*October 2007*

# Semantic Web Ontology Utilization for Heart Failure Expert System Design

Marin PRCELA [a], Dragan GAMBERGER [a] and Alan JOVIC [b]
[a] *Rudjer Boskovic Institute, Zagreb, Croatia*
[b] *Faculty of Electrical Engineering and Computing, Zagreb, Croatia*

**Abstract**. In this work we present the usage of semantic web knowledge representation formalism in combination with general purpose reasoning for building a medical expert system. The properties of the approach have been studied on the example of the knowledge base construction for decision support tasks in the heart failure domain. The work consisted of descriptive knowledge presentation in the ontological form and its integration with the heart failure procedural knowledge. In this setting instance checking in description logic represents the main process of the expert system reasoning.

**Keywords**. knowledge management, knowledge bases, cardiovascular, expert systems, heart failure, reasoning

## Introduction

Medical decision support systems are challenging because of the complexity and richness of the medical knowledge involved. Building a decision support system, which can make the procedures of diagnosis, prognosis and therapy more effective and reliable for the patient, and which is optimal in the use of medical and clinical resources, is yet an unattained goal and still presents a great challenge. It also imposes as a test bed for the knowledge representation formalisms which validates their adequacy and sufficiency in such applications.

It has been recognized in many cases that ontologies are appropriate for knowledge encoding within different systems [1,2,3,4,5]. Also, it has been noted that the application of general-purpose ontology reasoners is very beneficial since in that way the knowledge becomes more sharable and maintainable. The possibilities of using the reasoners in specific cases has been widely explored and tested [5,6,7]. In settings where, due to more demanding system requirements, such utilization is not adequate the system tends to become more complex and more difficult to maintain because some amount of knowledge and the reasoning procedure are encoded within the application itself [1,4,8]. Such knowledge representation approaches tend to be system-specific what scales down their reusability.

Semantic web ontology language OWL has recently emerged as de-facto standard for intelligent applications that utilize the ontologies as knowledge formalization tool. OWL, in combination with the SWRL rule language and with domain-independent reasoners, provides a generally recognized expert system development framework.

In this paper we describe the utilization of OWL in medical expert systems applications. We start by presentation of the descriptive ontology constructed for the heart failure domain and then analyse the possibility to include also procedural knowledge in the same ontological representation. Finally, based on experiments with real application we compare rule based reasoning with onological reasoning for the procedural type of the knowledge.

## 1. Knowledge Representation

Modern expert systems generally recognize a few knowledge types and make a clear distinction between them by imposing distinct formalization means and distinct usage routines. Even in the philosophical domain, the knowledge is divided into *descriptive* and *procedural* knowledge.

*Descriptive* knowledge (also referenced as conceptual knowledge) describes the concepts in the domain, and the relations among them. In that way, every concept is described by defining its relation to other, previously defined concepts. On the other hand, *procedural* knowledge describes the procedures and actions that should be taken in given situations. In that sense, the descriptive knowledge is commonly treated as a construct for which practical usage has yet to be stated in the system while the procedural knowledge is very narrowly defined and operationalized, and clearly states what has to be done in specific situations [9,10]. The procedural knowledge commonly refers to the “know what” of the domain, while the procedural knowledge refers to the “know how” of the domain.

The third type of the knowledge recognized in the expert systems is *factual* knowledge. It refers to formalization of facts that describe the given situation, i.e. the problem that is currently being solved. Compared to other knowledge types, the relevance of the factual knowledge is restricted to the ongoing decision tasks and as such considered as generally uninteresting.

## 2. Heart Failure Knowledge Base

The first step in the process of building the knowledge base for the heart failure domain has been the construction of the descriptive heart failure ontology. The ontology is constructed in OWL by the Protégé tool. It is available from the project website (http://www.heartfaid.org/links.php).

The second step in the knowledge base development has been collection and formalization of the related procedural knowledge. This knowledge has been presented in the form of 10 sets of rules. In its development we used only terminology systemized by the previously constructed descriptive ontology.

