–SALMONELLA ENTERICA"
rule "Classify antimicrobial CIP-Ciprofloxacin"
rule "Classify antimicrobial OFL–Ofloxacin"
rule "Classify antimicrobial MOX–Moxifloxacin"
Original CDSS rules
EucastInference(Culture = 'C1', Phenotype = 'RESISTANT', Antimicrobial = 'CIP')
EucastInference(Culture = 'C1', Phenotype = 'RESISTANT', Antimicrobial = 'MOX')
EucastInference(Culture = 'C1', Phenotype = 'RESISTANT', Antimicrobial = 'OFL')
CDSS SYSTEM READY TO RUN NEW KNOLWEDGE ACQUIRED

Figure 8: Running example of REO methodology applied.

We assume that the rule-based CDSS is already running in a hospital. HIS local terms are stored in master tables (MT) and include microorganisms and drugs (Microorganism-MT and Drug-MT), along with other datasets regarding patients. The original CDSS knowledge base consists of a set of facts, currently linked to local terms, and a set of clinical rules (original CDSS rules). For example, Figure 8.A shows that an *Escherichia coli* microorganism is detected in a patient's culture. This microorganism fact is mapped onto the local term *111-E.COLI*.

Following the proposed methodology, step 1 is carried out to obtain the reference ontology (REO), establishing taxonomies of microorganisms and antimicrobials. Steps 2 and 3a are then followed, and new rules are added to the knowledge base. These rules allow ontological inferences to be run. For instance, the first rule in Figure 8.B maps a microorganism fact of local term *111-E.COLI* with the *Escherichia_coli* concept of the REO. Moreover, once the fact has been mapped onto the ontology, the engine can infer that this fact is also an instance of the *Enterobacteriaceae* concept.

Finally, the outcome of step 4 is the set of rules that model the knowledge described in the EUCAST. These new rules are added to the knowledge base and the CDSS system is ready to run the knowledge acquired (Figure 8.C).

4. Use Case: Antibiotic Susceptibility Testing

4.1. Clinical context: Susceptibility and EUCAST rules

When an infection is diagnosed in hospitals, a sample is taken from the patient and then analysed in the microbiology laboratory in order to determine to which species the microorganism causing the infection belongs. An antimicrobial susceptibility test is also performed: the microorganism is exposed to different concentrations of a set of antimicrobials so as to study its reaction and estimate the outcome of clinical therapies. Depending on the antimicrobial concentration in which the microorganism dies or stops growing, it is reported as *resistant*, which indicates that a treatment with the antibiotic will not succeed in stopping the infection, or *susceptible*, which suggests the contrary. An *intermediate* susceptibility can also be reported, which suggests an intermediate or uncertain clinical effect of the antimicrobial.

The European Committee on Antimicrobial Susceptibility Testing (EUCAST) is a healthcare organism that deals with many aspects of antimicrobial susceptibility testing and harmonizes the guidelines from different European countries [38]. EUCAST additionally publishes a set of expert rules with the aim of assisting microbiologists with susceptibility test results. For example, the rules are used to suggest new (inferred) susceptibilities for agents that are actually non-tested or to recommend actions as regards reporting to clinicians. They are based on clinical and/or microbiological evidence, and some rules have an evidence degree, exceptions or comments attached to them. These meta-data play a key role when deciding on a treatment.

EUCAST expert rules are organized in tables, in which each row is considered as a rule. We use $X.Y$ to denote the rule in table X row Y. Tables 1 to 4 contain intrinsic resistance rules, while Tables 5 to 7 are exceptional phenotypes and Tables 8 to 13 include interpretative rules. They are defined mostly over species of bacteria (e.g. *IF Streptococcus pneumoniae is resistant to ... THEN ...*), but also over families, classes and other groups of bacteria. These rules could similarly be defined over a specific antimicrobial or over a group of them. Table 1 shows an example of EUCAST interpretive rules, from rule 13.1 to 13.5 (those used in the guiding example above), focused on a group of antimicrobials named *quinolones* [3].

4.2. Technological context

We have applied our methodology to a running CDSS denominated as the Wise Antimicrobial Sterwardship Program Support System (WASPSS) [39, 40]. The main objective of WASPSS is to provide support for antimicrobial stewardship at a hospital. The system uses Extract, Transform and Load (ETL) processes and an HL7 interface to gather information from other hospital systems and generate reports, alerts and a timeline view of patient's records by means of a web-based interface.

WASPSS is being used at the University Hospital of Getafe, Spain, and is currently being evaluated in another eight hospitals in Spain. Laboratory and pharmacy systems are different from one hospital to another and different

Rule no.	**Organism**	**Agents tested**	**Affected agents**	**Rule**	**Evidence grade**
13.1	Staphilococcus spp.	Ofloxacin, ciprofloxacin, levofloxacin, and moxifloxacin	Fluoroquinolones (all)	IF resistant to ofloxacin or ciprofloxacin, but not to levofloxacin or moxifloxacin, THEN report warning of risk for development of resistance during therapy with quinolones	C
13.2	Staphilococcus spp.	Levofloxacin and moxifloxacin	Fluoroquinolones (all)	IF resistant to levofloxacin or moxifloxacin, THEN report as resistant to all fluoroquinilones	C
13.3	Streptococcus pneumoniae	Ofloxacin, ciprofloxacin, levofloxacin and moxifloxacin	Fluoroquinolones (all)	IF resistant to ofloxacin or ciprofloxacin, but not to levofloxacin or moxifloxacin, THEN report warning that acquisition of a first-step mutation may lead to resistance development under therapy with other quinolones	C
13.4	Streptococcus pneumoniae	Levofloxacin and moxifloxacin	Fluoroquinolones (all)	IF resistant to levofloxacin or moxifloxacin, THEN report as resistant to al fluoroquinolones	B
13.5	Enterobacteriaceae	Ciprofloxacin	Fluoroquinolones (all)	IF resistant to ciprofloxacin, THEN report as resistant to all fluoroquinolones	B

Table 1: Example of interpretive rules of quinolones from EUCAST [3]. *Rule no.*: rule identifier of EUCAST; *Agent tested*: antimicrobials tested in laboratory; *Affected agents*: antimicrobials whose effects are inferred by the rule; *Rule*: description of the expert rule; and *Evidence Grade*: A, B or C, with A being the strongest evidence grade and C the least.

terminologies are, therefore, used for similar concepts. In order to incorporate global expert rules into WASPSS, we developed and tested the methodology described above.

