

Panacea, a Semantic-enabled Drug Recommendations Discovery Framework

Charalampos Doulaverakis^{1*}, George Nikolaidis², Athanasios Kleontas MD^{2,3} and Ioannis Kompatsiaris¹

¹Centre for Research and Technology Hellas, Information Technologies Institute, Thessaloniki, Greece

²Ergobyte S.A., Thessaloniki, Greece

³Theagenio Cancer Hospital, Thessaloniki, Greece

ABSTRACT

The paper presents Panacea, a semantic framework capable of offering drug-drug and drug-diseases interaction discovery. For enabling this kind of service, medical information and terminology had to be translated to ontological terms and be appropriately coupled with medical knowledge of the field. International standards, such as the ICD-10 and ATC classifications, provide the backbone of the common representation of medical data while the medical knowledge of drug interactions is represented by a rule base which makes use of the aforementioned standards. Representation is based on the light-weight SKOS ontology. A layered reasoning approach is implemented where at the first layer ontological inference is used in order to discover underlying knowledge, while at the second layer a two-step rule selection strategy is followed resulting in a computationally efficient reasoning approach. Details of the system architecture are presented while also giving an outline of the difficulties that had to be overcome. The paper compares the current approach to a previous published work by the authors, a service for drug recommendations named GalenOWL, and presents their differences in modelling and approach to the problem, while also pinpointing the advantages of Panacea.

1 INTRODUCTION

One of the health sectors where intelligent information management and information sharing compose valuable preconditions for the delivery of top quality services is personalized drug prescription. This is more evident in cases where more than one drug is required to be prescribed, a situation which is not uncommon, as drug interactions may appear. The problem is magnified by the wide range of available drug substances in combination with the various excipients in which the former are present.

If one takes into account that there exist more than 18,000 pharmaceutical substances, including their excipients, then it is clear that the continuous update of health care professionals is remarkably hard. Over this, the extensive literature makes discovery of relevant information a time consuming and difficult process, while the different terminologies that appear between sources add more burden on the efforts of medical professionals to study available information.

Semantic Web technologies can play an important role in the structural organization of the available medical information in a manner which will enable efficient discovery and access. Research projects funded for enabling Semantic Web technologies in the diagnosis and therapeutic procedures exist such as PSIP (Beuscart *et al.*, 2009) and Active Semantic Documents (Sheth, 2005) or

works such as (Adnan *et al.*, 2010), but they don't fully address the problem of automated drug prescription using drug-drug and drug-disease interactions.

Rule-based approaches have been proposed for addressing issues relating to biomedical ontologies research. It is common for ontologies written in expressive Semantic Web languages such as OWL, not be able to handle all requirements for capturing the knowledge in several biomedical and medicine domains. As a method for enriching the expressiveness of ontology languages, researchers have proposed the use of rules which act upon the defined ontological knowledge. According to (Golbreich, 2004) rules are helpful in the following situations relating to biomedical ontologies: defining "standard rules" for chaining ontology properties, "bridging rules" for reasoning across different domains, "mapping rules" for defining mappings between ontologies entities and "querying rules" for expressing complex queries upon ontologies. The author gives a thorough review of RuleML and SWRL, the two major ontology rule languages, the available rule formation tools and the reasoners. (Golbreich *et al.*, 2005) make use of the outcomes of the previous paper to showcase the need for rules in biomedical applications with a use case of a brain anatomy definition, where a brain structure ontology is defined in OWL but rules describing the relationships between the properties and entities are needed for correct annotation of MRI images. Another work citing the need for semantically enriched rules, where an ontology coupled with SWRL rules for annotating pseudogenes and answering research questions has been proposed in (Holford *et al.*, 2010). All the above papers present the need for extending ontologies with rules in order capture the knowledge of complex biomedical domains.

The paper presents Panacea, a semantic-enabled system for discovering drug recommendations and interactions. Panacea is based on experiences and lessons drawn from the development of GalenOWL (Doulaverakis *et al.*, 2012), a similar system which had Semantic Web technologies in its core. As such, Panacea can be considered the evolution of GalenOWL in terms of design and scalability. Panacea makes use of established and standardized medical terminologies together with a rich knowledge base of drug-drug and drug-diseases interactions expressed as rules. Panacea is implemented having in mind scalability, completeness of results and responsiveness in query answering.