### *2.1. Heart Failure Ontology*

The HF ontology presents the formalized description of concepts for the heart failure domain. It includes basic HF concepts, properties that characterize patients, all relevant diagnostic examinations and tests, and treatment procedures. The ontology also includes other cardiovascular system related concepts as well as concepts related to

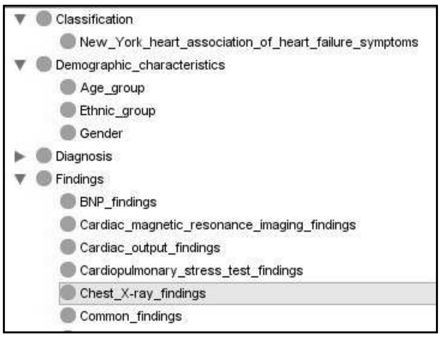


**Figure 1.** Excerpt from the domain description ontology representing the class hierarchy.

other organs when they are connected with HF. The information presented in the ontology has been obtained by human interpretation of guidelines for congestive and acute heart failure.

In its current form the ontology presents the detailed taxonomic overview of the HF domain with around 200 classes describing HF related concepts. These concepts are interconnected with super-class and sub-class relations into a hierarchical tree-like structure. At the basic level there are five relevant super-classes: *HF_concept*, *Patient_characteristic*, *Patient*, *Testing*, and *Treatment*. Figure 1 presents the Protégé tool displaying some of the classes from the HF ontology.

Individuals or instances are members of the classes and typically present exhaustive list of concrete concepts relevant for the class. The realized ontology includes more than 2000 individuals. When possible, classes are specified with their CUI number (Concept Unique Identifier according to UMLS) and with a list of synonyms. For example, for the class Heart diseases its CUI is C0018799 and its synonyms are *Disorder_of_heart*, *Cardiac_diseases*, *Cardiopathy*. Finally, the ontology contains properties that connect individuals in different classes. These properties are relevant because they enable introduction of relations among concepts. For example, individual *Valvular_heart_disease* from the class *Heart_valve_disease* is indicated by the individual *Dyspnea* from the class of *Signs_and_symptoms*. Or that *Hyperkalemia* from the class *Potassium_disorder* may be caused by medications like *Potassium_sparing_diuretics* or *Spironolactone*. The names of these properties are *Indicated* and *MayBeCausedByMedication*. The HF ontology includes definitions of more than 100 properties.

### *2.2. Heart Failure Procedural Knowledge*

Production rules in a form "IF *some condition is true* THEN *make some action*" are a widely used approach for the presentation of procedural knowledge. At the knowledge presentation level it is very important that production rules can be easily understood and corrected by medical experts. In this way the major advantage of presenting procedural knowledge in the form of production rules is that they present formal enough way to present knowledge that can be used by the decision support system and that at the same time medical experts can easily control the expected performance of the system.

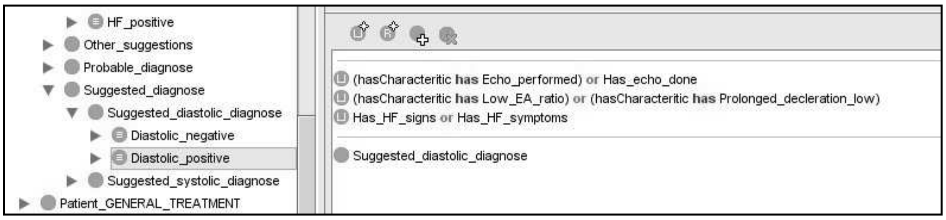


**Figure 2**. Example of concept constructor used for procedural knowledge encoding.

Figure 2 illustrates the possibility to present procedural knowledge in the OWL form. The OWL concept descriptors are used to formalize the conditions while the conclusions or actions that are made by the rule are represented as named OWL classes (concepts). In Figure 2 is presented the rule for the diastolic heart failure diagnosis. The presented conceptualization of the procedural knowledge is relevant because it enables its integration with before described descriptive knowledge. The additional advantage is that by the transformation from the rule form into ontological form the procedural knowledge must be ordered into a tree of sub-classes that stimulates systematization of this knowledge.