WASPSS stores the information gathered in a database management system, and interfaces with Drools [9] to allow rule-based reasoning. This makes it possible for the system to use complex alerts, which are defined as rules by knowledge engineers and clinicians.

4.3. REO Methodology Implementation

The team in charge of the implementation of the methodology was composed of two knowledge engineers and the head of the Infection Control Committee at the collaborating hospital. We additionally used Protégé v.5.2.0 [41] for the revision and curation of the REO and its mappings.

4.3.1. Step 1: REO using ATC and NCBI

We first designed a REO according to the terms described in EUCAST [3]. These rules are defined over families and species of bacteria and different groups of antimicrobials, and an ontology with the complex taxonomies is, therefore, required.

Although there are taxonomies of bacteria, the discovery of new species and similarities at genome level among previously misclassified species leads to different classifications. In this case, we have chosen the public organisms taxonomy created and maintained by the NCBI [29]. The NCBI taxonomy currently contains a total of 552,750 terms. There are concepts for a wide range of organisms, from viruses to extinct species. Of these, 24,481 belong to the bacteria rank and 18,224 represent bacteria species. We pruned this ontology to those bacterial species used in clinical practice only, and added some extra concepts, such as *Gram-negative bacteria*, which is not included in the original taxonomy and is required by the EUCAST rules. Finally, our REO for microorganisms consists of 1,714 concepts, 804 of which belong to the microorganisms' species.

Another standard clinical terminology was needed for antimicrobials. We used the available ATC classification [27]. The original ATC taxonomy contains more than 1,900 pharmacological compounds, some of which are repeated when used for pathologies in different organs. For our REO, we selected only those in the *J - Antiinfectives for systemic use* main group: that containing the antimicrobials used in our setting. We additionally included some of the concepts required for EUCAST rule definitions, such as *Ureidopenicillins* and *Isoxazolyl penicillins.* Finally, 584 concepts were stored in our REO for antimicrobials, with 466 children concepts denoting specific antimicrobial compounds.

We decided to maintain some extra terms that were not directly related to EUCAST rules (e.g. viruses, fungi, parasites, antibiotics not used in the hospital, etc.) in search of a compromise between the minimum coverage objective and the availability of terms for future extensions. It took one week to code the scripts used to generate the OWL files from the source ontologies and prune them according to our requirements, and an extra week was needed to revise and correct the resulting ontology.

4.3.2. Step 2: Mapping

The master tables to be mapped with our REO belong to two different hospital systems: microbiology (microorganisms) and pharmacy (antimicrobials). However, both tables are integrated into WASPSS and we, therefore, have access to all of the terms required from the rules engine.

The microorganisms master table contains a numerical ID and the description of the infectious agent. In most cases, it is a bacteria species. However, a small number of elements describe other circumstances, such as “unidentified bacillus” or “Gram-positive cocci”, because they are used to report preliminary test results.

The mapping was carried out using a semi-automated procedure: first, an initial automatic mapping by name was performed, linking each local term to the ontology class with the same description; a manual mapping was then performed with the unclassified terms. Finally, 1,285 local terms were matched with ontological concepts, 221 of which were included in the “Not classified” class, including vague terms, obsolete IDS and other terms which needed a more thorough classification.

With regard to the pharmacy table, the WASPSS system already contained the ATC codes for most of the drugs available and the mapping could, therefore, be carried out by means of a direct code match. The exceptions were some mixtures, such as decontamination products used in the intensive care unit. Finally, a total of 243 local terms were mapped onto concepts, 11 of which were included in the *Other antimicrobials* class.

It took two days to code the scripts required to perform the automatic matching of terms for both ontologies, and an extra week to tune the resulting mapping manually.

4.3.3. Step 3: production rules and ontologies

Drools, the RBS used in WASPSS, allows user-defined operators that can be included in the LHS of a rule. We were, therefore, able to test both the approaches for firing production rules based on ontological concepts suggested in Step 3:

- Step 3 - Option a: Ontology reasoning simulation. The classes present in the ontology were included as new types in Drools, and the local asserted terms were inserted as facts. A rule was created for each fact, linking it with the types inferred by the ontological reasoner. A total of 2,298 new types were generated to denote each REO concept, and 1,528 new rules were created to link each local term to them. This task was performed using the *traits* capability from Drools, which allows dynamic multi-hierarchical typing, as shown in previous examples. It took three days to code and test the scripts which were employed to navigate the REO and generate the Drools code according to it. It is also possible to

	EUCAST Table	Available	Fully implemented	Partially implemented	Not implemented
Intrinsic resistance rules[1]	1	18	18	0	0
	2	7	7	0	0
	3	5	5	0	0
	4	14	14	0	0
Exceptional phenotypes rules[1]	5	7	7	0	0
	6	8	8	0	0
	7	2	2	0	0
Interpretive rules[2]	8	6	2	0	4
	9	3	0	1	2
	10	4	3	0	1
	11	5	2	1	2
	12	10	6	3	1
	13	8	7	0	1

Table 2: Summary of the EUCAST rules available in different sources, along with the number that are fully implemented, partially implemented, or could not be included in our CDSS.

use RuQAR to perform this task, although the resulting knowledge bases are different, as explained in Section 5.