The paper is organized as follows: Section 2 gives details about the architecture and usage of the framework. In Section 3 the data modelling approach is presented and in Section 4 the core ontology and the layered reasoning process is described while also outlying

*To whom correspondence should be addressed: doulaver@iti.gr

two approaches for rule-based reasoning. Section 5 gives an evaluation of the framework in terms of scalability and performance and the paper concludes with Section 6.

2 ARCHITECTURE AND FUNCTIONAL DESIGN

The purpose of Panacea is to provide drug prescription recommendations based on a patient's medical record, i.e. advise physicians to prescribe medications according to the drugs active substance indications and contraindications. For details regarding the initiative that triggered development of Panacea and the initial medical and pharmaceutical data that were available, the reader is encouraged to read (Doulaverakis *et al.*, 2012). Panacea has been developed in the Java programming language and is built using Apache Jena. Jena provides the API and methods to translate the medical knowledge of terminologies and pharmaceutical rules to semantic entities, while also providing the reasoning engine to enrich the knowledge base.

Panacea follows a layered reasoning process which is depicted in Figure 1. During the start-up of the system, the medical terminologies, namely ATC, UNII, ICD -10 and custom encodings, are transformed to semantic entities, using an appropriate vocabulary, and the initial ontology is constructed. The ontology binds to a reasoner to infer relations such as inheritance and unions. This process is performed once offline during initialization and the knowledge base is available to the system for further utilization. In order to get recommendations in Panacea, a patient instance with the appropriate medical record data is created and fed to the knowledge base. The reasoning process enriches the patient instance with inferred knowledge, thus making it explicit. On this enriched instance, and by utilizing a different reasoning process, the set of medical rules is applied upon. The result of this final stage of rule-based reasoning is the recommendations list which can be retrieved through SPARQL querying.

A key characteristic of the suggested architecture is that, regarding second level reasoning, the framework can utilize any rule-based reasoner or rule engine. Since all the inferred knowledge of the medical definitions and patient data is materialized in the knowledge base, the medical rules can be expressed and loaded in an appropriate rule engine. The rule engine could be an ontology reasoner, such as Jena's reasoning engine, or a business rule manager such as Drools or even CLIPS with appropriate customizations in the data structures. This approach helps in bringing together the best of both worlds: semantic and meaningful representation of data using Semantic Web technologies and the maturity of traditional rule engines in efficiently handling complex and large amounts of rules.

2.1 Use case scenario

In order to demonstrate the benefits of the proposed semantic recommendation system, a use case regarding a possible scenario is described: An elder man visits his family doctor complaining for pain in his right lower back and abdominal region which is accompanied with fever. After appropriate clinical examination, he is diagnosed with right pyelonephritis (ICD-10 code: N11.0). According to the patient's medical history, he is suffering from chronic atrial fibrillation for which he receives clopidogrel (ATC code: B01AC04), vertigo for which he receives cinnarizine (ATC: N07CA02), high arterial blood pressure for which he receives candesartan (ATC: C09CA06) and amlodipine (ATC: C08CA01), and