The basic task of our expert system is to check on patients characteristics and to act upon them. The basic concept of ontological procedural knowledge is the concept of *Patient*. We have assigned to that concept properties that we have found meaningful, like for example *hasCharacteristic* property which allows multiple *PatientCharacteristic* instances, or *hasTestData* which contains the instances with the numerical values of the patients test measurements. The descriptive part of the knowledge defines what possible characteristics patient might be described by. All classes representing procedural knowledge is a tree of subclasses of the class *Patient*.

## 3. Reasoning in OWL

The major consequence of the transformation of the procedural knowledge into ontological form and its integration with descriptive knowledge is that decision making can be completely performed by the reasoning procedures on ontologies. The experience and conclusions apply also for expert systems in other domains, particularly in other medical domains.

### *3.1. Reasoning in Descriptive Knowledge*

One can recognize two main knowledge profiles in descriptive domain knowledge. The first is defined by the generality relations among instances and classes, as well as by the generality relations among subclasses and super-classes. In this way for each concept presented by some instance there is a series of *is-a* relation. For example, it means that *Cardiomegaly is-a Cardiac_hypertrophy* while Cardiac hypertrophy is-a Heart disease. The second profile of the descriptive knowledge comes from properties that define relations between individuals, such as *Indicated* or *MayBeCausedByMedication*, mentioned before.

The logic part of the OWL language (concept constructors), as we have noticed, in this case appeared to be rather superfluous. We have found that the knowledge is substantially pre-defined and rather static, and that there was no need for describing the terms by concept constructors. The descriptive knowledge took a shape of terminology, and found a purpose just in defining a domain of discourse by listing the concepts within it and placing them in a hierarchical structure. Reasoning in the descriptive knowledge is reduced to mere propagation of *is-a* relationships down the class hierarchy, and in that way is imposed only as a structure preparation phase for reasoning in the procedural knowledge.

Here we should emphasize that this does not in any case reduce the importance of the descriptive knowledge. Domain description provides a basis for the procedural knowledge, and poor design of descriptive knowledge significantly reduces the potency of the complete system.

### *3.2. Reasoning in Procedural Knowledge*

All reasoning tasks in the description logics are reducible to single one, e.g. satisfiability or subsumption [3]. Regarding the computational complexity, application of one or another reasoning task does not impose the additional constraints on the system. In our case, the *instance checking* takes the main role, since it assigns the patient individuals into the specific classes which represent the actions that should be performed on the patient, e.g. patient X to the class *PerformXRayTest.* Instance checking in a way simulates the execution of classical procedural rules. Due to the specific usage of the system, and due to the specific setting of the system, other reasoning tasks do not take such significant roles, although they are helpful in some situations. Namely, in the process of knowledge base building, *satisfiability check* and *consistency check* may detect some amount of contradictions in defining the concept constructors, and hence provide us with some kind of debugging tool, but such paradigm is neither requirement nor standard in classical expert systems.

By using exclusively OWL reasoning we have constrained the expressiveness of the procedural knowledge to the OWL syntax and to the reasoner semantics. In general this setting is appropriate for cases where the procedural knowledge does not require complex mathematical expressions or algorithmic control flow (like functions or loops).