- Step 3 - Option b: External ontological reasoner. In this case, a new binary operator is used when we need to check whether a bacterium or antimicrobial is related to a particular concept and no extra rules are, therefore, required. We specifically used HermiT [18] to perform the ontology reasoning required during concept evaluation. It took one week to study the expansion capabilities of Drools and to code and test the new binary operator.

4.3.4. Step 4: computing EUCAST rules

Finally, we implemented the EUCAST rules using the REO terms as a basis. However, not all the rules could be implemented because some of them require antimicrobial test parameters that are not available in the CDSS, or because they suggest that additional tests should be made rather than inferring a phenotype. For example, rule 13.7 indicates *IF a Haemophilus influenzae is resistant in nalidixic acid disk diffusion screen test, THEN determine MIC of the fluoroquinolone to be used in therapy.*, yet neither the kind of test (disk diffusion) nor the Minimum Inhibitory Concentration (MIC) values are available in our system. The rules eventually implemented are listed in Table 2.

When a rule is fired, it inserts a fact indicating the inference performed. The CDSS uses these new facts in different ways, depending on the source rule:

- Facts from intrinsic resistance rules: Used to detect inconsistencies in AST results and wo warn of possible ineffective treatments.

[1] Rules extracted from [42]

[2] Rules extracted from [3]

- Facts from exceptional phenotypes rules: Used to detect inconsistencies in AST results.
- Facts from interpretive rules: Used to detect inconsistencies in AST results, to complete AST reports and to warn of possible ineffective treatments.

We initially extracted the EUCAST expert rules from [3]. It took approximately two weeks to code the EUCAST rules, and an extra week to perform some tests on them.

In a later publication [42], the intrinsic and exceptional rules were revised to take into consideration the recent changes in phenotypes in many species. It took an extra week to perform a second iteration of the methodology in order to update and test the new resistance patterns. Extra rules were also extracted from the headings of some EUCAST tables, which include knowledge such as *"Gram-positive bacteria are also intrinsically resistant to aztreonam, temocillin, polymyxin B/colistin and nalidixic acid"*. Despite not having a specific rule number within the EUCAST document, these resistance patterns were coded as extra Drools rules.

Finally, a total of 48 Drools rules for intrinsic resistances, 17 for exceptional phenotypes and 25 for interpretive resistances were coded.

5. Performance experiments

One key aspect for a successful acquisition of new knowledge in a CDSS is the scalability of the methodology. The number of rules to implement, the concepts in use and the implementation of the methodology are essential factors for scalability. We therefore analyse and discuss the differences in performance between the approaches described in Step 3 as regards the size of the REO to be queried and the number of expert rules to be executed.

5.1. Scalability study

We tested three different approaches for Step 3: two based on Step 3.a and one for Step 3.b. We specifically used:

- A custom initial knowledge base. We followed the explanation in Step 3.a to build a script with which to transform all available classes in the ontology into traits in Drools, and to create rules to assign each fact to its most specific class in the ontology.
- An initial knowledge base created with RuQAR. We used the RuQAR tool to convert the ontological relationships into Drools rules. This basically transforms each available axiom into a rule that asserts facts of a *Triple* kind, which contains a subject, a predicate and an object emulating the triples used to describe ontologies in the Semantic Web. For example, if *Escherichia Coli* is a subclass of *Enterobacteriaceae* in the ontology, a rule will be created to search for triples linking any subject to *Escherichia coli*

with the *type* predicate, and will insert a *Triple* linking that subject to *Enterobacteriaceae* also by the *type* predicate. Additionally, the individuals within the ABox are directly converted into *Triple* objects. These objects must be inserted as facts before starting the forward chaining reasoning process.

- A custom operator which delegates ontology reasoning to HermiT. We followed the idea described in Step 3.b to create a custom binary operator with which to ask the ontological reasoner whether a fact is related to a class within the ontology.

We built test ontologies with different numbers of classes, from approximately 100 to 1,000, with the objective of measuring the relevance of the ontology size as regards its performance. These ontologies were subsets of our final ontology of microorganisms. We additionally created sets of rules that check whether a specific microorganism belongs to a species that is present in the ontology, simulating the requirements of an expert rule to be launched.

Three sets of 100, 500 and 1,000 “expert” rules were created with the objective of testing the impact of the number of rules launched.

All the experiments were executed on an Intel Xeon E5 at 3.60 Ghz with 8 GB of RAM and by running Windows 10, Drools 7.5.0 and HermiT 1.3.8.