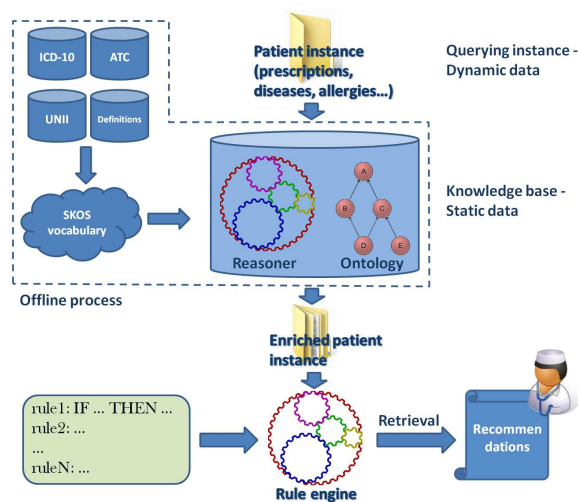


Fig. 1. Panacea framework architecture and data flow

diabetes mellitus for which he receives metformin (ATC: A10BA02) and sitagliptin (ATC: A10BH01). For the new condition of pyelonephritis that was diagnosed, the treating doctor must decide a number of things. Regarding the prescription for treating this new disease, the doctor has to decide which active substances to prescribe in order to treat the resulting inflammation, the cause of the inflammation, the back and abdominal pain and the resulting fever. However, before a decision is made the following factors regarding the patient's medical history should also be considered:

- There should be a check for drug-drug interaction that the patient is taking, before the onset of the new condition (the pyelonephritis)
- There should be a check for drug-disease interaction that the patient is taking, with the new condition
- The new prescription has to be verified that it will not have adverse effects or interactions with the previously prescribed medication and with the patient's medical history

It is clear that the task for the doctor can be hard and a misjudgement could lead to wrong prescriptions. Using an automated drug recommendation system can minimize this risk. The recommendation system will use the input data and the pharmaceutical rules in order to infer a treatment that will be safe for the patient.

3 SEMANTIC TRANSFORMATIONS

Panacea is built on top of international standards of medical terminology in order to represent medical and pharmaceutical information. The following standard terminologies are used:

ICD-10: The World Health Organization classification of diseases. It is used in Panacea for unique identification of diseases thus uniquely identifying drug indications and contraindications related to diseases.

UNII: Unique Ingredient Identifier. Used for the identification of active ingredients found in drugs. In Panacea it is used for uniquely identifying drug indications and contraindications related to ingredients.

ATC: The Anatomical Therapeutic Chemical Classification is used for the classification of drugs. In Panacea it is used in similar fashion to UNII.

IVT: The International Virus Taxonomy is used for the classification of viruses. In Panacea it is used in order to uniquely drug indications and contraindications related to viruses.

Apart from these international standards, a number of domain classifications have been declared and used in order to enhance the usability of the system or to represent data that are not included in the standards. These classifications act as supplementary to the standards.

Substance: As the use of encodings for drug ingredients is not convenient for humans, the identification of active substances is done using its common name references in medical bibliography. These names come from international standards such as the International Nonproprietary Names (INN) and others such as USAN (United States Adopted Name) or BAN (British Approved Name). Members of this identification list are substances such as *acetazolamide* or *isradipine*. In addition, substances correspond to ATC codes such that for example *acetazolamide* \equiv S01EC01. The substances are the actual recommendations of Panacea.

Custom Concepts: While the ATC, ICD-10, UNII and IVT standards are complete, they are designed for use in contexts different from Panacea and drug recommendations, e.g. for annotation, search or information retrieval. As such, it is often desirable to enrich the knowledge base with information that, while not standard, will aid in the usability and overall efficiency of the system. Especially for medical/pharmaceutical rules formulation, it was found out that there were occasions that the definition of diseases, drugs or other was either absent, incomplete or too general to be useful for a rule definition. An example for the lack of a definition in ICD-10 is the absence of a precise and specific code for “Chronic obstructive pulmonary disease” or for “Hypertrophy (benign) of prostate”. For this reason, a number of custom concepts have been defined. Examples of such concepts is disease definition such as “Narcolepsy”, microorganisms such as “clostridium clostridiiformis” or medical acts such as “upper extremity arteriography”.

Custom Concept Collections: Certain “groups” of substances and/or diseases are frequently present in drug interactions and these groups are not recorded explicitly in any standardized classification, so it’s more convenient for medical use to specify these custom groups. These often used groups are termed “conditions” in Panacea and are defined by medical experts. A condition can appear as a premise in other condition definitions, as in the Custom Concept Collection *cardiac-rhythm-abnormalities* below, thus enabling their recursive definition:

cardiac-rhythm-abnormalities = cc:bradycardia | icd:R00 | cc:tachycardia | icd:O68.0 | icd:O68.2

where *cc:bradycardia* is defined as (*icd:I49.5* | *icd:R00.1* | *icd:O68.0*) and *cc:tachycardia* as (*icd:R00.0* | *icd:I49.5* | *icd:I47* | *icd:O68.0*). “*icd:*” stands for the ICD-10 namespace.