The differences in reasoning between OWL and procedural rules are:

- Data transformations - Production rules generally support the common operations on data (e.g. math operators), while the description logics do not have that possibility. The cardinality restrictions are as close as the description logics have come to the numerical operations on data.
- Control flow - The common thing in procedure definition are the control flow primitives, which enable executing a statement block repeatedly or in a specific order (loops, branching, jumping, subroutines, etc.). The description logics do not use this paradigm.
- Open/Closed world semantics - Description logics use the open world semantics, which understands that the knowledge base in every moment might be incompletely defined, i.e. some statements in the knowledge base might be missing. The closed world semantics assumes that the knowledge in the knowledge base is complete. One of the crucial features of closed world assumption is negation-as-failure, which concludes that given statement is false if it is not currently reachable that it is true within the knowledge base. The production rules might follow either approach. For example SWRL semantics assume open world, while Jess semantics assume closed world.

- Reasoning tasks - Description logics perform many reasoning tasks, like *satisfiability*, *subsumption*, *classification*, *instance checking*, *consistency*, etc. Production rules have different approach; the main task is to update the knowledge base if some conditions are fulfilled.

## 4. Conclusion

A drawback of our approach is reasoning on data values. OWL has poor handling of numerical attributes, and therefore extension of system is necessary. The classical extension of such framework is SWRL, which is usually used to encode the procedural knowledge. In our case we have used it exclusively for simple data manipulation, e.g. determining whether the *E/A_fraction* of a patient is lower than *0,5*. This has shown to be sufficient in most cases. Still, it is not recommended to use it for calculating more complex expressions, e.g. *body_mass_index*. This calculation is done externally (in system specific component), and loaded into the knowledge base. However, by this we have excluded some relevant amount of knowledge out of the knowledge base. Additionally this makes the factual knowledge preparation phase significantly more complex.

OWL has already demonstrated its relevance in many semantic web applications. The idea of using it as the expert system framework is not new but the originality of our approach is to use it for the conceptualization of the procedural knowledge. The major advantage is natural integration of descriptive, procedural, and factual knowledge.

## References

[1] Enrico, C. (2003) *Clinical Decision Support Systems*. (2Rev ed.). Hodder Arnold.

[2] Tierney, W.M., Overhage, J.M., Takesue, B.Y., Harris, L.E., Murray, M.D., Vargo, D.L., McDonald, C.J. (1995) *Computerizing guidelines to improve care and patient outcomes: the example of heart failure*. J Am Med Inform Assoc., 2(5), 316-322.

[3] Baader et al. (2002). *Description Logic Handbook*, edited by F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, P.F. Patel-Schneider, Cambridge University Press, 2002, 5-44.

[4] de Clercq, P.A., Hasman, A., Blom, J.A., Korsten, H. H. M. (2001) *Design and implementation of a framework to support the development of clinical guidelines*. International Journal of Medical Informatics, 64, 285-318.

[5] Noy, N.F., Rubin, D.L., Musen, M.A. (2004) *Making Biomedical Ontologies and Ontology Repositories Work*. IEEE Intelligent Systems, 19(6), 78-81.

[6] D.L. Rubin, O Dameron, Y. Bashir, D. Grossman, P. Dev, M.A. Musen. *Using ontologies linked with geometric models to reason about penetrating injuries.* Artificial Intelligence in Medicine, (AIIM), 37:167 - 176. 2006

[7] Tu, S.W., Musen, M.A. (2001) *Representation Formalisms and Computational Methods for Modeling Guideline-Based Patient Care*. Proc First Europ Workshop on Computer-based Support for Clinic Guidelines and Protocols, 2000:125-142.

[8] Ceusters W, Smith B. *Ontology and Medical Terminology: why Descriptions Logics are not enough*. Proceedings of the conference Towards an Electronic Patient Record (TEPR 2003), San Antonio, 10-14 May 2003

[9] Star, J.R. (2000). *On the Relationship Between Knowing and Doing in Procedural Learning*. In B. Fishman & S. O'Connor-Divelbiss (Eds.), Proceedings of the Fourth International Conference of the Learning Sciences (pp. 80-86). Mahwah, NJ: Erlbaum.

[10] Peleg, M., Gutnik, L.A., Snow, V., Patel, V.L. (2006) *Interpreting procedures from descriptive guidelines*. Journal of Biomedical Informatics, 39(2), 184-195.