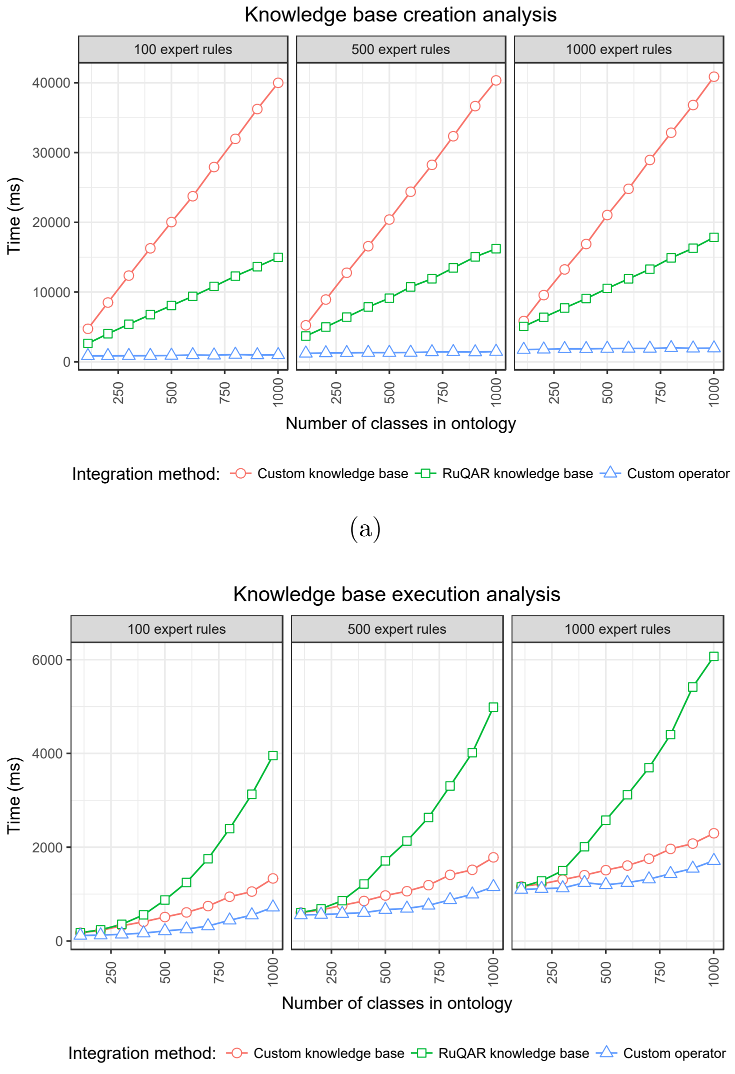
5.2. Results

We measured two relevant factors: a) the time required to create the knowledge base, that is, the time that it takes Drools to compile all the rules and prepare them to be launched, and b) the time required to execute all the rules. Each experiment was repeated 100 times in order to obtain relevant measures. Figure 9 depicts the mean times obtained for the initialization (Figure 9a) and execution (Figure 9b) of the knowledge base, depending on the number of classes in the ontology and the number of expert rules available.

As shown in these figures, the approach based on the definition of a custom operator performs better as regards both the initialization and execution. When creating an initial knowledge base, the approach based on RuQAR performs better than our custom approach during initialization, yet it is slower during execution. These results are discussed in detail in the following section.

5.3. Discussion

The results of the experiments show that the worst performance time in rule execution was attained when creating an initial knowledge base using RuQAR, while an intermediate performance was obtained with knowledge base creation. Furthermore, the knowledge base generated follows a similar conceptual approach to that used by Semantic Web techniques, relying on triples for the representation of relationships, which leads to a less clear rule definition than the examples shown in Figures 6 and 7. However, RuQAR is easy to use and a knowledge base can be obtained from an OWL file with a few lines of code.



(a)

(b)

Figure 9: Comparison of the mean time required by the proposed approaches to combine ontologies and production rules during the building of the knowledge base (a) and during rule execution (b). The graphs are divided into three parts, each of which shows the results obtained with different numbers of expert rules.

Furthermore, since it is a general solution, it can capture more complex ontological relationships (equality, reflexivity, etc.) than the other approaches, which are focused mostly on the subsumption of terms.

On the contrary, our approach for generating a custom knowledge base on the basis of dynamic typing performs worst in knowledge base creation, while it performs intermediately in rule execution. Using this approach requires skills in OWL in order to create a script that is capable of parsing the local ontologies. However, this strategy could be adapted to production rules engines other than Drools.

Finally, the use of a custom operator leads to the best performance as regards both initialization and execution. The knowledge base creation time is not affected by the ontology size because it does not require extra rule or type definitions. This is an additional advantage for maintenance, since with the previous approaches, a change in the ontology makes it obligatory to regenerate the files containing the initial knowledge base. Furthermore, the queries to the HermiT reasoner seem to be sufficiently fast to be executed while the rules are being evaluated without an excessive impact on the execution time. However, the development of a new operator requires advanced skills in Drools programming. Furthermore, it may not be possible to use this approach with a different rule engine since the definition of custom operators is not always permitted.

6. Related Work

The adoption of mechanisms with which to exchange knowledge among CDSSs has been widely analysed in recent years. In [43–45], the models used to acquire knowledge and grant the interoperability of CDSSs are reviewed. Based on such studies, we group the recent research available in literature as: *terminologies*, *interpretable structures*, *logic specifications*, *semantic technologies* and *methodologies*.

Clinical terminologies are the essential core of modern CDSSs and the first step required to allow logic constructions of clinical rules. In our opinion, this is the lowest level of modelling, required when an accurate description and standard language are needed. According to the studies carried out in [46], the clinical performance can improve by about 20% when using such standards. CDSSs are domain dependent systems and the terminology adopted, therefore, depends on the specific clinical problem, such as ICD for disease description, or ATC for pharmacy codification. However, as [44] highlights, the myriad of classifications is the main barrier to adopting the most suitable clinical terminology with which to acquire and transfer knowledge in CDSSs. In our case study, NCBI and ATC were selected under the supervision of clinical experts.