3.1 SKOS vocabulary

In the approach followed in (Doulaverakis *et al.*, 2012), the medical standards and the custom definitions were translated to OWL classes, primitive and defined. While this approach had the benefit

of using the language’s semantics to model the available information, there were problems resulting from this design decision. One of the major issues was the difficulty in scaling the system. Until currently, very few reasoners are available that can efficiently handle the amount of class definitions and reasoning required to run the system, both in terms of memory consumption and speed.

In Panacea, a different approach was adopted. The SKOS (SKOS: Simple Knowledge Organization System, 2009) vocabulary is a W3C recommendation, it’s built using RDF(S) semantics and has been developed as a low-cost migration path for porting existing knowledge organization systems, such as thesauri, taxonomies, classification schemes and subject heading systems, to the Semantic Web. It enables a “lightweight” semantic representation of such knowledge systems and is a good match for the medical standards that are used in Panacea. As such, all the terminologies which are mentioned in the previous section have been transformed using the SKOS vocabulary automatically using a parser.

Comparing SKOS to the approach followed in (Doulaverakis *et al.*, 2012), instead of representing the ATC, ICD-10 and UNII classifications as top-level classes, they are now represented as instances of the *skos:ConceptScheme* class. “*skos:*” stands for the SKOS namespace. Each entry in these classifications is represented as an instance of the *skos:Concept* class. The OWL class hierarchy of (Doulaverakis *et al.*, 2012) is represented in Panacea using the properties *skos:broaderTransitive* and *skos:narrowerTransitive*, while the unions of classes for Custom Concepts Collections are represented using the *skos:member* property. Correspondence between the semantic transformation methodologies that were followed in the current work and in (Doulaverakis *et al.*, 2012) is presented in Table 1.

It is interesting to note that the SKOS vocabulary offers exactly what is needed in order to capture the semantics of the medical classifications without making sacrifices in expressiveness. One can argue that it can be considered more precise than the OWL expressions, as in the case of the similarity of Substances and ATC codes. This similarity is better represented by the *skos:closeMatch* relation than *owl:equivalentClass*. For Panacea a total of 64, 658 definitions of classification codes have been expressed using SKOS.

4 PANACEA ONTOLOGY AND REASONING

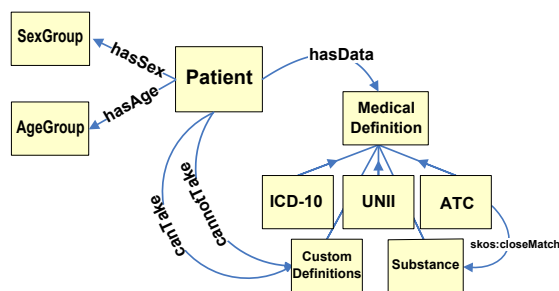
The core ontology of Panacea is visualized in Figure 2. The aforementioned SKOS ontologies were imported to the Panacea core ontology under the *MedicalDefinitions* class. The Patient class holds the patient instances and is connected to the *MedicalDefinitions* class with the *hasData* properties. The patient recommendations, indications and contraindications, regarding substances that should and should not be prescribed are expressed with the *canTake* and *cannotTake* properties, respectively. The patients age group and sex group are expressed through the *hasAgeGroup* and *hasSexGroup* properties.