Interpretable structures are used in CDSSs when data models need to interoperate. From a computational point of view, these structures can be considered as the syntax level of data processing. According to systematic reviews in literature [44], HL7 CDA is perhaps the most widely studied and extended to represent medical records and messages between institutions. Archetype-based proposals approach the problem from a knowledge maintenance and stability perspective. Archetypes are computable definitions of clinical concepts, specifying clinical content for its re-use and sharing. In [47], the use of openEHR and archetypes is studied in order to intercommunicate medical records and CDSSs. The interpretable structures focus on providing medical record mechanisms for their interoperation in the form of constrained descriptions of data structures, which are relatively complex. However, the reasoning capacities of rule-based CDSSs are limited and, in some cases, cannot directly manage this type of constraint models. In [43], the authors analyse the suitability of clinical data standards (HL7 vMR, HL7 CDA, ISO/CEN 13606 and openEHR archetypes) as regards supporting CDSS, and propose an evaluation methodology. This study identifies key drawbacks and states the benefits of collaboration between standardisation initiatives in order to advance an adequate support of CDSSs.

Logic specifications are needed to share any kind of clinical algorithms, including the recommendations from clinical guidelines (CG) and the knowledge management approaches that have appeared in recent literature [44]. Great efforts have been made to study the executable elements of a computerised clinical guideline, often requiring specific guideline languages such as ASBRU, GLIF or ProFORMA [48]. These approaches mainly focus on clinical guideline structures and the expressibility needs of the computational language required. Very little attention is paid to how the CG language and rule-based systems can interact. In [49], we propose the use of BPMN and DRL to encapsulate rule-based decision actions within a clinical activity flow of a CG. Some other researchers focus their attention on clinical knowledge representation and management. Arden syntax structures medical decisions in the form of interpretable rules, modularised in the MLM components as part of the HL7 standard [50].

Semantic Web technologies have been tested for CDSS development. For example, the knowledge of clinical pathways is modelled in [51] using Semantic technologies and tested in standardised caesarian interventions in order to predict outcomes in a maternity department. The essential Semantic Web architecture is presented in this work, describing a reference ontology as a formalism of the representation and exchange of knowledge. SWRL-rule-based methods are used for temporal reasoning purposes. Some other efforts focus on improving current standards with Semantic technologies. For example, in [52], the authors identify the lack of integration between archetype definitions and clinical ontologies. Lezcano et al. propose a method with which to translate archetype definitions of openEHR into OWL and then use SWRL for inferring purposes. In [53], we developed tools for the acquisition and management of medical knowledge for diagnosis using deep-causal models by employing a tailored ontology in paediatric departments and intensive care units. More recently, the use of the Linked Data architecture paradigm has been proposed to define the specifications of CDSS services [54]. Semantic representations of CDSS components are used to ease the interoperability and reuse of knowledge. However, these proposals are tested in scenarios in which new implementations of the CDSS knowledge base are possible. Despite the advantages of ontology reasoning, the reimplementation of the system is costly in existing rule-based CDSS scenarios and other alternatives should, perhaps, be considered.

An alternative option in such scenarios is the integration of ontologies into databases and production rules. The use of production rules to support clinical decisions is a mature field, although ontology integration is still an open research topic [55]. There is a wealth of proposals with which to map database data and ontological concepts [56], or to generate domain-specific ontologies from data contained in relational tables [57]. Furthermore, tools with which to translate ontological relationships into production rules have been developed [25, 26]. Despite the efforts made, the specific requirements of clinical contexts are not considered in the aforementioned proposals. We believe that the REO methodology is a clear contribution in this direction.

Despite the advances made as regards the acceptance of CDSS in daily practice, little efforts have recently been made to develop *methodologies* with which to extract and integrate knowledge for CDSS. The work described in [5] presents a knowledge engineering methodology that can be used to develop the knowledge base of a CDSS using HL7 RIM and ontologies to represent patient data. The proposed methodology focuses on the traditional knowledge cycle (acquisition, representation, application and evaluation). Knowledge is retrieved using semantic technologies in the form of SPARQL queries. In [6], the objective of the Knowledge Quality Assessment (KQA) is to improve the quality of knowledge acquired for CDSSs using Semantic Web technologies. One novel aspect of KQA is the definition of specific metrics in order to quantify the sources, context and applicability of the knowledge available. We believe that KQA and our methodology are complementary. While KQA focuses on quality indicators for knowledge acquisition during the design process, our proposal can be used as an implementation guideline.

7. Conclusion

In this paper, we have studied how knowledge is acquired and transferred in rule-based CDSSs already running in hospitals. We propose a straightforward 4-step methodology based on a reference ontology (REO). The objective of our methodology is to balance the importance placed on supporting the acquisition of new knowledge supported by ontologies and to preserve the current CDSS reasoning architecture and local hospital terminologies. We have successfully applied our approach in WASPSS, a running CDSS, in order to share expert rules in a EUCAST antimicrobial testing problem.

In order to explore the most suitable implementation of the methodology proposed, we tested three different strategies concerning the use of ontologies and rule engines: a) using a custom initial knowledge base derived from ontological relationships, b) using a similar approach, but employing RuQAR, and c) using a custom operator which delegates to an ontological reasoner. The third approach performs better in our setting, yet it is more difficult to implement than the others. Despite the fact that our empirical data is limited to Drools and HermiT, the main results obtained may be useful when deciding which strategy to follow in other scenarios.

In future work, we intend to extend our reference ontology by integrating and linking ontologies with adverse drug reactions. We additionally intend to use the rule-based approach as a basis on which to acquire other sources of knowledge, such as the procedural knowledge contained in clinical guidelines or models obtained with data mining techniques.

Acknowledgments

This work was partially funded by the Spanish Ministry of Economy and Competitiveness under the WASPSS project (Ref: TIN2013-45491-R) and its associated predoctoral grant (Ref: BES-2014-070682), and by the European Fund for Regional Development (EFRD, FEDER).

References

[1] R. A. Greenes, Clinical Decision Support: The Road to Broad Adoption, Academic Press, 2014.

[2] J. C. Giarratano, G. Riley, Expert systems : principles and programming, 4th Edition, Thomson Course Technology, 2005.

[3] R. Leclercq, R. Cantón, D. F. J. Brown, C. G. Giske, P. Heisig, a. P. Macgowan, J. W. Mouton, P. Nordmann, a. C. Rodloff, G. M. Rossolini, C. J. Soussy, M. Steinbakk, T. G. Winstanley, G. Kahlmeter, EUCAST expert rules in antimicrobial susceptibility testing, Clinical Microbiology and Infection 19 (2) (2013) 141–160. doi:10.1111/j.1469-0691.2011.03703.x.

[4] T. Berners-Lee, J. Hendler, L. Ora, The semantic web, Scientific american 284 (5) (2001) 28–37.

[5] Y.-F. Zhang, Y. Tian, T.-S. Zhou, K. Araki, J.-S. Li, Integrating hl7 rim and ontology for unified knowledge and data representation in clinical decision support systems, Computer Methods and Programs in Biomedicine 123 (C) (2016) 94–108. doi:10.1016/j.cmpb.2015.09.020.
URL http://dx.doi.org/10.1016/j.cmpb.2015.09.020

[6] S. Zolhavarieh, D. Parry, KQA: A knowledge quality assessment model for clinical decision support systems, in: MEDINFO 2017: Precision Healthcare through Informatics - Proceedings of the 16th World Congress on Medical and Health Informatics, Hangzhou, China, 21-25 August 2017., 2017, pp. 983–986. doi:10.3233/978-1-61499-830-3-983.
URL https://doi.org/10.3233/978-1-61499-830-3-983

[7] G. Riley, Clips: An expert system building tool, in: The Second National Technology Transfer Conference and Exposition, Vol. 2, Washington, D.C., 1991, pp. 149–158.

[8] E. Friedman-Hill, Jess in action: rule-based systems in java, Manning publications, 2003.

[9] The JBoss Drools Team, Drools Documentation - Version 7.5.0.Final, accessed on 13-02-2018 (2017).
URL https://docs.jboss.org/drools/release/7.5.0.Final/drools-docs/html{_}single/

[10] M. G. Kahn, S. a. Steib, V. J. Fraser, W. C. Dunagan, An expert system for culture-based infection control surveillance, in: Proceedings of the Annual Symposium on Computer Applications in Medical Care., 1993, pp. 171–175.

[11] W. Yao, A. Kumar, CONFlexFlow: Integrating Flexible clinical pathways into clinical decision support systems using context and rules, Decision Support Systems 55 (2) (2013) 499–515.

[12] D. Dunsmuir, J. Daniels, C. Brouse, S. Ford, J. M. Ansermino, A knowledge authoring tool for clinical decision support, Journal of Clinical Monitoring and Computing 22 (3) (2008) 189–198. doi:10.1007/s10877-008-9124-1.

[13] P. Van Hille, J. Jacques, J. Taillard, A. Rosier, D. Delerue, A. Burgun, O. Dameron, Comparing Drools and ontology reasoning approaches for telecardiology decision support, in: J. Mantas, S. K. Andersen, M. C. Mazzoleni, B. Blobel, S. Quaglini, A. Moen (Eds.), Quality of Life through Quality of Information, Vol. 180 of Studies in Health Technology and Informatics, IOS Press, 2012, pp. 300–304.

[14] P. D. Clayton, T. A. Pryor, O. B. Wigertz, G. Hripcsak, Issues and Structures for Sharing Medical Knowledge Among Decision-Making Systems: The 1989 Arden Homestead Retreat, in: Proceedings of the Annual Symposium on Computer Application in Medical Care, American Medical Informatics Association, 1989, p. 116.

[15] T. a. Pryor, G. Hripcsak, The Arden Syntax for medical logic modules, International Journal of Clinical Monitoring and Computing 10 (4) (1993) 215–224.

[16] W3C OWL Working Group, OWL 2 Web Ontology Language Document Overview (Second Edition), W3c recommendation, W3C, accessed on 13-12-2018 (2012).
URL http://www.w3.org/TR/owl2-overview/

[17] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, Y. Katz, Pellet: A practical OWL-DL reasoner, Web Semantics 5 (2) (2007) 51–53. doi:10.1016/j.websem.2007.03.004.

[18] R. Shearer, B. Motik, I. Horrocks, HermiT : A Highly-Efficient OWL Reasoner, Complexity 432 (2008) 10.

[19] I. Horrocks, P. F. Patel-schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, SWRL: A Semantic Web Rule Language Combining OWL and RuleML, Tech. Rep. 21 May 2004, W3C (2004).
URL https://www.w3.org/Submission/SWRL/

[20] M. Kifer, H. Boley, RIF Overview (Second Edition), W3c note, W3C, accessed on 9-2-2018 (2013).
URL https://www.w3.org/TR/rif-overview/

[21] H. Knublauch, J. A. Hendler, K. Idehen, SPIN - Overview and Motivation, W3c member submission, W3C, accessed on 9-2-2018 (2011).
URL https://www.w3.org/Submission/2011/SUBM-spin-overview-20110222/

[22] H. Knublauch, D. Kontokostas, Shapes Constraint Language (SHACL), W3c recommendation, W3C, accessed on 9-2-2018 (2017).
URL https://www.w3.org/TR/shacl

[23] G. Meditskos, N. Bassiliades, CLIPS-OWL: A framework for providing object-oriented extensional ontology queries in a production rule engine, Data and Knowledge Engineering 70 (7) (2011) 661–681. doi:10.1016/j.datak.2011.04.001.