4.1 Medical reasoning

When querying the system for recommendations, a patient instance is created with the initial patient data (through the *hasData*, *hasAgeGroup* and *hasSexGroup* properties) and is loaded in the knowledge base. The reasoner, using RDFS inference and a small number of additional rules, infers all the implicit patient data. As an example consider a patient who suffers from

Table 1. Correspondence between the semantic transformation in the early GalenOWL system and the proposed Panacea framework

	GalenOWL	Panacea
Annotation	rdfs:label	skos:prefLabel
Equivalence	owl:equivalentClass	skos:closeMatch
Custom collections	owl:unionOf	skos:member
Hierarchy	rdfs:subClassOf	skos:broaderTransitive

**Fig. 2.** Panacea ontology

a form of thrombocytopenia. An instance is created with the property $\langle \text{pan:patient pan:hasData icd:D69.6} \rangle$. The reasoner through the `skos:broaderTransitive` relation will infer the triples $\langle \text{pan:patient pan:hasData icd:D69} \rangle$, $\langle \text{pan:patient pan:hasData icd:D65-D69} \rangle$, $\langle \text{pan:patient pan:hasData icd:D50-D89} \rangle$. Additionally, the custom collection definition of `pnc-cc:deficiency-bone-marrow` has `icd:D69.6` as one of its members so the triplet $\langle \text{pan:patient pan:hasData pnc-cc:deficiency-bone-marrow} \rangle$ will also be inferred. At the end, the patient instance will be enriched with all the underlying implicit information.

4.2 Rule-based reasoning

Drug recommendations in Panacea are generated using a rule-based approach. The rules express the indications and contraindications of drug substances while their premises are the medical definitions and the patients' age and sex group. The rules use the logical operators *and* (&) and *or* (|) and parentheses. An example of a rule is for the substance "lisuride" which is expressed as

lisuride = `icd:E22.0` | (`icd:E22.1` & (`icd:N91.0` | `icd:N97`)),
ageGroup=*adult* or *elder*

The above rule reads that the substance "lisuride" is recommended for adult and elder patients who suffer from E22.0, OR suffer from E22.1 AND one of the N91.0 OR N97. For using these rules, they have to be properly parsed and transformed in order to match the knowledge base and the enriched, with implicit knowledge, patient instance. The proposed rule structure allows modifications to specific rules without the changes affecting the rest of the rule base. This enables the rule base to be up-to-date with the latest clinical advancements, which is a requirement as clinical pharmacology and medicine are constantly evolving. Analysing Panacea's architecture

in Figure 1 it can be seen that due to the layered reasoning approach, the knowledge base (medical definitions + reasoner) is actually used for producing the enriched patient instance. This means that the instance can be fed to a rule reasoner which has appropriately loaded the medical-pharmaceutical rules, without the reasoner having to communicate with the knowledge base for further utilization. Using this approach and with proper modifications, any rule engine can be used to produce the drug recommendations. To demonstrate this ability, two separate rule engine integrations have been developed and are presented below. The medical rule base consists of 1,342 rules which were extracted and encoded directly from official documents, such as Summary of Product Characteristics (SPC) Patient Information Leaflets (PIL), regarding drug indications, contraindications, interactions and dosage. The validity of the rule base has already been assessed in (Doulaverakis et al., 2012).

It should be noted that work is under way in order to add more functionalities in the drug proposed recommendation system. One of these is the ability to offer additional information such as the proposed dosage for a recommended substance. In order to accomplish such a task, the pharmaceutical rules are being enriched with clinical variables that are important, other than sex and age group. These variables include somatometric characteristics such as height and body weight, creatinin clearance (useful for calculating the dosage for antineoplastic drugs) and the disease itself as a substance could be indicated at a specific dosage to treat a certain disease, but a different dosage is recommended for another disease.

4.2.1 Jena rule engine For using the rule engine of the Apache Jena API (Apache Jena, 2012) the rules had to be translated to the Jena rule language. An automated parser was developed for this purpose. As for most semantic rule reasoners, OR clauses are not allowed in a rule definition so separate rules had to be expressed for every premise that was OR'ed in the original rule base. For example, the rule for "lisuride" was expressed by 3 different rules:

1. $(\text{?patient pan:hasData icd:E22.0}) \rightarrow (\text{?patient pan:canTake sub:lisuride})$
2. $(\text{?patient pan:hasData icd:E22.1}) \rightarrow (\text{?patient pan:hasData icd:N91.0}) \rightarrow (\text{?patient pan:canTake sub:lisuride})$
3. $(\text{?patient pan:hasData icd:E22.1}) \rightarrow (\text{?patient pan:hasData icd:N97}) \rightarrow (\text{?patient pan:canTake sub:lisuride})$

This rule expansion resulted in a total of 6,451 rules to be expressed in the Jena language. Trying to load the whole rule base and performing inference for recommendations proved inefficient for

real time use, requiring on average as much as 8 seconds. In order to tackle this issue a coarse rule selection phase was introduced. The selection was executed in 2 iterations. During the first iteration, a subset \mathcal{A} of candidate rules is created from the initial rule base, that match the patient's sex and age group. This subset is selected for further processing. In the second iteration, rules from \mathcal{A} that contain at least one of the patient's data, i.e. a *skos* term, in their premises are singled out and a final set $\mathcal{R} \subseteq \mathcal{A}$ is created from them. Remembering that the implicit knowledge extraction was performed during the introduction of the patient instance to the reasoning framework, creation of \mathcal{R} is actually a simple and fast process. It merely requires string matching and all the whole processing is executed in memory. As a result the overall burden that is added to the whole reasoning process is minimal. From the initial rule base of 6,451 rules it is common for \mathcal{R} to contain as less as 50 rules, whose evaluation is much more efficient. Rule execution is performed with the Jena rule engine and the patient instance is modified and now contains the drug recommendations. These recommendations are retrieved through SPARQL querying, using Jena's query engine. The advantage of the Jena engine is that it can readily consume the patient instance for producing the recommendations.

4.2.2 Drools rule engine As an alternative approach, the Drools (Drools, 2012) business rule engine was used. In contrary to the Jena engine, Drools could not directly use the patient instance for performing reasoning. For this purpose, the instance was transformed to a Java bean, where the properties of the ontology Patient class are mapped to Java methods using the JenaBean API (<http://code.google.com/p/jenabean/>). The bean was appropriately declared to Drools and was handled for rule execution. A similar approach for integrating Jena and Drools was used in (Bragaglia *et al.*, 2010). The Drools Rule Language (DRL) permits the use of OR'ed clauses in the body, so the 1,342 original medical rules were translated to the same amount of rules in DRL, using an automated parser similar to the one used in the Jena approach. For example, the rule for "lisuride" from the previous paragraph was expressed in DRL as:

```

RULE `lisuride`
  WHEN
    p: Patient(data : hasData)
    exists( (MedicalDef(uri==icd:E22.0)
            from data) ||
           (MedicalDef(uri==icd:E22.1 && uri==icd:N91.0)
            from data) ||
           (MedicalDef(uri==icd:E22.1 && uri==icd:N97)
            from data) )
  THEN
    Substance lisuride = (Substance)JenaBean.
    reader().load(sub:lisuride);
    modify(p) {p.canTake(lisuride)}
  END

```

Execution was straightforward with no preprocessing required. Drools is optimized for handling large rule bases, so no rule pre-selection step was required as this would have little impact in reasoning efficiency. The result of this reasoning process is a modified patient Java bean with the drug recommendations. The bean is transformed to Jena model instance and SPARQL querying for retrieving the recommendations is possible. What this approach demonstrates is that it's possible to integrate business rule engines as reasoners in the framework, thus being able to make use of the

high efficiency and optimizations of these engines with the semantic description and interpretation of data.

5 EVALUATION AND DISCUSSION

For evaluating the framework, a comparison was made between the two approaches for the final stage reasoning and with GalenOWL (with values taken from (Doulaverakis *et al.*, 2012)). The comparisons were focused on the usability of the framework in a production environment as the rule base has been validated in (Doulaverakis *et al.*, 2012). Three parameters were measured. These were initialization time, the time to get the system up and running, memory consumption after initialization, and query response time, i.e. the time that is needed to have the rule base executed and the results retrieved. Results are shown in Table 2.