[24] V. Vassiliadis, J. Wielemaker, C. Mungall, Processing OWL2 ontologies using Thea: An application of logic programming, in: Proceedings of the 6th International Conference on OWL: Experiences and Directions (OWLED'09), Vol. 529 of CEUR Workshop Proceedings, 2009.

[25] G. Meditskos, N. Bassiliades, DLEJena: A practical forward-chaining OWL 2 RL reasoner combining Jena and Pellet, Journal of Web Semantics 8 (1) (2010) 89–94. doi:10.1016/j.websem.2009.11.001.

[26] J. Bak, M. Nowak, C. Jedrzejek, RuQAR: Reasoning Framework for OWL 2 RL Ontologies, in: The Semantic Web: ESWC 2014 Satellite Events, Vol. 8798 of Lecture Notes in Computer Science, Springer, Cham, 2014, pp. 195–198. doi:10.1007/978-3-319-11955-7_17.
URL http://link.springer.com/10.1007/978-3-319-11955-7{_}17

[27] WHO Collaborating Centre for Drug Statistics Methodology, Guidelines for ATC classification and DDD assignment 2013, Tech. rep., WHO Collaborating Centre for Drug Statistics Methodology, Oslo (2012).
URL http://www.whocc.no/filearchive/publications/1{_}2013guidelines.pdf

[28] F. M. Cohan, What are Bacterial Species?, Annual Review of Microbiology 56 (1) (2002) 457–487.

[29] S. Federhen, The NCBI Taxonomy database, Nucleic acids research 40 (Database issue) (2012) D136–D143. doi:10.1093/nar/gkr1178.
URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3245000{&}tool=pmcentrez{&}rendertype=abstract

[30] T. S. De Silva, D. MacDonald, G. Paterson, K. C. Sikdar, B. Cochrane, Systematized nomenclature of medicine clinical terms (SNOMED CT) to represent computed tomography procedures, Computer Methods and Programs in Biomedicine 101 (3) (2011) 324–329. doi:10.1016/j.cmpb.2011.01.002.
URL http://linkinghub.elsevier.com/retrieve/pii/S0169260711000125

[31] D. Lee, R. Cornet, F. Lau, N. de Keizer, A survey of SNOMED CT implementations, Journal of Biomedical Informatics 46 (1) (2013) 87–96. doi:10.1016/j.jbi.2012.09.006.
URL http://dx.doi.org/10.1016/j.jbi.2012.09.006

[32] B. Smith, M. Ashburner, C. Rosse, J. Bard, W. Bug, W. Ceusters, L. J. Goldberg, K. Eilbeck, A. Ireland, C. J. Mungall, N. Leontis, P. Rocca-Serra, A. Ruttenberg, S.-A. Sansone, R. H. Scheuermann, N. Shah, P. L.

Whetzel, S. Lewis, S. Lewis, The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration, Nature Biotechnology 25 (11) (2007) 1251–1255. doi:10.1038/nbt1346.
URL http://www.nature.com/articles/nbt1346

[33] S. Auer, S. Dietzold, J. Lehmann, S. Hellmann, D. Aumueller, Triplify: light-weight linked data publication from relational databases, in: Proceedings of the 18th international conference on World wide web - WWW '09, ACM Press, New York, New York, USA, 2009, p. 621. doi:10.1145/1526709.1526793.
URL http://portal.acm.org/citation.cfm?doid=1526709.1526793

[34] C. Bizer, C. Bizer, D2RQ - treating non-RDF databases as virtual RDF graphs, in: Poster in 3rd International Semantic Web Conference (ISWC2004), 2004.
URL http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.126.2314

[35] O. Erling, I. Mikhailov, RDF Support in the Virtuoso DBMS, in: T. Pellegrini, S. Auer, K. Tochtermann, S. Schaffert (Eds.), Networked Knowledge - Networked Media, Studies in Computational Intelligence, vol 221, Springer, Berlin, Heidelberg, 2009, pp. 7–24. doi:10.1007/978-3-642-02184-8_2.
URL http://link.springer.com/10.1007/978-3-642-02184-8{_}2

[36] S. Bragaglia, F. Chesani, A. Ciampolini, P. Mello, M. Montali, D. Sottara, An Hybrid Architecture Integrating Forward Rules with Fuzzy Ontological Reasoning, in: M. Graña Romay, E. Corchado, M. Garcia Sebastian (Eds.), Hybrid Artificial Intelligence Systems. HAIS 2010, Vol. 6076 of Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2010, pp. 438–445. doi:10.1007/978-3-642-13769-3_53.
URL http://link.springer.com/10.1007/978-3-642-13769-3{_}53

[37] W. Drabent, J. Henriksson, J. Małuszyński, HD-rules: A hybrid system interfacing prolog with DL-reasoners, in: Proceedings of the ICLP'07 Workshop on Applications of Logic Programming to the Web, Semantic Web and Semantic Web Services (ALPSWS 2007), Vol. 287 of CEUR Workshop Proceedings, 2007, pp. 76–90.

[38] EUCAST - European Committee on Antimicrobial Susceptibility Testing, http://www.eucast.org/, accessed on 18-10-2017.
URL http://www.eucast.org/

[39] F. Palacios, M. Campos, J. M. Juarez, S. E. Cosgrove, E. Avdic, B. Cánovas-Segura, A. Morales, M. E. Martínez-Nuñez, T. Molina-García, P. García-Hierro, J. Cacho-Calvo, A clinical decision support system for an Antimicrobial Stewardship Program, in: HEALTHINF 2016 - 9th International Conference on Health Informatics, Proceedings, SciTePress, Rome, 2016, pp. 496–501.