There are some points to discuss in the table results. Initialization involves loading the ontology in memory, performing inference, and preparing the medical rule base for patient data reasoning. In the Jena implementation, the rule base is processed and loaded only after the patient instance has been introduced to the system, while the Drools implementation loads the whole rule base on the engine before any patient data are introduced. As a result, Drools appears slower than the Jena approach regarding initialization. For the same reason, memory consumption appears greater for Drools. This metric corresponds to memory consumption from initialization to recommendations retrieval. While in Drools the whole rule base is loaded on memory, in Jena the approach was to load a small subset of the rule base that could possibly match the patient data, which leads to a smaller memory footprint. Finally, for query response the advantage is with Drools, as was expected, mainly due to the fact that Drools is a dedicated rule engine while Jena's focus is not at providing a state of the art reasoner and rule engine, but a versatile API for ontology management.

Numerically, the Jena approach seems to be more efficient than Drools, apart from the query execution time but for which the difference is not important. However, while for the present knowledge base Jena seems to perform better, this fact could change as more and more rules are added. It is estimated that eventually at its final stage, Panacea will incorporate more than 9,000 drug-drug and drug-disease interactions. As already said, Jena is more focused as an ontology API and less as an efficient rule engine which could eventually lead to scaling problems. On the other hand, scaling with Drools is not an issue. The value of business rule engines as Semantic Web reasoners has been previously exploited using approaches such as (O'Connor and Das, 2012), where the authors implemented two OWL2-RL Motik *et al.* (2009) reasoners using the Drools and Jess rule engines respectively. The use of traditional rule engines with the Semantic Web technologies brings together the best of both worlds, i.e. increased efficiency coupled with interoperability and semantic annotation of information.

What is also noticeable from Table 2 is the decreased memory requirement of Panacea compared to the previous OWL- based GalenOWL system, although the two approaches offer very similar functionality. As a result of this achievement, Panacea can accommodate a far greater knowledge base thus supporting the claim of increased scalability.

Panacea will eventually be offered as a service with potential customers being health care professionals. Other possible exploitation routes are being investigated such as integration to patient

Table 2. Evaluation between the 2 Panacea reasoning approaches and GalenOWL

	Panacea-Jena	Panacea-Drools	GalenOWL
Initialization time	32.0 s	34.7 s	148 s
Memory consumption	169 MB	280 MB	649 MB
of which rule base consumes	0 MB	111 MB	—
Query response time	47 ms	5 ms	16 ms

management systems in health clinics. The use of personalized drug prescription systems, as Panacea, in everyday practice will have major advantages to the society and the economy. A major benefit from the use of such systems is the reduction of medical costs through rational drug prescriptions that personalized drug prescription allows (Fischer *et al.*, 2008). Another benefit is a positive effect in public health with reduction of outbreaks relating to drug interactions or adverse effects (Ammenwerth *et al.*, 2008). All knowledge regarding drug information is encoded and is available to the experts in order to aid them during prescriptions thus acting as decision support systems. It should be stressed out that drug recommendation systems do not aim to replace medical experts but to support them in their practice.

A limitation of the proposed approach is that a rather large amount of manual effort by experts is required in order to populate and enrich the rule base. Although the semantic technologies that have been employed can make rule authoring simpler, no automated method for pharmaceutical rule generation has been integrated. However, one would argue that since rule authoring is performed by experts then the rules are verified and guaranteed to be correct. Even if an automated method, such as rule mining, had been implemented, the generated rules would still have to be verified by an expert in the field. Manual verification, although less intensive, would still be required.

6 CONCLUSIONS

The paper presented Panacea, a framework for semantic-enabled drug recommendations discovery. The framework utilizes a layered reasoning approach where the medical ontology and the patient data instances are fed to an extended RDF(S) reasoner in order to infer implicit knowledge. Drug recommendations are generated using the second reasoning layer where any common rule engine can be used. As a proof of concept implementation, the Jena reasoner and the Drools rule engine has been integrated and separate tests regarding requirements and efficiency were conducted. For the Jena reasoner implementation a 2 step rule selection method was followed which resulted in computationally efficient reasoning. The tests concluded that both approaches perform significantly better than the earlier GalenOWL system, while at the same time maintaining the same quality of results and improving performance. Concerning future work, the addition of dosage recommendations in the rules is an ongoing work. Additionally, the possibility to add probabilistic reasoning will be investigated. To this end, Drools is being extended with a fuzzy reasoning engine (Sottara *et al.*, 2008), which while it's

still in development, it's actively supported and it is mature enough to be able to use it as a testing framework.