[40] B. Cánovas-Segura, M. Campos, A. M. Nicolas, J. M. Juarez, F. Palacios, Development of a clinical decision support system for antibiotic management in a hospital environment, Progress in Artificial Intelligence 5 (3) (2016) 181–197.

[41] M. A. Musen, The protégé project: A look back and a look forward, AI Matters 1 (4) (2015) 4–12. doi:10.1145/2757001.2757003.
URL http://dl.acm.org/citation.cfm?doid=2757001.2757003

[42] The European Committee on Antimicrobial Susceptibility Testing, EUCAST Expert Rules Version 3.1, Tech. rep., The European Committee on Antimicrobial Susceptibility Testing (2016).
URL http://www.eucast.org/fileadmin/src/media/PDFs/EUCAST{_}files/Expert{_}Rules/Expert{_}rules{_}intrinsic{_}exceptional{_}V3.1.pdf

[43] A. González-Ferrer, M. Peleg, Understanding requirements of clinical data standards for developing interoperable knowledge-based dss: A case study, Computer Standards and Interfaces 42 (2015) 125–136. doi:https://doi.org/10.1016/j.csi.2015.06.002.
URL http://www.sciencedirect.com/science/article/pii/S0920548915000641

[44] L. Marco-ruiz, A. Budrionis, K. Yitbarek, J. Gustav, Interoperability Mechanisms of Clinical Decision Support Systems: A Systematic Review, in: Proceedings from the 14th Scandinavian Conference on Health Informatics 2016, no. 122 in Linköping Electronic Conference Proceedings, Linköping University Electronic Press, Linköpings universitet, Gothenburg, Sweden, 2016, pp. 13–21.

[45] S. Zolhavarieh, D. Parry, Q. Bai, Issues associated with the use of semantic web technology in knowledge acquisition for clinical decision support systems: Systematic review of the literature, JMIR Medical Informatics 5 (3) (2017) e18.

[46] L. Ahmadian, M. van Engen-Verheul, F. Bakhshi-Raiez, N. Peek, R. Cornet, N. F. de Keizer, The role of standardized data and terminological systems in computerized clinical decision support systems: Literature review and survey, International Journal of Medical Informatics 80 (2) (2011) 81–93, special Issue: Security in Health Information Systems. doi:https://doi.org/10.1016/j.ijmedinf.2010.11.006.
URL http://www.sciencedirect.com/science/article/pii/S1386505610002261

[47] M. Marcos, J. A. Maldonado, B. Martínez-Salvador, D. Boscá, M. Robles, Interoperability of clinical decision-support systems and electronic health records using archetypes: A case study in clinical trial eligibility, Journal of Biomedical Informatics 46 (4) (2013) 676–689.

doi:https://doi.org/10.1016/j.jbi.2013.05.004. URL http://www.sciencedirect.com/science/article/pii/S1532046413000701

[48] M. Peleg, Computer-interpretable clinical guidelines: A methodological review, Journal of Biomedical Informatics 46 (4) (2013) 744–763.

[49] B. Cánovas-Segura, F. Zerbato, B. Oliboni, C. Combi, M. Campos, A. Morales, J. M. Juarez, R. Marin, F. Palacios, A process-oriented approach for supporting clinical decisions for infection management, in: 2017 IEEE International Conference on Healthcare Informatics (ICHI), 2017, pp. 91–100. doi:10.1109/ICHI.2017.73.

[50] M. Samwald, K. Fehre, J. de Bruin, K.-P. Adlassnig, The Arden Syntax standard for clinical decision support: Experiences and directions, Journal of Biomedical Informatics 45 (4) (2012) 711–718. doi:10.1016/j.jbi.2012.02.001.

[51] Y. Ye, Z. Jiang, X. Diao, D. Yang, G. Du, An ontology-based hierarchical semantic modeling approach to clinical pathway workflows, Computers in Biology and Medicine 39 (8) (2009) 722–732. doi:https://doi.org/10.1016/j.compbiomed.2009.05.005. URL http://www.sciencedirect.com/science/article/pii/S0010482509000985

[52] L. Lezcano, M.-A. Sicilia, C. Rodríguez-Solano, Integrating reasoning and clinical archetypes using owl ontologies and swrl rules, Journal of Biomedical Informatics 44 (2) (2011) 343–353. doi:10.1016/j.jbi.2010.11.005. URL http://dx.doi.org/10.1016/j.jbi.2010.11.005

[53] J. M. Juarez, M. Campos, J. T. Palma, R. Marin, Computing context-dependent temporal diagnosis in complex domains, Expert Systems with Applications 35 (3) (2008) 991–1010.

[54] L. Marco-Ruiz, C. Pedrinaci, J. Maldonado, L. Panziera, R. Chen, J. G. Bellika, Publication, discovery and interoperability of Clinical Decision Support Systems: A Linked Data approach, Journal of Biomedical Informatics 62 (2016) 243–264. doi:https://doi.org/10.1016/j.jbi.2016.07.011. URL http://www.sciencedirect.com/science/article/pii/S153204641630065X

[55] D.-E. Spanos, P. Stavrou, N. Mitrou, Bringing relational databases into the Semantic Web: A survey, Semantic Web 3 (2) (2012) 169–209. doi:10.3233/SW-2011-0055.

[56] M. Hert, G. Reif, H. C. Gall, A comparison of RDB-to-RDF mapping languages, in: Proceedings of the 7th International Conference on Semantic Systems - I-Semantics '11, ACM Press, New York, New York, USA, 2011, pp. 25–32. doi:10.1145/2063518.2063522.

[57] F. Cerbah, Learning Highly Structured Semantic Repositories from Relational Databases: The RDBToOnto tool, in: The Semantic Web: Research and Applications, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 777–781. doi:10.1007/978-3-540-68234-9_57.