ACKNOWLEDGMENTS

This work has been supported by the national project “Panacea”, funded by GSRT Hellas under the “Support for SMEs” programme.

REFERENCES

- Adnan, M., Warren, J., and Orr, M. (2010). Ontology based semantic recommendations for discharge summary medication information for patients. In *Computer-Based Medical Systems (CBMS), 2010 IEEE 23rd International Symposium on*, pages 456–461.
- Ammenwerth, E., Schnell-Inderst, P., Machan, C., and Siebert, U. (2008). The effect of electronic prescribing on medication errors and adverse drug events: A systematic review. *J Am Med Inform Assoc.*, **15**(5), 585–600.
- Apache Jena (2012). Semantic web framework. <http://jena.apache.org/>.
- Beuscart, R., McNair, P., J. J. B., and PSIP consortium (2009). Patient safety through intelligent procedures in medication: the PSIP project. *Studies in Health Technology and Informatics*, **148**, 6–13.
- Bragaglia, S., Chesani, F., Ciampolini, A., Mello, P., Montali, M., and Sottara, D. (2010). An hybrid architecture integrating forward rules with fuzzy ontological reasoning. In *Proceedings of the 5th international conference on Hybrid Artificial Intelligence Systems - Volume Part I*, HAIS'10, pages 438–445, Berlin, Heidelberg. Springer-Verlag.
- Doulaverakis, C., Nikolaidis, G., Kleontas, A., and Kompatsiaris, I. (2012). GalenOWL: Ontology based drug recommendations discovery. *Journal of Biomedical Semantics*, **3**(14).
- Drools (2012). Business logic integration platform. <http://www.jboss.org/drools>.
- Fischer, M., Vogeli, C., Stedman, M., Ferris, T., Brookhart, M., and Weissman, J. (2008). Effect of electronic prescribing with formulary decision support on medication use and cost. *Archives of Internal Medicine*, **168**(22), 2433–2439.
- Golbreich, C. (2004). Combining rule and ontology reasoners for the semantic web, invited talk, rules and rule markup languages for the semantic web. In *Boley Editors, LNCS 3323*. Springer.
- Golbreich, C., Dameron, O., Bierlaire, O., and Gibaud, B. (2005). What reasoning support for ontology and rules? the brain anatomy case study. In *Proceedings of the KR 2004 Workshop on Formal Biomedical*, pages 60–71.
- Holford, M., Khurana, E., Cheung, K.-H., and Gerstein, M. (2010). Using semantic web rules to reason on an ontology of pseudogenes. *Bioinformatics [ISMB]*, **26**(12), 71–78.
- Motik, B., Grau, B. C., Horrocks, I., Wu, Z., Fokoue, A., and Lutz, C. (2009). OWL 2 Web Ontology Language Profiles, W3C recommendation. <http://www.w3.org/TR/owl2-profiles/>.
- O'Connor, M. and Das, A. (2012). A Pair of OWL 2 RL Reasoners. In *Proceedings of OWL: Experiences and Directions Workshop 2012 (OWLED-2012)*, Heraklion, Greece.
- Sheth, A. (2005). Semantic web & semantic web services: Applications in healthcare and scientific research, keynote talk. In *IFIP Working Conference on Industrial Applications of Semantic Web*, Jyväskylä, Finland.
- SKOS: Simple Knowledge Organization System (2009). W3C recommendation. <http://www.w3.org/2009/08/skos-reference/skos.html>.
- Sottara, D., Mello, P., and Proctor, M. (2008). Adding uncertainty to a rete-oo inference engine. In *Proc. of the International Symposium on Rule Representation, Interchange and Reasoning on the Web. RuleML '08*, pages 104–118.